

LABEL 000000000PRINTER00175122CC EX OBJECT/KEYWORD;FILE SOURCEFILE=SYMBOL/TSSMCP;END\*

OBJECT /KEYWORD

SAVE REAL PROCEDURE GETSPACE(SIZE,TYPE,SAVEF);	00014000
SAVE REAL PROCEDURE WAITIO(IOD,MASK,U);%	00018000
PROCEDURE ERRORFIXER(TYPE); VALUE TYPE; REAL TYPE; FORWARD;	00019500
SAVE PROCEDURE SLEEP(ADDRESS,MASK);%	00020000
SAVE PROCEDURE MCPIN(S); VALUE S; REAL S; FORWARD;	00024500
SAVE PROCEDURE LOGTIMING(MIX,B); VALUE MIX,B; REAL MIX,B; FORWARD;%	00025000
SAVE PROCEDURE ESPBIT; COMMENT PRESENCE BIT ROUTINE FOR ESP SEGMENTS ;%	00025900
STREAM PROCEDURE MOVE(N)*WORDS FROM*(HERE)*TO*(THERE);%	00082000
PROCEDURE STOP; FORWARD;	00089100
PROCEDURE SETMONITORFILE(STOP); VALUE STOP; REAL STOP; FORWARD;	00089510
PROCEDURE GETMONITORROW; FORWARD;	00089520
PROCEDURE LOGDISK; FORWARD;	00089530
PROCEDURE LINEMESSAGES(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	00089540
PROCEDURE WHATSGOINGON(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	00089550
PROCEDURE CHANGEINTRINSICFILE(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	00089560
PROCEDURE CHANGEMCP(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	00089570
PROCEDURE PRINTDIRECTORY(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	00089590
PROCEDURE MIXPRINT(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	00089600
SAVE PROCEDURE FORGETSPACE(LOC);%	00090000
SAVE PROCEDURE LOGTIMING(MIX,B); VALUE MIX,B; REAL MIX,B;	00165050
REQUESTED ADDRESS IS UNLOCKED, THE PROCEDURE FINDFREEADDRESS	00296380
SAVE REAL PROCEDURE TWO(N); VALUE N; INTEGER N;	00306000
PROCEDURE FORGETUSERDISK(A,L);VALUE A,L;REAL A,L;FORWARD;%	00316000
REAL PROCEDURE PETUSERDISK(N,T);VALUE N,T;REAL N,T;FORWARD ;	00316100
PROCEDURE DT;FORWARD;	00317000
REAL PROCEDURE EXP; FORWARD;	00317010
PROCEDURE SCHEDIO(NUM,TYPE,ADR);	00318110
PROCEDURE SCHEDIDLE(ADR); VALUE ADR; REAL ADR; FORWARD;	00318140
PROCEDURE USERDISKSPECIALCASE(Q,R,U,J);VALUE Q,J;REAL Q,R,J;	00336100
PROCEDURE FORGETESPDISK(SEG);VALUE SEG;REAL SEG;FORWARD;	00364000
SAVE INTEGER PROCEDURE DISKSPACE(NWORDS,P1MIX,AUX);	00365000
PROCEDURE STATUS;%	00369000
PROCEDURE INTERRUPT(TYPE); VALUE TYPE; REAL TYPE; FORWARD;	00370500
REAL PROCEDURE FINDOUTPUT(MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE,KIND);%	00371000
REAL PROCEDURE FINDINPUT(MID,FID,REEL,CDATE,CYCLE,COBOL,UL,OF,MODE,FN);	00374000
PROCEDURE STARTIMING(FN,U); VALUE FN,U; REAL FN,U; FORWARD;%	00377000
PROCEDURE FILEOPEN(X,A); VALUE X,A; INTEGER X,A; FORWARD;	%R9000379000
PROCEDURE ENTERSYSMTR(N); VALUE N; REAL N; FORWARD;	00379010
SAVE PROCEDURE SAVEOPEN(A); VALUE A; REAL A;	%R9000379100
PROCEDURE SETNOTINUSE(U,RWL); VALUE U,RWL; REAL U,RWL; FORWARD;	00380000
PROCEDURE FILLBUFFERS(CURRENT,FINAL,COBOL,NR);	00385000
PROCEDURE REALFILECLOSE(A); VALUE A; REAL A; FORWARD;	%R9000389000
SAVE PROCEDURE FILECLOSE(A); VALUE A; REAL A;	%R9000389100
REAL PROCEDURE DISKADDRESS(MID,FID,FPB3,A,H,IO);	% (SHM)00390000
PROCEDURE BLASTQ(U); VALUE U; REAL U; FORWARD;%	00392000
REAL PROCEDURE FILEHEADER(MID,FID,NROWS,SIZE,BLEN,RLEN,S);%	00393000
PROCEDURE PURGET(U); VALUE U; INTEGER U; FORWARD;%	00397000
DIRECTORYSEARCH, NSECOND, AND CLEANOUT ARE THE PROCEDURES	00418970
PROCEDURE LOCKER(SEGMENT);	00422500
PROCEDURE SELECTRUN(F); VALUE F; REAL F; FORWARD;	00426900
PROCEDURE CONTROLCARD(A);VALUE A;REAL A; FORWARD;%	00427000
REAL PROCEDURE DIRECTORYSEARCH(A,B,C);VALUE A,B,C;%	00428000
PROCEDURE NEXTDCIO;FORWARD;	00429100
PROCEDURE ARTN(A,N); VALUE A,N; ARRAY A[*]; INTEGER N; FORWARD;%	00431000
SAVE PROCEDURE DISKWAIT(CORE,SIZE,DISK);	00431100
PROCEDURE MAKEPRESENT(A); VALUE A; REAL A; FORWARD;	00431500
SAVE PROCEDURE DISKIO(L,C,S,D); VALUE C,S,D; REAL L; INTEGER C,S,D;%	00432000
PROCEDURE SPOUTIT(MESSAGE,TYPE);	00450300
PROCEDURE SPOUTER(MESSAGE,UNITNO,TYPE);	00451800
PROCEDURE FILEMESSAGE(I,K,M,F,R,D,C,TYPE);	00452600

PROCEDURE LBMESS(FN,SN,I1,I2,E,UNITNO,X);	00454000
PROCEDURE TERMINATE(MIX); VALUE MIX; REAL MIX; FORWARD;	00463100
SAVE PROCEDURE TERMINALMESSAGE(N); VALUE N; REAL N; FORWARD;	00463200
BOOLEAN PROCEDURE SYSTEMFILE(A,B); VALUE A,B; REAL A,B; FORWARD;	00463300
PROCEDURE ENTERSYSFILE(N); VALUE N; REAL N; FORWARD;	00464000
PROCEDURE COM5; FORWARD;	00469000
PROCEDURE FILLSYSTAT; FORWARD;	00469100
PROCEDURE SAVESTATISTICS; FORWARD;	00469200
PROCEDURE ASR; FORWARD;	00474000
PROCEDURE COM11; FORWARD;	00475000
PROCEDURE COM13; FORWARD;	00477000
PROCEDURE COMMUNICATED; FORWARD;	00478000
PROCEDURE COMMUNICATE1; FORWARD;	00478500
PROCEDURE LIBRARYLOAD; FORWARD;	00479000
PROCEDURE LIBRARYZERO; FORWARD;	00479500
PROCEDURE LIBRARYDUMP; FORWARD;	00480000
PROCEDURE DUMPCORE(B); VALUE B; REAL B; FORWARD;	00480100
PROCEDURE COM19; FORWARD;	00483000
PROCEDURE COM23; FORWARD;	00487000
PROCEDURE INTRINSICTABLEBUILDER(FH);	00489000
PROCEDURE MESSAGETABLEBUILDER; FORWARD;	00491000
SAVE PROCEDURE RESULT;	00646000
SAVE PROCEDURE PUNT(I); VALUE I; REAL I;	00650000
PROCEDURE CREATELOG(DDD); VALUE DDD; ARRAY DDD[*]; FORWARD;	00750000
SAVE PROCEDURE PAUSE(R); VALUE R; REAL R;	00801000
SAVE PROCEDURE CHECKLINKS(MIX,LOC); VALUE MIX,LOC; REAL MIX,LOC;	00803000
SAVE PROCEDURE SWAP(STATE,B); VALUE STATE,B; REAL STATE,B; FORWARD;	00990000
PROCEDURE SHORTCOMMUNICATES; FORWARD;	00990500
PROCEDURE INITIALSWAP(N); VALUE N; REAL N; FORWARD;	00991000
PROCEDURE BRINGBACK(MIX); VALUE MIX; REAL MIX; FORWARD;	00992000
PROCEDURE REENTER(STUFF); VALUE STUFF; REAL STUFF; FORWARD;	00992100
PROCEDURE SWAPPER; FORWARD;	00992200
BOOLEAN PROCEDURE OUTWAIT(B); BOOLEAN B; FORWARD;	00993000
SAVE REAL PROCEDURE GETAREA(N); VALUE N; REAL N; FORWARD;	00994000
SAVE PROCEDURE FORGETAREA(N,A); VALUE N,A; REAL N,A; FORWARD;	00994500
SAVE INTEGER PROCEDURE SPACE(N);	00995000
SAVE PROCEDURE LINKEU; FORWARD;	00998500
SAVE PROCEDURE FORK(ROUTINE, PARAMETER, PRIORITY, SIZE, LOCATION);	01070000
SAVE PROCEDURE KILL(STACK);	01089000
SAVE PROCEDURE SLEEP(ADDRESS, MASK);	01099000
SAVE PROCEDURE COMPLEXSLEEP(CODE);	01123000
SAVE PROCEDURE RUN(MIX);	01127000
SAVE PROCEDURE SAVEMIX(MIX,LOGLINE);	01138000
SAVE PROCEDURE HALT;	01150000
SAVE PROCEDURE ENTERLINEQ(ADR,LINE,PRIRTY);	01163100
PROCEDURE DCWRITE(ADR,LINE,SIZE);	01163350
BOOLEAN PROCEDURE BLASTREAD(LINE,C);	01163410
SAVE PROCEDURE IOREQUEST(FINAL,IODESC,LOCATION);	01163500
PROCEDURE OLDWIERDHAROLD; FORWARD;	01164000
PROCEDURE NOTIFYCANDE(MIX); VALUE MIX; REAL MIX; FORWARD;	01164200
PROCEDURE SYSDISKIO(IO,L,A); VALUE IO,L,A; ARRAY A[*]; REAL IO,L; %R600	01164300
SAVE PROCEDURE NEWIO; FORWARD;	01164500
SAVE PROCEDURE STARTIO(U); VALUE U; REAL U; FORWARD;	01165000
SAVE PROCEDURE INITIATEDCIO(IODESC,S);	01166000
PROCEDURE NEXTDCIO;	01190000
SAVE PROCEDURE ENTERREADYQ(T);	01223000
PROCEDURE USASITAPE(AREA,TYPE,FROM,U,DIR); %RHR	01250100
INTEGER PROCEDURE AUXILIARYSPACE(SIZE);	01267000
PROCEDURE FORGETAUXILIARYSPACE(SIZE,LOC);	01271000
PROCEDURE FILLORKILL(A, START, SIZE, TYPE);	01275000

PROCEDURE REPORTBACK(WHY,P1,P2);	02016000
PROCEDURE MAKELOG(M,T); VALUE M,T; REAL M,T; FORWARD;	02020000
REAL PROCEDURE KEYIN(B); VALUE B; BOOLEAN B; FORWARD;%	02021000
BOOLEAN PROCEDURE WHYSLEEP(MASK); VALUE MASK; REAL MASK; FORWARD;%	02022000
BOOLEAN PROCEDURE OLAY(LOC,MIXX);	02052500
PROCEDURE SEEKNAM(A,B,C,D,E,N); VALUE A,B; REAL A,B,C,D,E,N; FORWARD;	02052700
PROCEDURE UNHOOQUE(MIX);%	02053000
REAL PROCEDURE GETESPDISK; FORWARD;%	02111000
PROCEDURE DIRECTORYBUILDER(A,DDD);	02112000
PROCEDURE TWXOUT(A,B,C,D); VALUE A,B,C,D; REAL A,B,C,D; FORWARD;	02113200
PROCEDURE MESSAGEWRITER;	02114000
PROCEDURE SPOUTIT(MESSAGE,TYPE);	02132000
PROCEDURE SPOUTER(MESSAGE,UNITNO,TYPE);	02132300
PROCEDURE ENDOFDECK(R); VALUE R; REAL R; FORWARD;	02177100
PROCEDURE PBIO(A,B); VALUE A; REAL A,B; FORWARD;	02178500
PROCEDURE TERMINATE(MIX); VALUE MIX; REAL MIX;%	02180000
REAL PROCEDURE PLACEFINDER(S,A,L);	02187100
PROCEDURE STOPCANDY; FORWARD;	02187600
PROCEDURE TERMINALMESSAGA(N); VALUE N; REAL N;	02188000
SAVE PROCEDURE TERMINALMESSAGE(N); VALUE N; REAL N;	02330100
BOOLEAN PROCEDURE READFROMDISK(H,IB);	02347150
PROCEDURE DRAIN0(UNIT,BUMP,ERROR);	02347200
REAL STREAM PROCEDURE UNITIN(TINU,WHAT); VALUE WHAT;%	02348000
REAL PROCEDURE EUF(A,B,L); VALUE A,B,L; REAL A,B,L; FORWARD;	02379000
INTEGER PROCEDURE CALCULATEPURGE(PURGE);%	02380000
PROCEDURE CHANGEDATE(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	02393100
REAL PROCEDURE TAPELABEL(M,F,R,C,P); VALUE M,F,R,C,P;	%AI02393400
PROCEDURE PRINTCORE(X); VALUE X; REAL X;	%10302394000
PROCEDURE DUMPCORE(BUFF);	%AI02434100
PROCEDURE NAMEID(A,KTR);%	02603000
REAL PROCEDURE TAPELABEL(MULFID,FID,REELNO,CYCLE,PURGE);%	02635000
REAL PROCEDURE LABELASCATCH(LBL); VALUE LBL; REAL LBL;%	02659000
PROCEDURE NSECOND; FORWARD;%	02692000
BOOLEAN PROCEDURE SYSTEMFILE(A,B); VALUE A,B; REAL A,B; FORWARD;	02692500
REAL PROCEDURE PUTORTAKE(MIX,WHERE,IO,WHAT);	02700000
PROCEDURE DIRECTORYFULL(PASSBY); VALUE PASSBY; REAL PASSBY;	02722000
PROCEDURE DCERR(R);	02740000
PROCEDURE DCBUSY(V);	02761000
PROCEDURE MOREAREAS;	03002000
SAVE PROCEDURE FORGETAREA(N,T); VALUE N,T; REAL N,T;	03015000
SAVE REAL PROCEDURE GETAREA(N); VALUE N; REAL N;	03029000
SAVE PROCEDURE QUEVENT(T,MIX);	03048000
PROCEDURE COMM1; % DISK I/O COMMUNICATE	03053000
PROCEDURE COMM2; % COMMAND LANGUAGE WAIT COMMUNICATE	03061000
PROCEDURE LOGOUT; FORWARD;	03070000
REAL PROCEDURE INPUTSCAN(MODE,SOURCE,DEST,NUM,FLAGS);	03071000
REAL PROCEDURE OUTRAN980(ADR,NUM,TYPE,LCC,B);	03075000
PROCEDURE GIVEAWAY(A); VALUE A; REAL A; FORWARD;	03079000
REAL PROCEDURE OUTRANBIDS(ADR,NUM,TYPE,B,C);	03080000
REAL PROCEDURE OUTRANTC(ADR,NUM,TYPE,B,C);	03080200
PROCEDURE RUNSEPTIC(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	03081000
SAVE PROCEDURE DISPOSAL(L,I,R); VALUE L,I,R; REAL L,I,R; FORWARD;	03081100
PROCEDURE MAKEPRESENT(A); VALUE A; REAL A; FORWARD;	03099000
PROCEDURE COMM9;	03100000
PROCEDURE REPORTBACK(WHY,P1,P2);	03111000
PROCEDURE TWXOUT(ADR,NUM,TYPE,LL);	03124000
PROCEDURE CLEARANK(LL,MCP);	03200000
REAL PROCEDURE DCWRITE(ADR,LINE,SIZE);	03214000
PROCEDURE SPOSET(TYPE,BUFH); VALUE TYPE,BUFH; REAL TYPE,BUFH; FORWARD;	03245900
PROCEDURE QUITTER(LINE);	03246000

PROCEDURE DCWAIT(ADR,LINE,R,MASK,CLCK);	03267000
REAL PROCEDURE INPUTSCAN(MODE,SOURCE,DEST,NUM,FLAGS);	03276800
PROCEDURE COMM13;	03292000
PROCEDURE COMM5;	03375000
PROCEDURE ENTERLINEQ(ADR,LINE,PRIRTY);	03417000
PROCEDURE NAKQUE;	03440000
PROCEDURE DCTIMEOUT(R); VALUE R; REAL R;	03458000
PROCEDURE NOTIFYCANDE;	03504000
PROCEDURE PAPERTAPEIO(R,STA,FLAGS,BUFSZ,MI); % HANDLES END OF MESSAGE	03550000
PROCEDURE HELLO(K);	03573000
PROCEDURE OLDWIERDHAROLD;	03608000
SAVE PROCEDURE GIVEAWAY(A);	03689000
PROCEDURE DCIOFINISH(R);	03710000
SAVE PROCEDURE INITIATEIO(IODESC,MIX,U);%	04000000
SAVE PROCEDURE WAITORSWAP(U,A); VALUE U,A; REAL U,A;	04014200
SAVE PROCEDURE QUEUEUP(U); VALUE U; REAL U;%	04016000
SAVE PROCEDURE LINKEU;	04019500
SAVE PROCEDURE STARTIO(U); VALUE U; REAL U;%	04020000
SAVE PROCEDURE PRINTERFINISH(U); VALUE U; REAL U;%	04035000
SAVE PROCEDURE IOREQUEST(FINAL,IODESC,LOCATION);%	04040000
SAVE PROCEDURE FINISHOFFIO(U); VALUE U; REAL U;%	04067000
PROCEDURE PROGRAMRELEASE;%	04099000
SAVE PROCEDURE NEWIO;%	04115000
PROCEDURE DISKORAUXERROR(R); VALUE R; REAL R; FORWARD;	04121410
PROCEDURE ACTUALIOERR(R); VALUE R; REAL R; FORWARD;	04121425
PROCEDURE LINKUP(TYPE,KEY); VALUE TYPE,KEY; REAL TYPE,KEY; FORWARD;%	04121450
PROCEDURE CHECKJOBORFILEMESS(MIX,FIB,U);%	04121500
PROCEDURE LOGOUTMAINT(B); VALUE B; REAL B; FORWARD;%	04121600
PROCEDURE MAINTLOGGER(B); VALUE B; REAL B; FORWARD;%	04121650
SAVE PROCEDURE IOFINISH(R,C); VALUE R,C; REAL R,C;	04122000
SAVE REAL PROCEDURE WAITIO(IO,MASK,U);%	04240000
REAL PROCEDURE TAPEPARITYRETRY(R,U,KEY);%	04254000
PROCEDURE DISKORAUXERROR(R); VALUE R; REAL R;	04256000
END PROCEDURE DISKORAUXERROR;	04353000
PROCEDURE ACTUALIOERR(R); VALUE R; REAL R;	04353200
REAL PROCEDURE TAPEPARITYRETRY(R,U,KEY);%	04548000
REAL PROCEDURE PLACEFINDER(S,A,L);	04700000
REAL PROCEDURE SECURITYCHECK(MID,FID,USE,HEAD);	04790000
PROCEDURE SHEETDIDDLER(BUFF,TYPE,SID); VALUE BUFF,TYPE,SID;	04798000
PROCEDURE ZIPPER(X,Y); VALUE X,Y; REAL X,Y; FORWARD;	04800000
PROCEDURE DISKLOG(MID,FID,H); VALUE MID,FID,H; ARRAY H[*];	%11204800100
PROCEDURE INDIANBOY;	05020000
PROCEDURE INDIANGIRL;	05220000
PROCEDURE COMM15; % INDEPENDENT STARTER FOR CANDE	05407000
PROCEDURE SYSDISKIO(IO,LINE,A);	05423000
PROCEDURE LOGWARN(RC); VALUE RC; REAL RC;	05461000
PROCEDURE FORMTIME(W,T); VALUE W,T; REAL W,T;	05607000
PROCEDURE MAKELOG(MESS,TYPE);	05620000
PROCEDURE STARTCANDY(ESD,PRIORITY);	05655000
PROCEDURE STOPCANDY;	05700000
PROCEDURE SPREADTHEWORD; %UPDATES ADINFO & SENDS H/L MESSAGE TO	05720000
PROCEDURE COMM17;	05750000
BOOLEAN PROCEDURE BLASTREAD(LINE,C);	05758000
PROCEDURE SCRATCHSORT(A,N);	05806500
PROCEDURE SCRATCHDIRECTORYERROR(A,N);	05809600
BOOLEAN PROCEDURE SCRATCHCHECK(I,H,S);	05811170
PROCEDURE SCRATCHSPECIALCASE(CN,A,N,CORADDR,SEGADDR,I,H,SCRATCHSEG);	05811340
PROCEDURE SCRATCHDIRECTORYENTER(A,N);	05813700
PROCEDURE SCRATCHDIRECTORYDELETE(A,N);	05827300
PROCEDURE SCRATCHCLEAN(ARY,BC,LINK);	05837800

REAL PROCEDURE PETUSERDISK(N,T); VALUE N,T; REAL N,T ;	05839400
PROCEDURE FORGETUSERDISK(A,N); VALUE A,N; REAL A,N ;	05847000
PROCEDURE KRUNCHER(H); ARRAY H[*]; FORWARD;	%R5005849900
PROCEDURE SCHEDLOOK(KTR,TYPE); VALUE KTR,TYPE; REAL KTR,TYPE;%	05850000
PROCEDURE SCHEDIDLE(ADR); VALUE ADR; REAL ADR;	05858700
PROCEDURE SCHEDIO(NUM,TYPE,ADR); %%IF FORQUED THEN ONLY 1 PARAM,	05893000
PROCEDURE DKBUSINESS(BUFF); VALUE BUFF; REAL BUFF;	%028-05950000
SAVE PROCEDURE DISKIO(LOCIOD,CORE,SIZE,DISK);%	06000000
PROCEDURE FORGETESPDISK(S); VALUE S; REAL S; FORWARD;	06020500
REAL PROCEDURE GETESPDISK;%	06021000
PROCEDURE FORGETESPDISK(SEGMENT); VALUE SEGMENT; REAL SEGMENT;%	06036000
PROCEDURE DISKBUG;%	06046000
PROCEDURE CLEANOUT(SYS);	06057340
END; % OF PROCEDURE CLEANOUT	06059820
SAVE PROCEDURE DISKWAIT(CORE,SIZE,DISK);	06061500
PROCEDURE DISKSQUASH(BUFF);	06068000
SAVE PROCEDURE INITIALIZE; FORWARD;	06179400
SAVE REAL PROCEDURE COREND; FORWARD;	06179600
PROCEDURE USERDISKSPECIALCASE(Q,R,UT,J) ;	06350000
PROCEDURE GETMOREOLAYDISK(MIX);%	06400000
REAL PROCEDURE SECURITYCHECK(MID,FID,USERID,HEADER);	06460000
BOOLEAN PROCEDURE OUTWAIT(B); BOOLEAN B;	06467000
REAL PROCEDURE OUTRAN980 (ADR,NUM,TYPE,LCC,B);	%10906500000
REAL PROCEDURE OUTRANBIDS(ADR,NUM,TYPE,B,T);	06590000
REAL PROCEDURE OUTRANIC(ADR,NUM,TYPE,B,C);	06610000
REAL PROCEDURE NEXTCDNUM(UPDATE); VALUE UPDATE; BOOLEAN UPDATE;	07001600
PROCEDURE STARTADECK(N); VALUE N; REAL N; FORWARD;	07002000
PROCEDURE ENTERCONTROLDECK(H); VALUE H; ARRAY H[*]; FORWARD;	07002100
PROCEDURE COM23;%	07004000
PROCEDURE STARTLOADN(KTR); VALUE KTR; REAL KTR;%	07243000
PROCEDURE TABLEOFCONTENTS(B,COUNT);%	07268000
PROCEDURE REMOVEDECK(N,U); VALUE N,U; REAL N,U;	07298000
PROCEDURE DECKREMOVER(B); VALUE B; REAL B;%	07354000
BOOLEAN PROCEDURE READFROMDISK(H,IB);%	07376000
BOOLEAN PROCEDURE PRINTORPUNCHWAIT(Q,PNCH);VALUE Q,PNCH;REAL Q,PNCH;	07405100
PROCEDURE ENDOFDECK(R); VALUE R; REAL R;	07406000
PROCEDURE STARTADECK(N); VALUE N; REAL N;	07422000
PROCEDURE RUNTHEDECK(B);VALUE B; REAL B;%	07457000
PROCEDURE EXTERNALEND(B); VALUE B; REAL B;	07473100
PROCEDURE CHANGEPRORITY(BUFF,MIX); VALUE BUFF,MIX; REAL BUFF,MIX;	07485000
PROCEDURE ENTERCONTROLDECK(H); VALUE H; ARRAY H[*];	07541000
BOOLEAN PROCEDURE MTXIN(I,U,BUFF);%	08000000
PROCEDURE TAPEPURGE(BUFF); VALUE BUFF; REAL BUFF;%	08024000
PROCEDURE MIXPRINT(BUFF); VALUE BUFF; REAL BUFF;%	08058000
PROCEDURE REWINDANDLOCK(WHAT); VALUE WHAT; REAL WHAT;%	08079000
PROCEDURE GIMEDATE(B,DT); VALUE B,DT; REAL B,DT; FORWARD;	%RH 08094500
PROCEDURE PRINTDIRECTORY(BUFF);	%DS%08095000
PROCEDURE PBIO(A,P); VALUE A; REAL A,P; FORWARD;	%P 08170100
PROCEDURE CONTINUITYBIT;%	08171000
BOOLEAN PROCEDURE PRINTORPUNCHWAIT(Q,PNCH);VALUE Q,PNCH;REAL Q,PNCH;	08255000
% THIS PROCEDURE IS RESPONSIBLE FOR STARTING PRNPBT/DISK, IT CHECKS	08255055
PROCEDURE PRINTBACKUP(BUFF); VALUE BUFF; REAL BUFF;	%P 08282000
% THIS PROCEDURE HANDLES THE PB MESSAGE, MAKING THE NECESSARY CHECKS	08282110
PROCEDURE TIMEOUT (B); VALUE B; REAL B;%	08305000
PROCEDURE GIMEDATE(B,DT); VALUE B,DT; REAL B,DT;	08317000
PROCEDURE DISKLOG(MID,FID,H); VALUE MID,FID,H;REAL MID,FID;ARRAY H[*];	08341100
PROCEDURE SETDATE(BUFF); VALUE BUFF; REAL BUFF;%	08343000
PROCEDURE CHANGEDATE(BUFF); VALUE BUFF; REAL BUFF;%	08376000
PROCEDURE SETIME(BUFF); VALUE BUFF; REAL BUFF;%	08390000
REAL PROCEDURE FORMESS(BUFF,H1); VALUE BUFF,H1; REAL BUFF,H1;	08418000

```

PROCEDURE SUSTATUS(A,DDD,B); VALUE A,DDD,B; REAL A,B; ARRAY DDD[*]; 08438900
PROCEDURE OUTPUTLABEL(B); VALUE B; REAL B;% 08439000
PROCEDURE TIMEUSED(B,X); VALUE B,X; REAL B,X;% 08525000
REAL PROCEDURE ANVIL(IL,Z); VALUE IL,Z; REAL IL,Z;% 08546000
PROCEDURE SAVETHEUNIT(B); VALUE B; REAL B;% 08575000
BOOLEAN PROCEDURE WHYSLEEP(MASK); VALUE MASK; REAL MASK; 08599000
END PROCEDURE WHYSLEEP; 08623000
PROCEDURE CHANGEOPTION(BUFF,RS);% 08624000
PROCEDURE TYPPOP(KTR,PO); VALUE KTR,PO; REAL KTR,PO; 08679000
PROCEDURE PBIO(ALPHA,POINTER); VALUE ALPHA; REAL ALPHA,POINTER; %P 08700000
% THIS PROCEDURE HANDLES I/O FOR THE CREATION OF BACK-UP FILES, FOR 08700910
PROCEDURE TIMERELAXER(KTR,TYPE,MIX);% 08730000
PROCEDURE CHANGEFACTOR(BUFF,TF); VALUE BUFF,TF; REAL BUFF; BOOLEAN TF; 08800000
PROCEDURE SHEETDIDDLER(BUFF,TYPE,SID); VALUE BUFF,TYPE,SID; 08850000
PROCEDURE LOGOUT; 09000000
PROCEDURE LOGDISK; 09050000
PROCEDURE LINEMESSAGES(BUFH); VALUE BUFH; REAL BUFH; 09100000
PROCEDURE CALLCANDE(BUFH,TYPE); VALUE BUFH,TYPE; REAL BUFH,TYPE; 09300000
REAL PROCEDURE NEXTAUXMEMWORD(HEADER,FILEPARAM,DISKADDRESS); 09400100
IF DISKADDRESS = 0 THEN % FIRST CALL ON PROCEDURE 09401300
EXIT; END PROCEDURE NEXTAUXMEMWORD; 09404300
PROCEDURE TRANSFERMCPTOAXMEM(HDRADRS,MAXLOC); VALUE HDRADRS,MAXLOC; 09404400
END PROCEDURE TRANSFERMCPTOAXMEM; 09410000
PROCEDURE SETMONITORFILE(STOP); VALUE STOP; REAL STOP; 09410200
END PROCEDURE SETMONITORFILE; 09423800
PROCEDURE GETMONITOROW; 09423900
END PROCEDURE GETMONITOROW; 09428800
PROCEDURE ENTERSYSMTR(N); VALUE N; REAL N; 09428900
EXIT; END PROCEDURE ENTERSYSMTR; 09433500
PROCEDURE CHANGEAUXFILES(BUFF,WA); 09433600
PROCEDURE WHATINTRNSIC(B); VALUE B; REAL B; FORWARD; 09491000
PROCEDURE INTRINSICTABLEBUILDER(FH); VALUE FH; REAL FH; 09500000
BOOLEAN PROCEDURE SYSTEMFILE(A,B); VALUE A,B; REAL A,B; FORWARD; 09550000
PROCEDURE CHANGEINTRINSICFILE(BUFF); VALUE BUFF; REAL BUFF;% 09600000
PROCEDURE CHANGEMCP(BUFF); VALUE BUFF; REAL BUFF; 09679100
BOOLEAN PROCEDURE SYSTEMFILE(A,B); VALUE A,B; REAL A,B;% 09700000
PROCEDURE WHATSGOINGON(BUFH); VALUE BUFH; REAL BUFH; %DS%09800000
PROCEDURE FENCEMOVER(B, BUFF); 09900000
PROCEDURE LINECLEAR(KTR); 09955000
REAL STREAM PROCEDURE GNC(B);% 10001000
BOOLEAN STREAM PROCEDURE TAN(B); VALUE C;% 10007000
REAL PROCEDURE EXP; 10012000
STREAM PROCEDURE FRMT1(Q,T,M,WB); VALUE Q,T,M;% 10047000
PROCEDURE DT; 10082000
REAL PROCEDURE PRNPBTSPECASE1(Z); 12500000
% THIS PROCEDURE HANDLES THE FOLLOWING FUNCTIONS FOR COM19, DEPENDING 12500110
PROCEDURE PRNPBTSPECASE2(Z); 12800000
% THIS PROCEDURE HANDLES ADDITIONAL THINGS FOR COM19, VALUES OF Z ARE: 12800110
PROCEDURE COM19; 13000000
PROCEDURE SPOSET(TYPE,BUFH); 13200000
REAL PROCEDURE ANALYSIS;% 14000000
SAVE INTEGER PROCEDURE ACTUALOVERLAYADDRESS(TYPE, MIX, LOC); 14105000
PROCEDURE AUXILIARYMEMORYCASUALTYRECOVERY; 14120000
PROCEDURE AUXILIARYTABLEINITIALIZE; 14122000
PROCEDURE MAKEPRESENT(C); VALUE C; REAL C;% 14155000
PROCEDURE COM5;% 14343000
PROCEDURE ZIPPER(W1,W2); VALUE W1,W2; REAL W1,W2;% 14531000
REAL PROCEDURE EUF(A,B,L); VALUE A,B,L; REAL A,B,L; 14543000
PROCEDURE COM11; COMMENT ALGOL I/O COMMUNICATE;% 14623000
PROCEDURE DISPLAY(X); VALUE X; REAL X;% 15019000

```

```

PROCEDURE COM13 ;%
PROCEDURE WHATSIT(BUFH); VALUE BUFH; REAL BUFH;
  BOOLEAN PROCEDURE CONQUER(C,N,L,S,G);
REAL PROCEDURE BATCHSELECT(F,N,B,L); VALUE F,N;
PROCEDURE COREPRINT(Q);
  BOOLEAN PROCEDURE PRTGAMES(BUFF,MIX); VALUE BUFF,MIX; REAL BUFF,MIX;
PROCEDURE SPOUTMCP(BUFF); VALUE BUFF; REAL BUFF;
PROCEDURE WHATINTRNSIC(BUFF); VALUE BUFF; REAL BUFF;
REAL PROCEDURE AUXPRINT(Q); VALUE Q; REAL Q;
  END PROCEDURE AUXPRINT;
REAL PROCEDURE KEYINSCAN(KTR,MIX); REAL KTR,MIX;
  END PROCEDURE KEYINSCAN;
PROCEDURE KEYINO(B,KTRX); VALUE B,KTRX; REAL B,KTRX;
  END PROCEDURE KEYINO;
PROCEDURE KEYIN1(B,KTRX); VALUE B,KTRX; REAL B,KTRX;
  END PROCEDURE KEYIN1;
PROCEDURE KEYIN2(KTRX); VALUE KTRX; REAL KTRX;
% AUXILIARY PROCEDURE TO "KEYIN".
% THIS PROCEDURE IS CALLED AS AN INDEPENDENT RUNNER FROM
% PROCEDURE "KEYIN";
  END PROCEDURE KEYIN2;
REAL PROCEDURE KEYIN(B); VALUE B; REAL B;
% THIS PROCEDURE FUNCTIONS AS A DRIVER FOR AUXILIARY PROCEDURES
% THIS PROCEDURE FUNCTIONS AS A DRIVER FOR AUXILIARY PROCEDURES
% "KEYINO","KEYIN1" AND "KEYIN2", PROCEDURES "KEYINO" AND "KEYIN1"
% ARE CALLED DIRECTLY, AND PROCEDURE "KEYIN2" IS FORKED AS AN
  END PROCEDURE KEYIN;
PROCEDURE LBMESS(FN,SN,I1,I2,E,UNITNO,X);
  PROCEDURE STOPM;
PROCEDURE FILEHOLD(A,B,TOG,LOC,HOLD);
  REAL PROCEDURE DIRECTORYSEARCH(A,B,OPTN);%
PROCEDURE COMMUNICATE1;
PROCEDURE COMMUNICATE0;
  PROCEDURE SHORTCOMMUNICATES;
PROCEDURE FRONTEND(MIX);
PROCEDURE SELECTRUN1;
  %%% THOSE DECLARED IN PROCEDURE SELECTRUN.
  THIS PROCEDURE WILL BE EXECUTED, THIS PROCEDURE CAN ALSO
  THIS PROCEDURE WILL BE EXECUTED, THIS PROCEDURE CAN ALSO
  RETURNING TO THAT PROCEDURE.
  % PROCEDURE ONLY
  END PROCEDURE SELECTRUN1;
PROCEDURE SELECTRUN2;
  %%% THOSE DECLARED IN PROCEDURE SELECTRUN.
  END PROCEDURE SELECTRUN2;
PROCEDURE SELECTRUN(F); VALUE F; REAL F;
  %%% VARIABLES DECLARED IN THIS PROCEDURE.
  PROCEDURES "SELECTRUN1" AND/OR "SELECTRUN2" WILL BE EXECUTED,
  PROCEDURE "SELECTRUN1" AND "SELECTRUN2" MAY, IN TURN, SPECIFY
  %%% SEE ALSO "SEGMENT ZERO" SECTION IN PROCEDURE "SELECTRUN2" FOR
PROCEDURE PRINTTHECOVER(CARD,UNITNO,PS);
  PROCEDURE FETCH(UNITNO,CARDLOC,SOURCE);
REAL PROCEDURE SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,
  PROCEDURE SEEKNAM(A,B,C,D,E,N); VALUE A,B; REAL A,B,C,D,E,N;
  REAL PROCEDURE PPC
PROCEDURE SECURITYMAINT( TYPE,SMID,SFID,CMM,SFH,CARD);
  **** ALSO SEE PROCEDURE "SELECTRUN1" (SEQ.NO,20055600) FOR
REAL PROCEDURE LIBCC;
END; % LIBCC PROCEDURE
REAL PROCEDURE CCSET; FORWARD;

```

```

15060000
%DS%15106500
15168000
15200000
15300000
15400000
15500000
15534000
15605000
15609600
16029000
16037900
16038000
16360000
16361000
16609000
16610000
16611000
16612000
16613000
16950000
16951000
16951100
16951100
16951200
16951300
16971000
17000000
17900000
18000000
18155000
18500000
18700000
19500000
%R7519991000
20011200
20016400
20018200
20018200
20018500
20023000
20080200
20080500
20085600
20140300
20140800
20146000
20147200
20147300
20176200
20289010
20292000
20314000
20382010
20383000
20511100
20542800
20566000
20580350
20580400

```



PROCEDURE CCFINISH;	20580800
REAL PROCEDURE CCCOMPILE;	20583800
REAL PROCEDURE INITIALIZEIT;	20586700
REAL PROCEDURE CCUNIT;	20589700
REAL PROCEDURE CCSECMAINT;	20590850
REAL PROCEDURE CCLABEL;	20594850
PROCEDURE CONTROLCARD(CARD); VALUE CARD; REAL CARD;	20597550
REAL PROCEDURE CCSET;	20700000
SAVE PROCEDURE SWAP(STATE,B); VALUE STATE,B; REAL STATE,B;	21000100
PROCEDURE REENTER(STUFF); VALUE STUFF; REAL STUFF;	21005000
PROCEDURE BRINGBACK(MIX); VALUE MIX; REAL MIX;	21012100
SAVE PROCEDURE MCPIN(S); VALUE S; REAL S;	21013000
PROCEDURE EXPANDER(MIX,R);	21026500
BOOLEAN PROCEDURE UNHOOKANDWAIT(MIX,MCP); VALUE MIX,MCP; REAL MIX,MCP;	21027000
PROCEDURE HOOKUPMCP(MIX); VALUE MIX; REAL MIX;	21100000
PROCEDURE SWAPINGIO(MIX,R); VALUE MIX,R; REAL MIX,R;	21112000
STREAM PROCEDURE MOVE(N,H,T); VALUE N,H,T;	21116100
PROCEDURE SWAPPER;	21200000
PROCEDURE INITIALSWAP(N); VALUE N; REAL N;	21265000
PROCEDURE FRONTEND(MIX);	21293000
PROCEDURE EXPANDER(MIX,R);	21313000
PROCEDURE FINDFREEADDRESS(N); VALUE N; REAL N; FORWARD;	22000500
PROCEDURE NSECOND;%	22001000
PROCEDURE STATUS;%	22055000
BOOLEAN PROCEDURE OLAY(LOC,MIXX);	22228000
SAVE PROCEDURE FORGETSPACE(LOC);%	24000000
SAVE REAL PROCEDURE ACTSPACE(SIZE,SAVEF,MIX);	24032000
SAVE INTEGER PROCEDURE DISKSPACE(WORDS,MIX,AUX);	24101000
SAVE REAL PROCEDURE GETSPACE(SIZE,TYPE,SAVEF);	24300000
PROCEDURE FINDFREEADDRESS(N); VALUE N; REAL N;	24600000
IF (FINDFREECTR:=FINDFREECTR+1) GEQ 32 THEN% THIS PROCEDURE IS	24610000
% CALLED FREQUENTLY SO MAKE IT A SAVE PROCEDURE	24611000
SAVE PROCEDURE DISPOSAL(L,I,R);	26000000
PROCEDURE RUNSEPTIC(BUFF);	26060000
BOOLEAN PROCEDURE B6500FORMATTER(CT,I,X,XX,BCL,H,OPTION);	27990100
PROCEDURE LIBRARYLOADSPECIALCASE(Z); VALUE Z; REAL Z;	27997600
PROCEDURE DOES INITIAL SET-UP AND OTHER SPECIAL FUNCTIONS	27997615
PROCEDURE LIBRARYLOAD;	28000000
END; % OF LIBRARY LOAD PROCEDURE	28399000
PROCEDURE LIBRARYDUMP;	28400000
END; % LIBRARY MAINT. PROCEDURE	28730000
PROCEDURE LIBRARYZERO;	28800000
PROCEDURE FILLORKILL(A, START, SIZE, TYPE);	29000000
INTEGER PROCEDURE AUXILIARYSPACE(SIZE);	29100000
PROCEDURE FORGETAUXILIARYSPACE(SIZE,LOC);	29300000
PROCEDURE AUXILIARYTABLEINITIALIZE;	29400000
PROCEDURE AUXILIARYMEMORYCASUALTYRECOVERY;	29500000
PROCEDURE ERRORMESSER(TYPE); VALUE TYPE; REAL TYPE;	30903000
PROCEDURE ERRORFIXER(TYPE); VALUE TYPE; INTEGER TYPE;	31000000
PROCEDURE SKIPFILE(U,D); VALUE U,D; REAL U,D;	31100000
REAL PROCEDURE FINDOUTPUT(MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE,KIND);%	37000000
REAL PROCEDURE FINDINPUT(MID,FID,REEL,CDATE,CYCLE,COBOL,UL,OF,MODE,FN);	37177000
PROCEDURE STARTIMING(FN,U); VALUE FN,U; REAL FN,U;%	37271000
REAL PROCEDURE DISKADDRESS(MID,FID,FPB3,A,H,I0);	% (SHM) 37286000
PROCEDURE SETNOTINUSE(U,RWL); VALUE U,RWL; REAL U,RWL;	37302000
PROCEDURE BLASTQ(U);	37320000
PROCEDURE BUILDLABEL(LABLE,MID,FID,REEL,CDATE,CYCLE,PFACT,PTN,BLKODE,%)	37337000
PROCEDURE FILEMESSAGE(I,K,M,F,R,D,C,TYPE);	37357000
PROCEDURE FILLBUFFERS(CURRENT,FINAL,COBOL,NR);	37385000
REAL PROCEDURE FILEHEADER(MID, FID, NROWS, SIZE, BLEN, RLEN, S);	37418000

PROCEDURE PURGEIT(U); VALUE U; INTEGER U;%	37449000
PROCEDURE KRUNCHER(H); ARRAY H[*];	%R1737500000
PROCEDURE DISKFILEOPEN(ALPHA); VALUE ALPHA; INTEGER ALPHA;%	%R9038000000
PROCEDURE OTHERFILEOPENIN(ALPHA); VALUE ALPHA; INTEGER ALPHA;	38102000
PROCEDURE OTHERFILEOPENOUT(ALPHA); VALUE ALPHA; INTEGER ALPHA;	38200000
PROCEDURE DISKCLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;%	%R9038355000
PROCEDURE BACKCLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;%	%R9038540000
PROCEDURE OTHERCLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;%	%R9038648000
PROCEDURE FILEOPEN(XTRA,ALPHA);	%R9039000000
PROCEDURE CREATELOG(DDD); VALUE DDD; ARRAY DDD[*];	39500000
PROCEDURE SUSTATUS(A,DDD,B); VALUE A,DDD,B; REAL A,B; ARRAY DDD[*];	39900000
PROCEDURE DIRECTORYBUILDER(A,DDD);	40000000
PROCEDURE REALFILECLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;%	%R9041000000
PROCEDURE LINKUP(TYPE,KEY); VALUE TYPE,KEY; REAL TYPE,KEY;	41310100
PROCEDURE CHECKJOBORFILEMESS(MIX,FIB,U);	41312000
PROCEDURE LOGOUTMAINT(B); VALUE B; REAL B;	41316000
PROCEDURE MAINTLOGGER(B); VALUE B; REAL B;	41327000
PROCEDURE MESSAGEBUILDER;	41430000
[2416 ] = - KEYIN PROCEDURE TO BE CALLED -	41450900
0 = PROCEDURE KEYINO ( DIRECT CALL )	41451000
1 = PROCEDURE KEYINI ( DIRECT CALL )	41451100
2 = PROCEDURE KEYIN2 ( INDEPENDENT RUNNER )	41451200
[36112] = LABEL NUMBER ( SWITCH LOCATION IN PROCEDURE )	41451900
"1**77**",                    %***% END OF FIRST KEYIN PROCEDURE CALLS	41459900
"1**77**",                    %***% END OF SECOND KEYIN PROCEDURES	41469900
PROCEDURE ENTERSYSFILE(N); VALUE N; REAL N;	41600000
PROCEDURE ARTN(A,N); VALUE A,N; ARRAY A[*]; INTEGER N;%	42474000
PROCEDURE ASR; BEGIN INTEGER I,BCNTR; ARRAY AIT[*]; REAL TEMP;%	42482000
SAVE REAL PROCEDURE COREND; FORWARD;	42509000
PROCEDURE INTERRUPT(TYPE); VALUE TYPE; REAL TYPE;	42510000
PROCEDURE FILLSYSTAT;	42600000
PROCEDURE SAVESTATISTICS;	42700000
PROCEDURE INTFINISH; FORWARD;	44008998
SAVE PROCEDURE INITIALIZE;%	44009000
+ (T +                        % FIXEX PROCEDURES	44152500
SAVE REAL PROCEDURE COREND;%	44441000
PROCEDURE INTFINISH;	45000000
PROCEDURE AND EXPECT IT TO RETURN WITH THE SAME VALUE;	45997000

What does procedure  
do?  
TERMINAL MESSAGE  
refer to line 15822100

RUNSEPTIC 26060000

interrupt service routines for  
errors in user programs for  
near line 42510000

tables  
start at 41430800  
OPTIONS  
TERMINAL MESSAGES  
KEYIN MESSAGES (from SPO)  
CC reserved words  
LB messages

KEYBOARDREQUEST 48128000

TANKS 00326000  
TWXOUT 03125000  
DCWRITE 03214000  
DCWAIT 03267000

CALL CANDE 09300000  
examples  
use of KEYBOARD COUNTER and UNIT NO

UNIT NO 16970000

```

%
% B 5 7 0 0   T S - M C P   M A R K   X V , 3 , 0 0           0 2 / 0 7 / 7 4
%
$ SET OMIT = NOT(DEBUGGING)
  $ SET CHECKLINK DUMP SAVERESULTS
$ POP OMIT
$ SET OMIT = DFX
  $ RESET DKBNODFX
$ POP OMIT
BEGIN
  COMMENT: * TITLE: B5500/B5700 MARK XV SYSTEM RELEASE *
           * FILE ID: SYMBOL/TSSMCP TAPE ID: SYMBOL1/FILE000 *
           * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION *
           * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED *
           * EXPECT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON *
           * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF *
           * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232 *
           *
           * COPYRIGHT (C) 1971, 1972, 1973 BURROUGHS CORPORATION *
           * AA320206 AA393180 *
DEFINE MIXMAX=29#;
DEFINE MAXLMAX=48#;
REAL LMAX;
  $ SET OMIT = TWXONLY
REAL STAMAX;
  $ POP OMIT
  $ SET OMIT = NOT(TWXONLY)
DEFINE STAMAX=LMAX#;
  $ POP OMIT
DEFINE FREG=5#; %REMEMBER TO CHANGE THIS DAMN THING IF MORE BED STUFF
DEFINE MARKLEVEL= "XV,3" % MARK LEVEL IN ALPHA
  #, PATCHLEVEL= "00" % PATCH RELEASE LEVEL IN ALPHA
  #, LOCALEVEL= " " % LOCAL LEVEL IN ALPHA
  #;
DEFINE MCPTYPE = 63#;
  DCINTYPE = 62#;
  TSSINTYPE = 61#;
COMMENT THE ESPOL COMPILER APPROPRIATELY TYPES THE MCP &
  INTRINSICS FILE HEADERS SO THAT A VALIDITY CHECK MAY BE MADE
  DURING INITIALIZATION AND AT CI AND CM TIME, HEADER[4],[36:6]
  IS THE FIELD USED TO CONTAIN THE TYPE;
DEFINE ESAD = [1:15]#;
  UNUM = [16:15]#;
  BYBY(BYBY1,BYBY2) =
    BEGIN STREAM(A:=SPACE(10));
      BEGIN DI:=A; DS:=BYBY2 LIT BYBY1; END;
    PUNT(0);
  END#;
DEFINE RESERVEDISKSIZE=2000#;
DEFINE TRACESIZE=30#;%
  $ SET OMIT = NOT(SAVERESULTS OR DEBUGGING)
ARRAY RESULTHOLDER[*]; INTEGER LASTRESULT; % HOLDS LAST DC RESULT DESC
DEFINE
  RESULTMAX=128#; % MUST BE A POWER OF TWO
  STORAWAY=RESULTHOLDER[LASTRESULT+(LASTRESULT+1) AND (RESULTMAX-1)]#;
  $ POP OMIT
REAL JUNK=5;%

```

```

00000010
00001000
00002000
00002100
00002110
00002120
00002160
00002170
00002180
00003000
00003010
00003011
00003012
00003013
00003014
00003015
00003016
00003017
00003018
00003019
%R1300004000
%R1300004500
%R1300004600
00004699
00004700
00004701
00004799
00004800
00004801
00005000
00005010
00005020
00005030
00005040
00005050
00005060
00005070
00005100
00005120
00005140
00005160
00005180
00005185
00005190
00005200
00005210
00005220
00005230
00005240
00005250
00005260
00005300
00006000
00006299
00006300
00006400
00006500
00006600
00006601
00007000

```



```

SAVE PROCEDURE ESPBIT; COMMENT PRESENCE BIT ROUTINE FOR ESP SEGMENTS ;% 00025900
  BEGIN REAL PRTL0C,SYLLABLE; 00026000
    REAL RCW=+0; 00029000
  $ SET OMIT = NOT(NEWLOGGING) 00030000
    BOOLEAN LOGTURNEDOFF; 00030099
    STOPLOG(P1MIX,1); LOGTURNEDOFF+P; 00030100
  $ POP OMIT 00030200
    PRTL0C=(RCW INX 0)&RCW[30;10;2];% 00030201
    STREAM(RSLT+[SYLLABLE],CL+PRTL0C);% 00031000
    BEGIN SI+CL; SI+SI=2; DI+RSLT; DI+DI+6; DS+2 CHR END; 00032000
    PRTL0C + IF SYLLABLE THEN NT4% 00033000
      ELSE SYLLABLE,[36;10];% 00034000
      ELSE SYLLABLE,[36;10];% 00035000
    MCPIN(PRTL0C); 00036000
  $ SET OMIT = NOT(NEWLOGGING) 00053099
    IF LOGTURNEDOFF THEN STARTLOG(P1MIX,0); 00053100
  $ POP OMIT 00053101
    POLISH(0,RDF,0,XCH,FCX,STS);% 00054000
    GO TO POLISH(MEMORY[PRTL0C]);% 00055000
    GO TO START; % PLACE DESC.IN PRT FOR MCP TO AUXMEM TRANSFER 00055100
    END ESPBIT;% 00056000
LABEL FINDIT; 00057100
REAL RESULT1=12 ,RESULT2=13 ,RESULT3=14 ,RESULT4=15 ;% 00058000
DEFINE SIZE=[8;10]#, FILEBIT=[1;1]#,OWNBIT=[2;1]#,% 00059000
  DIMENSIONS=[3;5]#,BLKCNTN=[8;10]#,MOM=[18;15]#,CURBLKCNTN=16#,% 00060000
  AITNDX=6#,PBIT=[2;1]#;% 00061000
  DEFINE FF=18;15#,% 00062000
    MSFF = [16;1]#, 00063000
    CF=33;15#,% 00064000
    CTF=18;33;15#,% 00065000
    FTF=18;18;15#,% 00066000
    CTC=33;33;15#,% 00067000
    FTC=33;18;15#, %PB00067100
    DELTA=11#;% 00068000
REAL CLICK; 00069000
ARRAY TAR[*]; %CONTAINS TOGLE BITS SET BY EACH JOB 00079100
DEFINE LOCKTOG(LOCKTOG1)= BEGIN TOGLE:=TOGLE AND NOT LOCKTOG1; 00079200
  TAR[P1MIX]:=TAR[P1MIX] OR LOCKTOG1; END#; 00079300
DEFINE UNLOCKTOG(UNLOCKTOG1)= BEGIN TOGLE:=TOGLE OR UNLOCKTOG1; 00079400
  TAR[P1MIX]:=TAR[P1MIX] AND NOT UNLOCKTOG1; END#; 00079500
REAL TOGLE; 00080000
DEFINE HP2TOG = TOGLE,[47;1]#, HP2MASK = @1# 00080100
  ,STATUSBIT = TOGLE,[46;1]#, STATUSMASK = @2# 00080200
  ,SHEETFREE = TOGLE,[45;1]#, SHEETMASK = @4# 00080300
  ,STACKUSE = TOGLE,[44;1]#, STACKMASK = @10# 00080400
  ,USERDISKREADY= TOGLE,[42;1]#, USERDISKMASK= @40# 00080600
  ,HOLDFREE = TOGLE,[41;1]#, HOLDMASK = @100# 00080700
  ,NSECONDREADY = TOGLE,[40;1]#, NSECONDMASK = @200# 00080800
  ,SYSDISKTOG = TOGLE,[39;1]#, SYSDISKMASK = @400# 00080900
  ,NEEDSELECT = TOGLE,[38;1]# 00080950
  ,KEYBOARDREADY = TOGLE,[37;1]#, KEYBOARDMASK =@2000# 00081000
  ,NOBACKTALK = TOGLE,[36;1]#, NOBACKTALKMASK=@4000# 00081100
  ,INTFREE = TOGLE,[34;1]#, FREEMASK =@20000# 00081300
  ,WORKING = TOGLE,[33;1]# 00081400
  ,CANDEINPUTREADY= TOGLE,[32;1]# % WONT WORK FOR MULTIPLE C&ES 00081410
  ,AREARDY = TOGLE,[26;1]#, AREARDYMASK = @10000000# 00081500
  ,AREASNEEDED = TOGLE,[25;1]# 00081600
  ,MCPFREE=TOGLE,[24;1]#, MCPMASK=@40000000# 00081670
  % USED TO PROTECT DISK SEGMENT ZERO 00081675

```

```

,SCRATCHDIRECTORYREADY = TOGGLE,[23:1]#, 00081680
      SCRATCHDIRECTORYMASK = @100000000# 00081690
      % USED TO PROTECT THE SCRATCHDIRECTORY 00081695
,FINDINGADDRESS=TOGGLE,[22:1]# 00081700
      % SET TRUE WHENEVER THE INDEPENDENT RUNNING ROUTINE 00081705
      % "FINDFREEADDRESS" IS STARTED SO THAT ONLY ONE COPY 00081706
      % WILL BE RUN AT ONE TIME, 00081707
,CDFREE=TOGGLE,[21:1]#,CDMASK=@400000000# 00081710
      % SET TRUE WHEN CONTROL DECK QUEUE IS FREE 00081711
,NOMEMTOG = TOGGLE,[20:1]# % ON IF NOMEM SINCE LAST NSECOND 00081720
,NOMEM = [18:3]# % 18:2 = COUNTER FOR NSECOND 00081725
,SEPTICTANKING = TOGGLE,[13:1]# 00081972
,DIRECTORYTOG = TOGGLE,[12:1]# 00081974
,DIRECTORYMASK = @400000000000# 00081976
$ SET OMIT = NOT(STATISTICS) 00081979
      ,PBUSY = TOGGLE,[4:1]# 00081980
$ POP OMIT 00081981
; 00081999
STREAM PROCEDURE MOVE(N)"WORDS FROM"(HERE)"TO"(THERE);% 00082000
      VALUE N,HERE,THERE;% 00083000
COMMENT WILL MOVE 0 TO 4095 WORDS;% 00084000
      BEGIN LOCAL NDIV64;% 00085000
          SI+LOC N; DI+LOC NDIV64; SI+SI+6; DI+DI+7; DS+1 CHR; 00086000
          SI+HERE; DI+THERE;% 00087000
          NDIV64(DS+32 WDS; DS+32 WDS); DS+N WDS;% 00088000
      END MOVE;% 00089000
PROCEDURE STOPM; FORWARD; 00089100
LABEL DIFFCOM; 00089200
$ SET OMIT = NOT(AUXMEM OR MONITOR) 00089509
PROCEDURE SETMONITORFILE(STOP); VALUE STOP; REAL STOP; FORWARD; 00089510
PROCEDURE GETMONITOROW; FORWARD; 00089520
$ POP OMIT 00089521
PROCEDURE LOGDISK; FORWARD; 00089530
PROCEDURE LINEMESSAGES(BUFF); VALUE BUFF; REAL BUFF; FORWARD; 00089540
PROCEDURE WHATSGOINGON(BUFF); VALUE BUFF; REAL BUFF; FORWARD; 00089550
PROCEDURE CHANGEINTRINSICFILE(BUFF); VALUE BUFF; REAL BUFF; FORWARD; 00089560
PROCEDURE CHANGEMCP(BUFF); VALUE BUFF; REAL BUFF; FORWARD; 00089570
PROCEDURE PRINTDIRECTORY(BUFF); VALUE BUFF; REAL BUFF; FORWARD; 00089590
PROCEDURE MIXPRINT(BUFF); VALUE BUFF; REAL BUFF; FORWARD; 00089600
SAVE PROCEDURE FORGETSPACE(LOC);% 00090000
      VALUE LOC;% 00091000
      REAL LOC;% 00092000
FORWARD;% 00093000
DEFINE WAITSTORE(WAITSTORE1)= 00093100
      IF NOT MEM[WAITSTORE1,0],[17:1] THEN 00093200
          SLEEP([MEM[WAITSTORE1,0]],0&1[17:47:1]); 00093300
DEFINE STOREDY[STOREDY1]=MEM[STOREDY1,0],[17:1]#; 00093400
ARRAY UVROW[*]; 00095000
ARRAY UV = UVROW[*,*]; 00095100
ARRAY UV3 = UVROW[*,*,*]; 00095200
COMMENT THE FOLLOWING ARE ALL DEFINES FOR CONTENTS OF UV ARRAY ; 00095300
DEFINE ELAPSEDLIMIT[ELAPSEDLIMIT1] = UV [ELAPSEDLIMIT1, 0]#, 00095400
      PROCLIMIT[PROCLIMIT1] = UV [PROCLIMIT1, 1]#, 00095500
      IOCOUNT[IOCOUNT1] = UV [IOCOUNT1, 2]#, 00095600
      TOPSK[TOPSK1] = UV [TOPSK1, 3]#, 00095700
      USERCODE[USERCODE1] = UV [USERCODE1, 4]#, 00095800
      PRYOR[PRYOR1] = UV [PRYOR1, 5]#, 00095900
      FSROW[FSROW1] = UV [FSROW1, 6]#, 00096000
      FS[FS1,FS2] = UV3[FS1,6,FS2]#, 00096100
      FPBD[FPBD1] = UV [FPBD1, 7]#, 00096200

```

SEGD[SEGD1]	= UV [SEGD1, 8]#,	00096300
SINFO[SINFO1]	= UV [SINFO1, 9]#,	00096400
DALOCROW[DALOCROW1]	= UV [DALOCROW1, 10]#,	00096500
DALOC[DALOC1,DALOC2]	= UV3[DALOC1,10,DALOC2]#,	00096600
IOTIME[IOTIME1]	= UV [IOTIME1, 11]#,	00096700
INTABLEROW[INTABLEROW1]	= UV [INTABLEROW1, 12]#,	00096800
INTABLE[INTABLE1,INTABLE2]	= UV3[INTABLE1,12,INTABLE2]#,	00096900
PROCTIME[PROCTIME1]	= UV [PROCTIME1, 13]#,	00097000
EVENT[EVENT1]	= UV [EVENT1, 14]#,	00097100
LOGSTOPPED[LOGSTOPPED1]	= UV [LOGSTOPPED1, 15]#,	00097200
NEXT1[NEXT11]	= UV [NEXT11, 16]#,	00097300
NEXT2[NEXT21]	= UV [NEXT21, 17]#,	00097400
OLAYCTR[OLAYCTR1]	= UV [OLAYCTR1, 18]#,	00097500
MCPROCTIME[MCPROCTIME1]	= UV [MCPROCTIME1, 19]#,	00097600
MCPIOTIME[MCPIOTIME1]	= UV [MCPIOTIME1, 20]#,	00097700
UVMAX = 21#;		00099000
\$ SET OMIT = STATISTICS		00099999
DEFINE UVSIZE = UVMAX#;		00100000
\$ POP OMIT		00100001
\$ SET OMIT = NOT(STATISTICS)		00100099
DEFINE UVSIZE = 35#;		00100100
DEFINE SWAPS[SWAPS1]	= UV [SWAPS1, 21]#,	00100200
SWAPOUTS[SWAPOUTS1]	= UV [SWAPOUTS1, 22]#,	00100300
CODEPBITS[CODEPBITS1]	= UV [CODEPBITS1, 23]#,	00100400
DATAPBITS[DATAPBITS1]	= UV [DATAPBITS1, 24]#,	00100500
CODEOLAYS[CODEOLAYS1]	= UV [CODEOLAYS1, 25]#,	00100600
DATAOLAYS[DATAOLAYS1]	= UV [DATAOLAYS1, 26]#,	00100700
CORETIME[CORETIME1]	= UV [CORETIME1, 27]#,	00100800
TIMING[TIMING1]	= UV [TIMING1, 28]#,	00100900
MORECPBITS[MORECPBITS1]	= UV [MORECPBITS1, 29]#,	00101000
READYQUETIME[READYQUETIME1]	= UV [READYQUETIME1, 30]#,	00101100
QUETIMING[QUETIMING1]	= UV [QUETIMING1, 31]#,	00101200
INITIALRQTIME[INITIALRQTIME1]	= UV [INITIALRQTIME1, 32]#,	00101300
MOREDPBITS[MOREDPBITS1]	= UV [MOREDPBITS1, 33]#,	00101400
OLAYUSED[OLAYUSED1]	= UV [OLAYUSED1, 34]#,	00101500
STATUVMAX = UVMAX+14#;		00102000
\$ POP OMIT		00102001
REAL DATE=@167;		00111000
COMMENT DATE CONTAINS TODAYS DATE;%		00112000
REAL CLOCK=@170;		00113000
REAL XCLOCK=@171;		00114000
COMMENT CLOCK CONTAINS THE NUMBER OF TIME INTERVAL INTERRUPTS% PROCESSED SINCE HALT-LOAD IN 9-41;%		00115000
REAL READY=@172;		00116000
COMMENT READY CONTAINS THE CONTENTS OF THE READY REGISTER ON% THE LAST READ;%		00121000
		00122000
		00123000
		00124000
COMMENT STATUSBIT IS FALSE IF THE STATUS ROUTINE IS RUNNING AND% TRUE OTHERWISE, THIS PREVENTS TWO COPIES OF STATUS FROM% RUNNING TOGETHER;%		00125000
		00126000
		00127000
ARRAY PRT[*,*];%		00128000
COMMENT PRT[I,*] CONTAINS A DATA DESCRIPTOR WITH PROPER SIZE% FIELD POINTING AT PRT FOR JOB WITH MIX INDEX = I;%		00129000
ARRAY PRTROW=PRT[*];	% MIXMAX+1%	00130000
COMMENT PRTROW IS DOPE VECTORS FOR PRT;%		00131000
ARRAY JAR[*,*];%		00132000
% JAR HOLDS INFO OF JOBS IN PROCESS%		00133000
ARRAY INTRNSC[*]; REAL INTSIZE;% RE=ENTRANT INTRINSICS ON USER DISK		00134000
DEFINE REENTRANTINTABLEMAP(REENTRANTINTABLEMAP1)		00135000
=(P(REENTRANTINTABLEMAP1,DUP) AND 15)-(P(XCH)=2)*2#;		00135300
		00135310



```

COMMENT THIS DEFINE MAPS THE VALUES 2,@21-@24 INTO 0 THRU 4. 00135320
IF USED FOR NON-REENTRANT INTRINSICS IT MAY DO STRANGE THINGS. 00135325
IT MUST BE CHANGED IF OTHER INTRINSICS ARE DESIGNATED AS 00135330
REENTRANT. ; 00135335
% THE INTABLEROW FOR JOBS ABOVE THE FENCE IS NOW AS FOLLOWS: 00135340
% WORDS 0 AND 1 ENTRIES FOR REENTRANT INTRINSICS, AS IN 00135350
% BATCH MCP, EXCEPT FOR ABOVE MAPPING. 00135360
% WORDS 2 THRU N ENTRIES FOR PRESENT TYPE 13 INTRINSICS: 00135370
% [FF] = INTRINSIC NUMBER 00135380
% [CF] = CORE ADDRESS 00135390
DEFINE INT13SIZE = 5#; %SIZE OF TYPE 13 INTABLE PORTION 00135400
INT13START = 2#; %FIRST ENTRY IN TYPE 13 INTABLE PORTION 00135500
ARRAY SHEET[*]; % 5% 00136000
% TOGGLES FOR INTEGERLOCKING THE CLOSET & SHEET% 00137000
ARRAY JARROW=JAR[*]; % MIXMAX+1% 00138000
DEFINE TABCNT[TABCNT1] = JARROW[TABCNT1].[FF]#; 00138100
COMMENT TABCNT IS THE NUMBER OF PROCESSES WHICH HAVE CHECKED 00138110
JARROW AND ARE CURRENTLY ACCESSING MIX TABLES. IT ASSURES 00138120
THAT THE TABLES DONT VANISH BENEATH THOSE PROCESSES; 00138130
00151000
COMMENT STACKUSE IS TRUE IF THE INDEPEDENT STACK IS NOT IN USE, 00152000
OTHERWISE FALSE;% 00153000
BOOLEAN NOPROCESSTOG;% 00154000
COMMENT NOPROCESSTOG IS TRUE IF NORMAL STATE PROCESSING IS% 00155000
ALLOWED, OTHERWISE IT IS FALSE. IT IS USED BY OVERLAY AND 00156000
OTHERS TO PREVENT CONFUSION;% 00157000
REAL WITCHINGHOUR,WORDOFEASE; 00157500
COMMENT THESE USED TO BE CONSTANTS IN THE OUTER BLOCK BUT WERE 00157600
MOVED HERE SO EVERYONE COULD USE THEM, THEY CONTAIN: 00157700
WITCHINGHOUR 5184000 00157800
WORDOFEASE @2525252525252525 00157900
00158000
; 00158000
ARRAY ISTACK[*]; 00159000
COMMENT THE INDEPENDENT STACK;% 00160000
% SET OMIT = NOT(NEWLOGGING) 00165009
BOOLEAN LOGTURNEDOFF=+7; % FOR NOTHINGTODO TO USE 00165030
SAVE PROCEDURE LOGTIMING(MIX,B); VALUE MIX,B; REAL MIX,B; 00165050
BEGIN 00165070
COMMENT THE CONTENTS OF THE PARAMETERS ARE AS FOLLOWS: 00165090
MIX MIX INDEX OF PROCESS 00165130
00165150
B.[1:1] 1=STOPLOGGING 0=STARTLOGGING 00165160
B.[47:1] 1=RTN 0=XIT 00165170
; 00165190
IF MIX=0 THEN 00165210
IF B THEN P(0,RTN) ELSE P(XIT); 00165230
IF B.[1:1] THEN 00165250
BEGIN % STOPLOGGING 00165270
IF LOGSTOPPED[MIX] THEN 00165310
BEGIN 00165330
IF B THEN P(0,RTN) ELSE P(XIT); 00165350
END; 00165370
PROCTIME[MIX]+(+P(DUP))+CLOCK+P(RTR); % OFF 00165390
LOGSTOPPED[MIX]+1; 00165410
IF B THEN P(1,RTN) ELSE P(XIT); 00165430
END ELSE 00165450
BEGIN % STARTLOGGING 00165470
IF NOT LOGSTOPPED[MIX] THEN 00165490
BEGIN 00165510
P(XIT); 00165530

```

```

END;
LOGSTOPPED[MIX]*0;
PROCTIME[MIX]+(*P(DUP))-CLOCK=P(RTR); % ON
END;
END LOGTIMING;
$ POP OMIT
DEFINE EUIOHOLDER=DIRECTORYTOP-5#;
EUTAPER=,98#;
DISKAVAILTABLEMAX=130#;
INTEGER NEUP: ARRAY EUIOT[*]; ARRAY PEUIO[*] ;
$ SET OMIT = NOT(SHAREDISK )
INTEGER AVS ;
$ POP OMIT
$ SET OMIT = SHAREDISK
ARRAY AVTABLE[*] ;
$ POP OMIT
COMMENT NEUP,[CF] CONTAINS THE NUMBER OF EUS ON DKA,
NEUP,NEUF CONTAINS THE TOTAL NUMBER OF EUS ON THE SYSTEM,
EUIO AND PEUIO CONTAIN THE I-O TIME USED BY A GIVEN EU,
THIS INFORMATION IS USED BY GETUSERDISK IN AN ATTEMPT TO
MINIMIZE EU CONFLICT;
ARRAY CHANIO[*]; %R5900169000
ARRAY CHANNEL[*]; % 5% 00170000
COMMENT CHANNEL[I] CONTAINS LOGICAL UNIT OF LAST DESCRIPTOR% 00171000
SENT OUT ON CHANNEL I;% 00172000
ARRAY FINALQUE[*]; % 32% 00173000
ARRAY LOCATQUE[*]; % 32% 00174000
COMMENT IOQUE,FINALQUE, AND LOCATQUE TOGETHER WITH UNIT FORM% 00175000
THE I-O QUEUE. AN I-O REQUEST FOR LOGICAL UNIT U REQUIRES% 00176000
THREE WORDS OF SPACE IN THE I-O QUEUE. IF THE REQUEST% 00177000
OCCUPIES POSITION S IN THE I-O QUEUE, THEN IOQUE[S] )% 00178000
I-O DESCRIPTOR FOR THIS REQUEST, FINAL[S] = I-O DESCRIPTOR% 00179000
SKELETON TO BE USED AT I-O COMPLETE TIME TO REBUILD% 00180000
I-O DESCRIPTOR, LOCATQUE[S] = LOCATION OF I-O DESCRIPTOR% 00181000
AT TIME OF REQUEST, LOCATQUE[S] CONTAINS SOME ADDITIONAL 00182000
INFORMATION, IN PARTICULAR;% 00183000
0- 2 = 5% 00184000
3- 7 = MIX INDEX OF REQUESTER% 00185000
8 = I/O IS READ LOCK WHICH HAD ERROR (SHAREDISK), 00185100
9 = OLAY I/O (IOFINISH PLACES RESULT ON ERROR), 00185500
10 = CANDE I/O OR NO MEM MESSAGE, 00186000
11 = ERROR RECOVERY IN PROCESS ON THIS I-O 00186100
12-17 = LOGICAL UNIT NUMBER% 00187000
18-32 = INDEX OF NEXT REQUEST TO BE DONE ON THIS UNIT% 00188000
OR @77777 IF NO NEXT REQUEST% 00189000
33-47 = ORIGINAL LOCATION OF I-O DESCRIPTOR,% 00190000
UNIT[U] CONTAINS INFORMATION ABOUT LOGICAL UNIT U,% 00191000
1- 4 = TYPE OF I/O DEVISE% 00192000
5-12 = ERROR FIELD OF LAST I/O DONE ON THIS UNIT% 00193000
13 = UNIT NOT READY BIT% 00194000
14 = ERROR BIT (ON IF ERROR)% 00195000
15 = WAIT BIT (ON IF UNIT IS WAITING FOR A CHANNEL% 00196000
16-17 = PROCESS BITS (USUALLY BOTH ON IF UNIT IS IN% 00197000
PROCESS OR BOTH OFF. WITH PRINTERS THE% 00198000
I-O FINISH SETS OFF 16 AND THE PRINTER% 00199000
FINISH SETS OFF 17)% 00200000
18-32 = INDEX OF FIRST I-O REQUEST FOR WHICH SERVICE 00201000
IS NOT COMPLETE% 00202000
33-47 = INDEX OF LAST UNSERVICED I-O REQUEST,% 00203000
THE SPACES NOT USED IN THE I-O QUEUE ARE LINKED TOGETHER% 00204000

```

```

    THROUGH IOQUE, THE FIRST AVAILABLE IS IN IOQUEAVAIL;%
REAL IOQUESLOTS,IOQUEAVAIL;
ARRAY IOQUE[*];
DEFINE RETURNIOSPACE(RETURNIOSPACE1) =
    BEGIN IOQUESLOTS:=IOQUESLOTS+1;
        IOQUE[RETURNIOSPACE1]:=IOQUEAVAIL;
        IOQUEAVAIL:=RETURNIOSPACE1;
    END#;

```

```

ARRAY UNIT[32] +%

```

```

@0400007777777777, COMMENT MTA = 0;%
@0400007777777777, COMMENT MTB = 1;%
@0400007777777777, COMMENT MTC = 2;%
@0400007777777777, COMMENT MTD = 3;%
@0400007777777777, COMMENT MED = 4;%
@0400007777777777, COMMENT MEF = 5;%
@0400007777777777, COMMENT MEH = 6;%
@0400007777777777, COMMENT MEJ = 7;%
@0400007777777777, COMMENT MEK = 8;%
@0400007777777777, COMMENT MEL = 9;%
@0400007777777777, COMMENT MEM = 10;%
@0400007777777777, COMMENT MEN = 11;%
@0400007777777777, COMMENT MEP = 12;%
@0400007777777777, COMMENT MER = 13;%
@0400007777777777, COMMENT LES = 14;%
@0400007777777777, COMMENT MET = 15;%
@0600007777777777, COMMENT DRA = 16;%
@0600007777777777, COMMENT DRB = 17;%
@1000007777777777, COMMENT DKA = 18;%
@1000007777777777, COMMENT DKB = 19;%
@0200007777777777, COMMENT LPA = 20;%
@0200007777777777, COMMENT LPB = 21;%
@1400007777777777, COMMENT CPA = 22;%
@0000007777777777, COMMENT CRA = 23;%
@0000007777777777, COMMENT CRB = 24;%
@1200007777777777, COMMENT SPO = 25;%
@2000007777777777, COMMENT PPA = 26;%
@2200007777777777, COMMENT PRA = 27;%
@2200007777777777, COMMENT PRB = 28;%
@2000007777777777, COMMENT PPB = 29;%
@2400007777777777, COMMENT DCA = 30;%
@3600007777777777, COMMENT = 31;%

```

```

$ SET OMIT = SHAREDISK
ARRAY TINU[37] :=
$ POP OMIT
$ SET OMIT = NOT(SHAREDISK)
ARRAY TINU[41] :=
$ POP OMIT

```

```

@ 020010040446321, COMMENT MTA;%
@ 060020040446322, COMMENT MTB;%
@ 120040040446323, COMMENT MTC;%
@ 160110040446324, COMMENT MTD;%
@ 220120040446325, COMMENT MTE;%
@ 260140040446326, COMMENT MTF;%
@ 320210040446330, COMMENT MTH;%
@ 360220040446341, COMMENT MTJ;%
@ 420240040446342, COMMENT MTK;%
@ 460310040446343, COMMENT MTL;%
@ 520320040446344, COMMENT MTM;%
@ 560340040446345, COMMENT MTN;%
@ 620410040446347, COMMENT MTP;%

```

```

00205000
00205500
00206000
00206500
00207000
00207500
00208000
00208500
00209000
00210000
00211000
00212000
00213000
00214000
00215000
00216000
00217000
00218000
00219000
00220000
00221000
00222000
00223000
00224000
00225000
00226000
00227000
00228000
00229000
00230000
00231000
00232000
00233000
00234000
00235000
00236000
00237000
00238000
00239000
00240000
00241000
00241600
00241700
00241701
00241800
00241900
00241901
00242000
00243000
00244000
00245000
00246000
00247000
00248000
00249000
00250000
00251000
00252000
00253000
00254000

```

```

@ 660420040446351, COMMENT MTR;% 00255000
@ 720440040446362, COMMENT MTS;% 00256000
@ 760510040446363, COMMENT MTT;% 00257000
@ 100520040245121, COMMENT DRA;% 00258000
@ 200540040245122, COMMENT DRB;% 00259000
@ 140610040244221, COMMENT DKA;% 00260000
@ 300620040244222, COMMENT DKB;% 00261000
@ 540640000434721, COMMENT LPA;% 00262000
@ 640710000434722, COMMENT LPB;% 00263000
@ 240720000234721, COMMENT CPA;% 00264000
@ 240740040235121, COMMENT CRA;% 00265000
@ 341010040235122, COMMENT CRB;% 00266000
@ 741020000624746, COMMENT SPO;% 00267000
@ 441040000474721, COMMENT PPA;% 00268000
@ 441110000475121, COMMENT PRA;% 00269000
@ 501120000475122, COMMENT PRB;% 00270000
@ 501140000474722, COMMENT PPB;% 00271000
@ 401210000242321, COMMENT DCA;% 00272000
@ 001220000713147, COMMENT ZIP;% 00273000
@ 001240000232421, COMMENT CDA;% 00274000
@ 001310000232422, COMMENT CDB;% 00275000
@ 001320000232423, COMMENT CDC;% 00276000
@ 001340000232424, COMMENT CDD; 00277000
% 00277100
$ SET OMIT = NOT(SHAREDISK) 00277987
@ 000000000627021, COMMENT SYA; 00277988
@ 000000000627022, COMMENT SYB; 00277990
@ 000000000627023, COMMENT SYC; 00277992
@ 000000000627024, COMMENT SYD; 00277994
$ POP OMIT 00277995
@ 000000000446367; COMMENT MTX, ALL SCRATCH TAPES; 00277998
ARRAY WAITQUE[*]; % 8% 00278000
REAL NEXTWAIT,FIRSTWAIT;% 00279000
COMMENT WAITQUE IS A QUEUE OF UNITS FOR WHICH THERE ARE% 00280000
REQUESTS BUT NO CHANNEL IS AVAILABLE, NEXTWAIT AND% 00281000
FIRSTWAIT ARE POINTERS AT THE WAITQUE, NEXTWAIT IS THE% 00282000
NEXT AVAILABLE SLOT IN WAITQUE AND FIRSTWAIT POINTS AT% 00283000
NEXT UNIT TO BE USED WHEN A CHANNEL IS AVAILABLE;% 00284000
ARRAY LABELTABLE[*]; % 32% 00285000
ARRAY MULTITABLE[*]; % 32% 00286000
ARRAY RDCTABLE[*]; % 32% 00287000
ARRAY PRNTABLE[*];% 00288000
ARRAY REPLY[*];% 00289000
COMMENT LABELTABLE, MULTITABLE, AND RDCTABLE CONTAIN LABEL INFORMATION% 00290000
BY LOGICAL UNIT NUMBER AS FOLLOWS;% 00291000
LABELTABLE[I] CONTAINS THE FILE ID FOR LOGICAL UNIT I,% 00292000
MULTITABLE[I] CONTAINS THE CORRESPONDING MULTI-FILE ID,% 00293000
RDCTABLE[I] CONTAINS THE CORRESPONDING REEL NUMBER (IN [14:10]),% 00294000
CREATION DATE (IN [24:17]), AND CYCLE (IN [41:7]);% 00295000
$ SET OMIT = NOT(SHAREDISK) 00295999
DEFINE LQMAX=20#; % SIZE OF THE LQUE ARRAY 00296000
ARRAY LQUE[*]; 00296100
REAL LQAVAIL; 00296200
COMMENT LQUE CONTAINS ONE ENTRY FOR EACH IO THAT IS WAITING FOR 00296300
A LOCKED DISK SFGMENT, 00296310
00296315
LQUE[N],[8:40] CONTAINS THE DISK ADDRESS (BCD) 00296320
LQUE[N],[1:7] CONTAINS AN INDEX INTO THE IOQUE, 00296330
LQAVAIL IS AN INDEX POINTING TO THE FIRST EMPTY WORD IN LQUE, 00296340
00296345

```

```

        WHEN A LOCKED SEGMENT RESULT DESCRIPTOR IS ENCOUNTERED,          00296350
        THE IOQUE ENTRY FOR THAT IO IS DISCONNECTED FROM THE           00296360
        UNIT TABLE AND AN ENTRY IS MADE FOR IT IN LQUE. WHEN THE      00296370
        REQUESTED ADDRESS IS UNLOCKED, THE PROCEDURE FINDFREEADDRESS    00296380
        REMOVES THE ENTRY FROM LQUE AND RECONNECTS IT TO THE UNIT TABLE.00296390
END COMMENT;
$ POP OMIT
REAL OPTION;
REAL SYSDISK,SYSDISKADR;
ARRAY LINETABLE[*]; % LMAX LONG = USED TO KEEP INFO ABOUT LINES
DEFINE
  LOCKED[LOCKED1]=LINETABLE[LOCKED1],[1:1]#, % LOCK BIT FOR DCWAIT    00297140
  DIRECTLINE = [2:1]#, % DIRECT CONNECT FLAG (ALSO SCHEDBUSY)        00297160
  LINEDISC[LINEDISC1]=LINETABLE[LINEDISC1],[3:3]#,
  LINEDIS = [3:3]#, % THE LINE DISCIPLINES ARE:                      00297180
  TTY = 0#,
  CONTENTION = 1#,
  MULTIPOINT = 2#, MULTI = MULTIPOINT#,
  SCHED = 7#,
  BUFSIZE = [6:2]#, % BUFFER SIZE = 0=28, 1=56, 2=112                00297300
  PINGPING = [8:1]#, % ON IF BUFFERS ARE PINGPING                    00297320
  % [9:4] % TERMINAL UNIT                                           00297340
  % [13:1] % GROUPMARK FLAG                                          00297360
  % [14:4] % BUFFER NUMBER                                           00297380
  % [18:2] % ADAPTER TYPE 0=960, 1=SAS                                00297400
  THROWAWAY[THROWAWAY1]=LINETABLE[THROWAWAY1],[20:1]#,              00297420
  LSTATUS[LSTATUS1] = LINETABLE[LSTATUS1],[21:5]#, % SEE BELOW      00297425
  READYQED = [26:1]#, % LINE IS IN READYQ FOR OUTPUT,                00297430
  % [27:5] % NOT USED                                               00297440
  LONGCARRIAGE[LONGCARRIAGE1]=LINETABLE[LONGCARRIAGE1],[32:1]#,      00297460
  % [33:15] % SUPPRESSES LINE FOLDING FOR TWX                        00297480
  % % HEAD OF INPUT Q = ADDRESS FOR RD IF LC                          00297500
  % % THE STATUSES ARE:                                             00297540
  % WRITING STATUSES                                               00297560
  WRB = 0#,
  POLLING = 1#,
  SELECT = 2#,
  ACKING = 3#,
  NAKING = 4#,
  ACKINGENQ = 5#,
  NAKINGENQ = 6#,
  WRBUSY = 7#,
  IDL = 8#, % IDLE STATI
  IDLPOLLING = 9#,
  WAITING = 10#,
  WAITINGENQ = 11#,
  % NOTUSED 12
  % NOTUSED 13
  % NOTUSED 14
  % NOTUSED 15
  NORMAL = 16#, % READING STATI
  FIRSTIME = 17#,
  SELECTANS = 18#,
  ENQREAD = 19#,
  BROKEN = 20#,
  POLLTIMEOUT = 21#,
  TIMEDOUT = 22#,
  EOTREAD = 23#,
  RRA = 24#,
  MSGANSWER = 25#,
  % NOTUSED 26

```

```

%      NOTUSED      27      00298100
%      NOTUSED      28      00298120
%      NOTUSED      29      00298140
%      NOTUSED      30      00298160
DISCON      = 31#;      % END OF LINETABLE DEFINES      00298180
ARRAY STABLE[*];      % STATIONMAX LONG - KEEPS INFO ABOUT STATIONS      00298200
DEFINE      00298220
  PAPERTAPE[PAPERTAPE1]=STABLE[PAPERTAPE1],[1:1]#,      00298240
  ACTIVITY      = [2:1]#,      % SET ON SWAP OR DC10, RESET BY NSECOND      00298260
  SWAPPED      = [3:1]#,      % ON WHENEVER ATTACHED JOB IS SWAPPED ON      00298280
  CANDEFLAG      = [4:1]#,      % ON MEANS INPUT GOES TO CANDE      00298300
  MIXNR      = [5:5]#,      % MIX NUMBER OF JOB TO WHICH LINE      00298320
  MIXFLAG      = [4:6]#,      % IS ATTACHED      00298340
  STATIONTYPE      = [10:3]#,      % THE STATION TYPES ARE:      00298360
  TWX      = 0#,      00298380
  CONRAC      = 1#,      00298400
  TC500      = 2#,      00298420
  BIDS      = 3#,      00298440
  OWHTHROWOUT      = [13:1]#,      % HARRY IS DISCARDING INPUT ON THIS LINE      00298460
  BREAK[BREAK1]      = STABLE[BREAK1],[14:1]#,      % BREAK HAS OCCURRED      00298480
  DIALEDUP      = [15:1]#,      % ON IF STATION IS ALIVE      00298500
  DISCONNECTING[DISCONNECTING1]=STABLE[DISCONNECTING1],[16:1]=0#,      00298520
%      % OFF IF QUITTER OR JOB IS STILL RUNNING      00298540
  QUITN[QUITN1]      = STABLE[QUITN1],[17:1]#,      % INTERLOCK FOR QUITTER      00298560
%      [18:3]      % NOT USED      00298580
  NAKKER      = [21:1]#,      % OWH WILL RETRIEVE LAST TANK ADDRESS      00298581
%      [22:1]      % TEXT BIT FOR PREVIOUS OUTPUT BUFFER      00298582
%      [23:1]      % TEXT BIT FOR CURRENT OUTPUT BUFFER      00298584
  OUTPUTANKING      = [24:1]#,      00298590
  LEENKER      = [25:8]#,      % READY QUEUE LINK IF STA LEQ LINEMAX      00298600
%      % LINE NUMBER IF STA GEQ LINEMAX      00298620
%      [33:15]      % HEAD OF OUTPUT QUEUE      00298640
  ENDOFSTABLE      = #;      00298670
ARRAY SEQARRAY[*];      % STATIONMAX LONG - CONTENTS DEPEND ON LINEDISC      00298680
%      % TWX      00298700
%      [1:1]      % FLAG FOR SEQUENCING      00298720
%      [2:19]      % INCREMENT      00298740
%      [21:27]      % CURRENT SEQUENCE NUMBER      00298760
DEFINE      00298780
%      % MULTIPOINT OR CONTENTION      00298800
  SELECTED      = [1:1]#,      % SELECTED OR ENQ=ED      00298820
  TANKOK[TANKOK1]      = SEQARRAY[TANKOK1],[2:1]#,      % ON IF FULL INPUTANK      00298860
%      [3:1]      % STATION IS NOT IN POLLING LIST      00298880
%      [4:2]      % TIME OUT COUNT      00298900
  NAKMAX      = [6:3]#,      % MAXIMUM NUMBER OF NAK=S BEFORE ABORT      00298920
  NAKCNT      = [9:3]#,      % NUMBER OF NAK=S ON CURRENT IO      00298940
%      [12:12]      % ADDRESS CHARACTERS      00298960
%      [24:1]      % # BEFORE FIRST CHARACTER      00298980
%      [25:1]      % # BEFORE SECOND CHARACTER      00299000
%      [26:6]      % INDEX TO TNAOG      00299020
%      % MULTIPOINT ONLY      00299040
  LINELINK      = [32:8]#,      % CIRCULAR LINK TO ALL STATIONS ON LINE      00299060
%      [40:8]      % STATION IN READY QUEUE (MASTER ONLY)      00299080
%      00299100
%      % SCHEDULE LINES      00299120
%      [1:1]      % ON IF CE IS READY FOR INPUT FROM LINE      00299140
  SCHEND[SCHEM1]      = SEQARRAY[SCHEM1],[2:1]#,      % SCHEDULE TERMINATING      00299160
%      [33:15]      % ADDRESS OF 80 WORD BUFFER ARRAY      00299180
%      ENDOFSEQARRAY      = #;      00299200
  REAL LLNR; %KEEPS LOGICAL LINE NR OF AN IO WHILE IT IS ACTUALLY DONE.      00299300

```

```

REAL BASEDISKADR; % BASE ADDRESS OF C&E'S CURRENT TANK CHUNK 00299400
REAL WORKERSTACK; % ADDRESS OF OLDWIERDHAROLDS STACK, GOTTEN @ INITIALIZE 00299500
REAL WORKERINQ; 00299600
REAL DISKADR, LASTOFFSET, FIRSTOFFSET; 00299700
% MUST BE ARRAY FOR MULTIPLE C&E'S, 00299800
DEFINE WORKER = OLDWIERDHAROLD#; 00299900
ARRAY LASTSEG[*], FIRSTSEG[*]; 00300000
% THESE ARRAYS ARE THE BUFFER SEGMENT NOW BEING FILLED FOR C&E, 00300100
% AND THE BUFFER NEXT TO BE PASSED TO C&E. THEY MAY POINT TO THE 00300200
% SAME AREA IN CORE OR NOT, 00300400
% BOTH MUST BE 2-D FOR MULTIPLE C&E'S, 00300500
DEFINE STARTWORKING = 00300600
IF NOT WORKING THEN 00300700
BEGIN WORKING ;= TRUE; 00300800
FORK(P(.WORKER), 0, =2, 0, WORKERSTACK); 00300900
END #; 00301000
INTEGER INTRGATCTR; 00302000
ARRAY TRANSACTION[*]; % 32% 00304000
DEFINE ETRLNG = 5#; % LENGTH OF ENTRY IN FILE BLOCK% 00305000
SAVE REAL PROCEDURE TWO(N); VALUE N; INTEGER N; 00306000
BEGIN REAL T=+1; 00307000
STREAM(N:=N:=47-N, T:=T); 00308000
BEGIN SKIP N DB; DS:=SET; END; 00308500
END TWO;% 00309000
REAL SYLLABLE;% 00310000
$ SET OMIT = NOT(SHAREDISK) 00310099
REAL SYSNO, SYSMAX; 00310100
COMMENT SYSNO CONTAINS THE HARDWARE SYSTEM NUMBER, 00310110
SYSMAX CONTAINS THE MAXIMUM NUMBER OF SYSTEMS THAT CAN 00310120
BE CONNECTED TOGETHER IN THIS CONFIGURATION, 00310130
BOTH OF THESE PARAMETERS ARE PASSED TO THE MCP BY THE 00310140
MCP LOADER AT HALT/LOAD TIME; 00310150
SYSNO IN M[0], [16:2] AND 00310160
SYSMAX IN M[0], [14:2] 00310170
END COMMENT; 00310180
$ POP OMIT 00310191
$ SET OMIT = SHAREDISK 00310199
DEFINE SYSNO=0#, SYSMAX=1#; 00310200
$ POP OMIT 00310201
COMMENT ANALYSIS PLACES THE SYLLABLE THAT CAUSED THE INTERRUPT 00311000
IN SYLLABLE, THIS IS USED BY PRESENCE BIT, FLAG BIT, AND 00312000
VARIOUS ERRORS;% 00313000
PROCEDURE FORGETUSERDISK(A, L); VALUE A, L; REAL A, L; FORWARD;% 00316000
REAL PROCEDURE PETUSERDISK(N, T); VALUE N, T; REAL N, T; FORWARD ; 00316100
$ SET OMIT = NOT(DEBUGGING) 00316999
PROCEDURE DT; FORWARD; 00317000
REAL PROCEDURE EXP; FORWARD; 00317010
$ POP OMIT 00317011
REAL SCHEDWRD; 00318100
PROCEDURE SCHEDIO(NUM, TYPE, ADR); 00318110
VALUE NUM, TYPE, ADR; 00318120
REAL NUM, TYPE, ADR; FORWARD; 00318130
PROCEDURE SCHEDIDLE(ADR); VALUE ADR; REAL ADR; FORWARD; 00318140
DEFINE 00319100
SCHEDNUM =SCHEDWRD, [CF]#, 00319140
FRSTSCHED =SCHEDWRD, [FF]#, 00319150
LSTSCHED =SCHEDWRD, [3:15]#, 00319160
SCHEDTOG =(NOT(SCHEDWRD, [1:1]))#, 00319170
TYPEINFO =10#, %C&E FILE TYPE FOR SCHEDULE OUTPUT FILE 00319180
SCH[ SCH1 ] = (SCH1[4], [3:3]=SCHED)#, 00319200

```

```

SCHEDBUSY[SCHEDBUSY1]=LINETABLE[SCHEDBUSY1],[2:1]#, 00319250
SCHEDLINE[SCHEDLINE1]=(LINEDISC[SCHEDLINE1]=SCHED)#, 00319300
TANKNDIAL = 27:42:6 #, 00319350
CONNECT(CONNECT1)=STATABLE[CONNECT1];=(*P(DUP))&1[15:47:1] 00319500
&(CANDEMIX[CONNECT1]+32)[4:42:6]#, 00319600
FIOADR = (FIRSTSEG,[CF] - 2)#, 00320000
IOADR = (LASTSEG,[CF] - 2)#, 00320600
LINENR = [10:8]#, 00320950
LINKER = [10:8]#, 00321000
LINEMAX = LMAX#, 00321050
MESSEND = [5:1]#, 00321100
NDSABLE = [7:1]#, 00321300
OFFSET = [3:6]#, 00321400
ROWNR = [10:8]#, 00322300
STATIONMAX = STAMAX#, 00323000
SYSDISKRL = SYSDISK,[1:14]#, 00324000
SYSDISKRPB = SYSDISK,[30:12]#, 00324050
TAILOUT = TANKS[0],[CF]#, 00324500
HEADOUT = TANKS[0],[2:8]#, 00324670
TANKCHUNKSIZE = 256#; 00324700
ARRAY INPUTANK[*]; % STATIONMAX LONG = KEEPS TRACK OF INPUT TANKS 00326000
DEFINE
% [1:1] % LOCK BIT 00326100
% [2:8] % NEXT CHARACTER TO BE DETANKED 00326200
% INPUTL = [10:8]#, % OLDEST SEGMENT IN TANK 00326300
% [18:15] % HEAD OF QUEUE OF 30 WORD AREAS 00326400
% INPUTREADY = [33:1]#, % ON IF JOB SWAPPED TO WAIT FOR INPUT 00326500
% SLOWDOWN = [34:1]#, % ON IF TANKS ARE ALMOST FULL 00326600
% FIRSTBUF = [35:1]#, % ETX IN PREVIOUS INPUT BUFFER 00326700
% [36:1] % NOT USED 00326800
% [38:2] % MODE BITS FOR COMM13 00326900
% INPUTN = [40:8]#, % NUMBER OF SEGMENTS IN TANK 00327000
% CLUMPSIZE = 32#; % EACH LINE HAS 1 CLUMP OF DISK FOR INP 00327100
REAL PROGTANK; % BASE ADDRESS FOR INPUT TANKS 00327200
ARRAY TANKS[*]; % STATINMAX LONG = HANDLES OUTPUT TANKS 00327300
DEFINE
% [1:1] % TANK INTERLOCK BIT 00327400
% TANKLINE[TANKLINE1]=TANKS[TANKLINE1],[2:8]#, % DETANKING QUEUE 00327500
% [10:1] % CANDE SHUT-UP FLAG 00327510
% TANKFUL[TANKFUL1]=TANKS[TANKFUL1],[11:1]#, % TANK FULL BIT 00327515
% [12:1] % SOH BIT ON=SOH OFF=ETX 00327520
% [13:1] % LAST MSG NAKKED WHEN ON 00327525
% TANKL = [14:5]#, % NEXT SEG TO DETANK 00327530
% [19:5] % NEXT WORD OF SEG 00327535
% [24:3] % NEXT CHAR OF WORD 00327540
% SOUSE = 27:6 #, % NO OF SEGS IN USE 00327545
% TANKN = [27:6]#, 00327550
% TANKA[TANKA1]=TANKS[TANKA1],[33:15]#, % CORE ADDRESS OF TANK 00327555
% GLOMSIZE = 32#; 00327560
REAL TANKADDRESS; % BASE ADDRESS FOR OUTPUT TANKS 00327565
% SET OMIT = TWXONLY 00327600
ARRAY TNAOG[*]; % TRANSMISSION NUMBERS AND OTHER GOODIES 00327700
% [1:13] % USE DEPENDS ON STATION TYPE 00327800
% [1:13] % LAST TANK POINTER (FOR NAKS) 00327900
% % TC500 00328990
% [14:14] % NOT USED 00329000
% [28:10] % INPUT TRANSMISSION NUMBER 00329800
% [38:10] % OUTPUT TRANSMISSION NUMBER 00329850
% 00329900
% 00330000
% 00330100
% 00330200

```

TANKS



%	% PAGED SCREEN DEVICES	00330300
%	[14:6] % NUMBER OF LINES PER PAGE	00330500
%	[20:8] % NUMBER OF CHARACTERS PER LINE	00330600
%	[28:8] % NUM OF CHARS SENT ON CURRENT LINE	00330700
%	[36:6] % NUM OF LINES ON PREVIOUS PAGE	00330800
%	[42:6] % NUM OF LINES ON CURRENT PAGE	00330900
	\$ POP OMIT	00330901
	\$ SET OMIT = NOT(DEBUGGING)	00330999
	REAL DTCALL = DT;%	00331000
	DEFINE DDT = P(DTCALL,DEL,ZP1)#;	00332000
	\$ POP OMIT	00332001
	\$ SET OMIT = NOT(AUXMEM OR MONITOR)	00335000
	ARRAY CTABLE[*]; % 10 LONG, AUXERRORTOG IN CTABLE[8]	00335100
	DEFINE AUXERRORTOG = CTABLE[8]#;	00335105
	REAL SYSMTR; % 64 WORD AREA FOR MONITORING SYSTEM FUNCTIONS	00335110
	\$ SET OMIT = NOT(AUXMEM)	00335120
	DEFINE FTABLE[FTABLE1] = CTABLE[2+(FTABLE1)]#;	00335150
	AUXMEMDSK = MCPNAMESEG#; % CONTAINS MCP & INT FILE IDS	00335200
	\$ RESET OMIT	00335201
	DEFINE MCPNAMESEG = (DIRECTORYTOP-7)#;	00335400
	COMMENT MCPNAMESEG CURRENTLY CONTAINS THE FOLLOWING:	00335500
	WORD[ 0]=WORD[15] = FILE IDS OF THE AUXDATA FILES FOR MCP & INTRINCS,	00335600
	WORD[16]=WORD[19] = CONTAIN THE WORD "AUXMEM " AS A MARKER,	00335700
	WORD[20]=WORD[27] = FILE IDS OF THE MCP'S AT HALT/LOAD,	00335800
	WORD[28] = USED BY DISKSQUASH FOR COMM. BETWEEN SHAREDISK SYSTEMS,	00335810
	;	00335900
	REAL OLAYMASK;% FOR LOCKING OUT GETMOREOLAYDISK BY MIX INDEX	00336000
	PROCEDURE USERDISKSPECIALCASE(Q,R,U,J);VALUE Q,J;REAL Q,R,J;	00336100
	ARRAY UL[*]; FORWARD ;	00336110
		00363000
	PROCEDURE FORGETESPDISK(SEG);VALUE SEG;REAL SEG;FORWARD;	00364000
	SAVE INTEGER PROCEDURE DISKSPACE(NWORDS,P1MIX,AUX);	00365000
	VALUE NWORDS,P1MIX,AUX;	00366000
	INTEGER NWORDS,P1MIX; REAL AUX;	00367000
	FORWARD;%	00368000
	PROCEDURE STATUS;%	00369000
	FORWARD;%	00370000
	PROCEDURE INTERRUPT(TYPE); VALUE TYPE; REAL TYPE; FORWARD;	00370500
	REAL PROCEDURE FINDOUTPUT(MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE,KIND);%	00371000
	VALUE MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE,KIND;%	00372000
	REAL MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE,KIND; FORWARD;%	00373000
	REAL PROCEDURE FINDINPUT(MID,FID,REEL,CDATE,CYCLE,COBOL,UL,OF,MODE,FN);	00374000
	VALUE MID,FID,REEL,CDATE,CYCLE,COBOL,UL,OF,MODE,FN;%	00375000
	REAL MID,FID,REEL,CDATE,CYCLE,COBOL,UL,OF,MODE,FN; FORWARD;	00376000
	PROCEDURE STARTIMING(FN,U); VALUE FN,U; REAL FN,U; FORWARD;%	00377000
	PROCEDURE FILEOPEN(X,A); VALUE X,A; INTEGER X,A; FORWARD;	%R9000379000
	\$ SET OMIT = NOT(AUXMEM OR MONITOR)	00379009
	PROCEDURE ENTERSYSMTR(N); VALUE N; REAL N; FORWARD;	00379010
	\$ POP OMIT	00379011
	DEFINE AUXTRACE(AUXTRACE1,AUXTRACE2)=IF CTABLE[4],[1:1] THEN	00379020
	ENTERSYSMTR(NFLAG(AUXTRACE2)&AUXTRACE1[3:42:6])#;	00379030
	SAVE PROCEDURE SAVEOPEN(A); VALUE A; REAL A;	%R9000379100
	BEGIN FILEOPEN(2,A) END;	%R9000379200
	PROCEDURE SETNOTINUSE(U,RWL); VALUE U,RWL; REAL U,RWL; FORWARD;	00380000
		00381000
	DEFINE STOPTIMING=STARTIMING#;	00382000
	PROCEDURE FILLBUFFERS(CURRENT,FINAL,COBOL,NR);	00385000
	VALUE CURRENT,FINAL,COBOL,NR; REAL CURRENT,FINAL,COBOL,NR;	00385500
	FORWARD;	00386000
	DEFINE GETBUFFERS=FILLBUFFERS#;	00387000

PROCEDURE REALFILECLOSE(A); VALUE A; REAL A; FORWARD;	00388000
SAVE PROCEDURE FILECLOSE(A); VALUE A; REAL A;	%R9000389000
BEGIN REALFILECLOSE(A) END;	%R9000389100
REAL PROCEDURE DISKADDRESS(MID,FID,FPB3,A,H,IO);	%R9000389200
VALUE MID,FID,FPB3,A,H,IO;	% (SHM)00390000
REAL MID,FID,FPB3,A,IO; ARRAY H[*];	% (SHM)00390100
FORWARD;	% (SHM)00390200
PROCEDURE BLASTQ(U); VALUE U; REAL U; FORWARD;%	00391000
REAL PROCEDURE FILEHEADER(MID,FID,NROWS,SIZE,BLEN,RLEN,S);%	00392000
VALUE MID,FID,NROWS,SIZE,BLEN,RLEN,S;%	00393000
REAL MID,FID;%	00394000
INTEGER NROWS,SIZE,BLEN,RLEN,S; FORWARD;%	00395000
PROCEDURE PURGEIT(U); VALUE U; INTEGER U; FORWARD;%	00396000
REAL ESPTAB,ESPCOUNT;	00397000
REAL DIRDSK=@177;	00399000
REAL ESPDISKBOTTOM; % LOWEST ADDRESS OF ESPDISK	00400500
REAL ESPDISKTOP; % HIGHEST ADDRESS OF ESPDISK	00401000
REAL MESSAGEHOLDER;%	00401100
DEFINE USEDRA = OPTION,[47:1]#,%	00402000
USEDRB = OPTION,[46:1]#,%	00403000
BOJMESS =OPTION,[45:1]#,%	00404000
EOJMESS =OPTION,[44:1]#,%	00404000
OPNMESS =OPTION,[43:1]#,%	00405000
TERMGO =OPTION,[42:1]#,%	00406000
GIVEDATE = OPTION,[41:1]#,%	00407000
GIVETIME = OPTION,[40:1]#,%	00408000
%39(USED BY DCMCP)	00409000
AUTOPRINT=OPTION,[38:1]#,%	00410000
%37(USED BY DCMCP)	00411000
%36(USED BY DCMCP)	00412000
COPNMESS=OPTION,[35:1]#,%	00413000
CLOSEMESS=OPTION,[34:1]#,%	00414000
ERRORMSG=OPTION,[33:1]#,%	00415000
RETMMSG=OPTION,[32:1]#,%	00416000
LIBMSG=OPTION,[31:1]#,%	00416050
SCHDMSG=OPTION,[30:1]#,%	00416100
SECMSG=OPTION,[29:1]#,%	00416200
DSKTOG=OPTION,[28:1]#,%	00416300
RELTOG=OPTION,[27:1]#,%	00416400
PBDREL=OPTION,[26:1]#,%	00416500
CHECK = OPTION,[25:1]#,%	00416520
DISKMSG = OPTION,[24:1]#,%	00416550
DKLOG = OPTION,[23:1]#,%	00416560
LIBERR=OPTION,[22:1]#,%	00416570
USEPBD=OPTION,[21:1]#,%	00416580
SVPBT =OPTION,[20:1]#,%	00416590
RSTOG=OPTION,[19:1]#,%	%DS00416600
AUTOUNLD=OPTION,[18:1]#,%	00416610
RNALL=OPTION,[17:1]#,%	00416620
CODEOLAY=OPTION,[16:1]#,%	00416630
%15(USED BY DCMCP)	00416640
DATAOLAY=OPTION,[14:1]#,%	00416650
HALTSET=OPTION,[13:1]#,%	00416660
REMOTE=OPTION,[12:1]#,%	00416670
CANDYMESS=OPTION,[11:1]#,%	00416680
BATCHTOG=OPTION,[10:1]#,%	00416690
BACKGROUND=(NOT OPTION,[9:1])#,%	00416700
STOPTEST=OPTION,[8:1]#,%	00416710
PUNCHLCK=OPTION,[7:1]#,%	00416720
	00416730
	00416740

OPTION  
LIST

CDONLY=OPTION,[6:1]#;	00416780
PKTONLY=OPTION,[5:1]#;	00416790
SEPARATE=OPTION,[4:1]#;	00416800
AUTOCE=OPTION,[3:1]#;	00416810
MOD3IOS=OPTION,[2:1]#;	00416990
AUTOMESS = OPTION,[1:1]#;	00416992
XXXXXX= OPTION,[0:0]#;#	00417000
REAL USERDISKBOTTOM;	00418000
% DISK ADDRESS OF USER DISK AVAILABLE TABLE	00418010
REAL DIRECTORYTOP;	00418050
% DISK ADDRESS OF DIRECTORYTOP SEGMENT--STORED IN M[1]	00418060
% BY MCP LOADER	00418070
REAL DISKBOTTOM;	00418100
% DISK ADDRESS OF TOP OF BYPASS DIRECTORY, USED IN SCRAMBLE.	00418200
\$ SET OMIT = NOT(SHAREDISK)	00418799
DEFINE DIRECTORYSEG = (DIRECTORYTOP+2)#;	00418800
\$ SET OMIT = NOT STATISTICS OR OMIT	00418805
REAL BYPASSBOTTOM;	00418810
\$ POP OMIT OMIT	00418811
\$ SET OMIT = SHAREDISK	00418849
REAL HOLDER,NEXTSLOT,BYPASS;	00418850
\$ SET OMIT = NOT STATISTICS OR OMIT	00418859
DEFINE BYPASSBOTTOM = BYPASS,[CF]#;	00418860
\$ POP OMIT OMIT	00418861
DEFINE HOLDMAX = 30#;           % MAXIMUM NUMBER OF ENTRIES IN HOLDLIST	00418900
COMMENT THE HOLDLIST CONTAINS A ONE WORD ENTRY FOR EACH PROCESS	00418910
THAT IS WAITING TO USE A FILE THAT IS ALREADY IN USE.	00418915
HOLDLIST[I],[FF]=THE CORE ADDRESS OF THE WORD THAT THE	00418920
WAITING PROCESS IS SLEEPING ON.	00418925
HOLDLIST[I],[CF]=THE DISK ADDRESS OF THE FILE HEADER	00418930
THAT IS BEING WAITED FOR.	00418935
HOLDLIST[I],[10:8]=MIX INDEX OF THE PROCESS THAT MADE THE	00418937
ENTRY, (TSSMCP ONLY)	00418938
HOLDLIST[I],[2:2]=THE SYSTEM NUMBER (SYSNO) OF THE SYSTEM	00418940
THAT MADE THE ENTRY (SHAREDISK ONLY),	00418945
HOLDLIST[I],[1:1] IS SET BY A SYSTEM TO NOTIFY ANOTHER	00418950
SYSTEM TO AWAKEN THE PROCESS THAT MADE THE ENTRY,	00418955
THE NSECOND ROUTINE EXAMINES THE HOLDLIST IN	00418960
ORDER TO CHECK FOR THIS CONDITION (SHAREDISK ONLY),	00418965
DIRECTORYSEARCH, NSECOND, AND CLEANOUT ARE THE PROCEDURES	00418970
THAT MANIPULATE THE HOLDLIST,	00418975
	00418980
THE WORDS ASSOCIATED WITH DIRECTORY HANDLING ARE:	00418985
HOLDER,[CF] = DISK ADDRESS OF HOLDLIST,	00418990
,[FF] = NUMBER OF ENTRIES IN HOLDLIST,	00418995
NEXTSLOT     = DISK ADDRESS OF FIRST HEADER IN QUEUE OF	00419000
EMPTY SPOTS IN DIRECTORY (NEXTSLOT QUEUE),	00419005
BYPASS,[CF] = LOWEST ADDRESS OF THE BYPASS DIRECTORY,	00419010
,[FF] = HIGHEST ADDRESS OF THE MAIN DIRECTORY,	00419015
ON SHAREDISK, HOLDER, NEXTSLOT AND BYPASS ARE KEPT IN THE FIRST	00419020
THREE WORDS OF THE DISK SEGMENT LOCATED AT DIRECTORYTOP+2, A	00419025
READ LOCK MUST BE DONE BEFORE ACCESSING THE HOLDLIST OR NEXTSLOT	00419030
QUEUE OR EXPANDING EITHER THE MAIN OR BYPASS DIRECTORIES,	00419035
END COMMENT;	00419040
DEFINE SCRAMBLE(SCRAMBLE1,SCRAMBLE2)=(-2*	%10400419100
((SCRAMBLE1,[6:18]+SCRAMBLE1,[24:24]) MOD MODULUS * MODULUS +	%10400419200
(SCRAMBLE2,[6:18]+SCRAMBLE2,[24:24]) MOD MODULUS)+	%10400419300
DISKBOTTOM)#;	%10400419400
MODULUS=13#; DIRMOD=169#;	%10400419500
COMMENT	00419600



```

DEFINE HEADERUNLOCK=HU#,
    HU(HU1,HU2,HU3)=
    P(MKS,HU3,HU1,HU2,9,DIRECTORYSEARC,DEL)#;
REAL DIRECTORYSEARC=DIRECTORYSEARCH;
%%HEADERUNLOCK CAN BE USED TO WRITE IN THE DIRECTORY A CHANGED
%% HEADER. TURN OFF THE INTERLOCK BIT AND DO THE FORGETSPACE
%% IT MAY BE CALLED ONLY AFTER A DIRECTORYSEARCH(A,B,4)
%% THE PARAMETERS PASSED MUST BE (A,B,DS);
%% WHERE A,B ARE THE SAME AS PASSED TO THE DIRECTORYSEARCH
%% AND DS IS THE RESULT OF THAT DIRECTORYSEARCH
PROCEDURE ARTN(A,N); VALUE A,N; ARRAY A[*]; INTEGER N; FORWARD;%
SAVE PROCEDURE DISKWAIT(CORE,SIZE,DISK);
    VALUE CORE,SIZE,DISK;
    REAL CORE,SIZE,DISK;
    FORWARD;
PROCEDURE MAKEPRESENT(A); VALUE A; REAL A; FORWARD;
SAVE PROCEDURE DISKIO(L,C,S,D); VALUE C,S,D; REAL L; INTEGER C,S,D;%
    FORWARD;%
ARRAY MESSAGETABLE[*];
DEFINE MESSAGETABLESIZE = 5#; % NUMBER OF MESSAGETABLE ENTRIES
DEFINE
    OPTIONSZ = (MESSAGETABLE[0],[8:10])#,
    TERMSGSZ = (MESSAGETABLE[1],[8:10])#,
    KEYMSGSZ = (MESSAGETABLE[2],[8:10])#,
    CCTABLSZ = (MESSAGETABLE[3],[8:10])#;
$ SET OMIT = PACKETS
DEFINE
    SPOUT(SPOUT1)=SPOUTIT(SPOUT1,1)#,
    SPOUTER(SPOUTER1,SPOUTER2,SPOUTER3)=SPOUTIT(SPOUTER1,SPOUTER3)#;
PROCEDURE SPOUTIT(MESSAGE,TYPE);
    VALUE MESSAGE,TYPE;
    REAL MESSAGE,TYPE;
    FORWARD;
$ POP OMIT
$ SET OMIT = NOT(PACKETS)
DEFINE
    SPOUT(SPOUT1)=SPOUTER(SPOUT1,0,1)#,
    SPOUTIT(SPOUTIT1,SPOUTIT2)=SPOUTER(SPOUTIT1,0,SPOUTIT2)#;
PROCEDURE SPOUTER(MESSAGE,UNITNO,TYPE);
    VALUE MESSAGE,UNITNO,TYPE;
    REAL MESSAGE,UNITNO,TYPE;
    FORWARD;
$ POP OMIT
DEFINE
    FILEMESS=FMS#,
    FMS(FMS1,FMS2,FMS3,FMS4,FMS5,FMS6,FMS7)=
    FILEMESSAGE(FMS1,FMS2,FMS3,FMS4,FMS5,FMS6,FMS7,1)#;
PROCEDURE FILEMESSAGE(I,K,M,F,R,D,C,TYPE);
    VALUE I,K,M,F,R,D,C,TYPE;
    REAL I,K,M,F,R,D,C,TYPE;
    FORWARD;
PROCEDURE LBMESS(FN,SN,I1,I2,E,UNITNO,X);
    VALUE FN,SN,I1,I2,E,UNITNO,X;
    REAL FN,SN,I1,I2,E,UNITNO,X;
    FORWARD;
PROCEDURE TERMINATE(MIX); VALUE MIX; REAL MIX; FORWARD;
SAVE PROCEDURE TERMINALMESSAGE(N); VALUE N; REAL N; FORWARD;
BOOLEAN PROCEDURE SYSTEMFILE(A,B);VALUE A,B; REAL A,B; FORWARD;
REAL FENCE; ARRAY MEMROW[30];
PROCEDURE ENTERSYSFILE(N); VALUE N; REAL N; FORWARD;

```

```

00430000
00430100
00430200
00430225
00430250
00430275
00430300
00430400
00430500
00430600
00431000
00431100
00431200
00431300
00431400
00431500
00432000
00433000
00435000
00436000
00437000
00438000
00439000
00440000
00441000
00449999
00450000
00450100
00450200
00450300
00450400
00450500
00450600
00450601
00451499
00451500
00451600
00451700
00451800
00451900
00452000
00452100
00452101
00452200
00452300
00452400
00452500
00452600
00452700
00452800
00452900
00454000
00454100
00454200
00454300
00463100
00463200
00463300
00463400
00464000

```

```

PROCEDURE COM5; FORWARD;% 00469000
$ SET OMIT = NOT(STATISTICS) 00469099
PROCEDURE FILLSYSTAT; FORWARD; 00469100
PROCEDURE SAVESTATISTICS; FORWARD; 00469200
DEFINE COUNTUP(COUNTUP1,COUNTUP2)=BEGIN COUNTUPBY(COUNTUP1,1); 00470800
    COUNTUPBY((COUNTUP1)+30,COUNTUP2) END#, 00470900
    COUNTUPBY(COUNTUPBY1,COUNTUPBY2)=COUNTARRAY[COUNTUPBY1];= 00471000
    *P(DUP)+(COUNTUPBY2)#; 00471100
ARRAY COUNTARRAY[*]; 00471200
REAL MCPTOP; 00471300
REAL SYSTATBASE; 00471400
ARRAY SWAPDELAY[*]; 00471600
ARRAY DISKWAITIME[*]; 00471700
REAL LEFTHALF1; % 0&1[23:47:1] = IN PRT SINCE IT OCCURS IN OUTER BLOG 00471750
REAL JOBNUM; 00471800
REAL INTERVAL; 00471900
$ POP OMIT 00471901
PROCEDURE ASR; FORWARD;% 00474000
PROCEDURE COM11; FORWARD;% 00475000
PROCEDURE COM13; FORWARD;% 00477000
PROCEDURE COMMUNICATE0; FORWARD; 00478000
PROCEDURE COMMUNICATE1; FORWARD; 00478500
PROCEDURE LIBRARYLOAD; FORWARD; 00479000
PROCEDURE LIBRARYZERO; FORWARD; 00479500
PROCEDURE LIBRARYDUMP; FORWARD; 00480000
$ SET OMIT = NOT(DUMP OR DEBUGGING) 00480099
PROCEDURE DUMPCORE(B); VALUE B; REAL B; FORWARD; 00480100
$ POP OMIT 00480101
PROCEDURE COM19; FORWARD;% 00483000
PROCEDURE COM23; FORWARD;% 00487000
PROCEDURE INTRINSICTABLEBUILDER(FH); 00489000
    VALUE FH; REAL FH; FORWARD; 00490000
PROCEDURE MESSAGEBUILDER; FORWARD; 00491000
ARRAY PUNTER[*]; 00643000
DEFINE PUNTSIZE = 8 00643100
$ SET OMIT = NOT SHAREDISK 00643200
    + 2 % LQUE OVERFLOW 00643300
$ SET OMIT = NOT AUTODUMP 00643400
    + 19 % DUMP CARD 00643500
$ POP OMIT OMIT 00643600
    #; 00643700
$ SET OMIT = NOT AUTODUMP 00644000
$ SET OMIT = NOT SHAREDISK OR OMIT 00644100
DEFINE DUMPCRD = 10#; 00644200
    DUMPADR = 23#; 00644300
$ POP OMIT 00644350
$ SET OMIT = SHAREDISK OR OMIT 00644400
DEFINE DUMPCRD = 8#; 00644500
    DUMPADR = 21#; 00644600
$ POP OMIT 00644700
COMMENT THIS IS THE CODE ON THE DUMP CARD (ALL NUMBERS ARE OCTAL); 00645000
:20: 20,20,NOP,NOP TELLS ANALYZER ALL I/O RES ARE OK 00645010
:21: STD,5,BFW BRANCH TO 23 00645020
:22: INI,0,LFU TIMER = LOOP UNTIL INTERRUPTED 00645030
:23: 10,LOD,21,STD SAVE M[8], RESTORED BY 2ND CARD 00645040
:24: 25,IIO,2,LBU START I/O THEN WAIT AT TIMER 00645050
:25: 0140000007700035 I/O DESC FOR 77 SEG WRITE FROM 35 00645060
:26: 0140000047400157 I/O DESC FOR 74 SEG READ OF CODE 00645070
:27: OPDC 14,DIA 26,10,BFW I/O 1 = PICK UP RES DESC, 00645080
:30: OPDC 15,DIA 26,6,BFW I/O 2 = DIAL TO ERR FIELD, 00645090

```

```

:31: OPDC 16,DIA 26,2,BFW      I/O 3 - BRANCH INTO I/O 4      00645100
:32: OPDC 17,DIA 26,          I/O 4                          00645110
      DESC 24,CBD 7          BRANCH TO 24 FOR RETRY IF ERRORS 00645120
:33: DESC 37,BFW             GO TO 37 1ST TIME, SEE 41 FOR 2ND 00645130
:34: INI,0,LFU              DATACOM - LOOP UNTIL INTERRUPTED 00645140
:35: 0000000000000501      DISK ADDRESS FOR WRITE          00645150
:36: INI,0,LFU              FREEADDRESS - LOOP ON INTERRUPT 00645160
:37: 200,157,SND,240        STORE DISK ADR FOR READ, SET 240 00645170
                                TO OPERAND FOR DESC AT 41      00645180
:40: STD,OPDC 26,25,STD     PUT I/O DESC INTO 25           00645190
:41: DESC 240,37,STD,NOP   SET 37 FOR BRANCH TO 240 FROM 33 00645200
:42: 16,LBU                 BRANCH TO 24 TO START THE READ; 00645210
SAVE PROCEDURE RESULT;      00646000
BEGIN                        00647000
  GO TO P([18]);             % TIMER IS A LOOP ON INTERRUPTS 00648000
END;                          00649000
$ POP OMIT                   00649100
SAVE PROCEDURE PUNT(I); VALUE I; REAL I; 00650000
  BEGIN REAL T=-3;          00650250
  $SET OMIT = NOT AUTODUMP  00650450
    REAL TMB, RSLT=RESULT;  00650500
    LABEL HA,HB;           00650750
  $ POP OMIT                 00650800
    I:=IF I=0 THEN T ELSE PUNTER INX I; 00651000
    IF MEMROW[P1MIX],[CF] GEQ FENCE THEN 00651300
    BEGIN TERMINATE(P1MIX);  00651400
      TERMINALMESSAGE(P(1,I)); % THE 1 IS THE FROMPUNT FLAG 00651500
    END;                     00651600
    STREAM(Q:=P(0,RDF)); I,  00651800
  $ SET OMIT = NOT AUTODUMP  00651900
    A:=18, D:=I:=PUNTER INX 0); 00652000
  $ SET OMIT = AUTODUMP      00652100
    PUNTER);                 00652200
  $ POP OMIT OMIT           00652300
    BEGIN DS:=13 LIT"=SYS HANG, F="; 00652400
      SI:=LOC Q; SI:=SI+3;    00652600
      5(DS:=3 RESET;        00652800
        3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB)); 00653000
      DS:=2 LIT" "; SI:=I;   00653200
      63(IF SC#" THEN DS:=CHR); DS:=LIT"+"; 00653400
  $ SET OMIT = NOT AUTODUMP  00653500
    DI:=A; DS:=8 LIT"29290+JI"; % INI,INI,4,BBW 00653600
    SI:=A; DS:=44 WDS;       00653800
    DI:=A; DI:=DI+8;         % IOBUSY= 00654000
      DS:=4 LIT"002(";      % 0,RTN 00654200
    DI:=DI+28;               % IOCOMPLETE=L0D R,RTN 00654400
    DS:=32 LIT"0 +A+12(OU+A+12(OY+A+12(O+A+12(") 00654600
  END;                       00654800
  P(HP2);                    00655000
HA: TMB:=I&60[3:42:6];      00655200
P([TMB],I10);               00655400
HB: DO IF (TMB:=P(MKS,RSLT)) = 0 THEN % IO BUSY 00655600
  BEGIN P(MKS,RSLT,DEL); GO HA END 00655800
UNTIL TMB,[3:6]=60;         00656000
IF TMB,[CF]<I THEN GO TO HB; 00656200
IF TMB,[FF]#0 THEN GO TO HA; 00656400
IF NOT HALTSET AND PUNTER[DUMPADR]=#501 THEN 00656600
BEGIN                        00656800
  STREAM(S:=[PUNTER[DUMPCRD]], DI=#20); 00657000
  BEGIN SI:=S; DS:=19 WDS; END; 00657200

```

```

                GO TO P(0,STS,0,STF,[M@20]);
            END;
            DO UNTIL FALSE;
$ SET OMIT = AUTODUMP
            END;
            WHILE HALTSET DO;
            P(WAITIO(PUNTER INX 0,0,25));
            SLEEP(0,0); % MAKE IT EASY FOR ANALYZER TO FIND STACK
$ POP OMIT OMIT
            END;
PROCEDURE CREATELOG(DDD); VALUE DDD; ARRAY DDD[*]; FORWARD;
REAL CANDYINX = CREATELOG;
DEFINE CANDEMIX[CANDEMIX1]=CANDYINX#;
ARRAY MEM=MEMROW[*,*];
DEFINE DSW = 3#,
        BOJW = 4#,
        IGNORE = 5#,
        REMOVED = 6#,
        CHANGED = 7#,
        SECURED = 8#,
        ZIPER = 9#,
        NOTIN = 10#,
        NOTX = 11#,
        WWW = 99#;
$ SET OMIT = NOT(DEBUGGING)
REAL PAUSEVALUE=127;
SAVE PROCEDURE PAUSE(R); VALUE R; REAL R;
BEGIN IF R>PAUSEVALUE THEN P(R,ZP1,DEL); END;
$ POP OMIT
$ SET OMIT = NOT(CHECKLINK OR DEBUGGING)
SAVE PROCEDURE CHECKLINKS(MIX,LOC); VALUE MIX,LOC; REAL MIX,LOC;
    BEGIN REAL B,S,F,L,MS,T,C;
        LABEL PUN;
        P(M[1]);
        B=M[S*MS+MEM[MIX,0]],[CF]],[FF];
        WHILE (F+(L*M[S]),[CF])#MS DO
        BEGIN IF P(L,LOD),[FF] NEQ B THEN
            BEGIN C:=1; GO TO PUN; END;
            IF FSS THEN
                BEGIN C:=2; GO TO PUN; END;
            IF S=LOC THEN LOC*=-1;
            B*S; S+F;
        END;
        IF S=LOC THEN LOC:=-1;
        C*3;
        IF =1#LOC THEN GO TO PUN;
        B*M[(S*MS+S+1)+1];
        WHILE (F+(L*M[S]),[CF])#MS DO
        BEGIN IF M[S+1]#B THEN
            BEGIN C:=4; GO TO PUN; END;
            IF S#MS THEN
                BEGIN
                IF (T*M[S-1])>0 THEN
                    BEGIN C:=5; GO TO PUN; END;
                IF T,[CF]-S-1#L,[FF] THEN
                    BEGIN C:=6;
                    PUN; PUNT(4); % INVALID LINK
                    END;
                END;
            END;
        B*S; S+F;

```

```

00657400
00657600
00657800
00657900
00658000
00659000
00660000
00661000
00661010
00662000
00750000
00750100
00751000
00753000
00754000
00755000
00756000
00757000
00758000
00759000
00760000
00761000
00762000
00799000
00799999
00800000
00801000
00802000
00802001
00802999
00803000
00804000
00804050
00804100
00805000
00806000
00807000
00808000
00809000
00810000
00810500
00811000
00812000
00812500
00813000
00813500
00814000
00815000
00816000
00817000
00817100
00817200
00818000
00819000
00820000
00821000
00821100
00821200
00821500
00822000

```



```

        END;
    END;
$ POP OMIT
ARRAY CT[*];
ARRAY SQ[*];
ARRAY DAT[*];
ARRAY LOGARRAY[*];
DEFINE STASUS[STASUS1]=SQ[STASUS1],[18:6]#; COMMENT
ARRAY STASUS[*]; COMMENT STASUS[I] GIVES STATUS WITH RESPECT TO SWAP
        SYSTEM OF JOB WITH MIX=I;
COMMENT POSSIBLE STATES FOR STASUS;
DEFINE TIMEND = 0#;
        WAITSWAP = 1#;
        BOJSTATE = 2#;
        SATISFY = 3#;
        EOJSTATE = 4#;
        FORCESWAP = 5#;
        TRANSIT = 6#;

        WAITSTATE = 8#;
        READYSTATE = 9#;
        RDYRPT = 10#;
        READYBOJ = 11#;
        STABLE = 56#;
        RUNNING = 16#;
        SELECTING = 32#;
REAL READYEND, FORCEND, RDYRPTEND, SWAPEND;
        COMMENT READYEND, SWAPEND, AND RDYRPTEND POINT TO LAST
        ITEM IN THEIR RESPECTIVE QUEUES;
DEFINE LINK[LINK1]=SQ[LINK1],[42:6]#; COMMENT
ARRAY LINK[*]; COMMENT LINK CONTAINS LINKS FOR READY AND SWAP QUEUES,
        LINK[I] IS MIX OF NEXT JOB IN QUEUE FOLLOWING JOB
        WITH MIX=I, LINK[0]=HEAD OF READY QUEUE, LINK[3]=
        HEAD OF SWAP QUEUE;
DEFINE SC[SC1]=SQ[SC1],[36:6]#; COMMENT
ARRAY SC[*]; COMMENT SC[I] IS FIRST CHUNK NUMBER FOR JOB WITH MIX=I;
DEFINE LC[LC1]=SQ[LC1],[30:6]#; COMMENT
ARRAY LC[*]; COMMENT LC[I] IS LAST CHUNK NUMBER FOR JOB WITH MIX=I;
DEFINE COUNT[COUNT1]=SQ[COUNT1],[24:6]#; COMMENT
ARRAY COUNT[*]; COMMENT COUNT[I]=(NUMBER OF CHUNKS "POSSESSED" (OR
        ASSIGNED) TO JOB WITH MIX=I)-1, THIS POSSESSION IS
        TO BE UNDERSTOOD DYNAMICALLY;
DEFINE SLN[SLN1]=SQ[SLN1],[15:3]#;
        COMMENT SLN[I] IS SLICE NUMBER FOR JOB WITH MIX=I;
DEFINE NLS[NLS1]=SQ[NLS1],[8:4]#;
DEFINE EXPAND[EXPAND1]=SQ[EXPAND1],[13:2]#;
        COMMENT EXPAND CONTAINS INFORMATION TO ALLOW AREAS TO EXPAND;
DEFINE CANTEXPAND[CANTEXPAND1]=SQ[CANTEXPAND1],[7:1]#;
        COMMENT MAY BE DUE TO COUNT=CHUNKMAX OR MISSING MEMORY MODS;
DEFINE MAXCORE[MAXCORE1]=SQ[MAXCORE1],[6:1]#;
        COMMENT SET BY MAXIMUM CORE CARD;
DEFINE DONTEXPANDBITS[DONTEXPANDBITS1]=SQ[DONTEXPANDBITS1],[6:2]#;
DEFINE TOTAL[TOTAL1]=CT[TOTAL1],[42:6]#; COMMENT
ARRAY TOTAL[*]; COMMENT TOTAL[I] IS TOTAL NUMBER OF JOBS USING THE
        I TH CHUNK;
DEFINE ACTIVE[ACTIVE1]=CT[ACTIVE1],[36:6]#; COMMENT
ARRAY ACTIVE[*]; COMMENT ACTIVE[I] IS NUMBER OF JOBS READY TO RUN
        USING THE I TH CHUNK;
DEFINE POSSESS[POSSESS1]=CT[POSSESS1],[30:6]#; COMMENT
ARRAY POSSESS[*]; COMMENT POSSESS[I] IS MIX INDEX OF JOB WHICH POSSESSES

```

```

00823000
00824000
00824001
00896000
00897000
00898000
00898100
00899900
00900000
00901000
00902000
00903000
00904000
00905000
00906000
00907000
00908000
00909000
00910000
00911000
00912000
00913000
00914000
00915000
00919000
00920000
00921000
00921500
00922000
00922900
00923000
00924000
00925000
00926000
00926900
00927000
00929900
00930000
00930900
00931000
00932000
00933000
00933100
00933200
00933210
00933300
00933400
00933500
00933600
00933650
00933700
00933750
00933900
00934000
00935000
00935900
00936000
00937000
00937900
00938000

```

```

        THE I TH CHUNK;                                00939000
REAL CHUNKMAX; COMMENT CHUNKMAX IS MAXIMUM ALLOWABLE CHUNK NUMBER; 00940000
DEFINE CHUNKZIZE=1024#; COMMENT CHUNKSIZE IS LENGTH OF ONE CHUNK; 00941000
DEFINE DISKSTORE[DISKSTORE1]=DAT[DISKSTORE1],[ 8:25]#; COMMENT 00941900
ARRAY DISKSTORE[*]; COMMENT DISKSTORE[I] IS ADDRESS ON DISK OF 00942000
        SWAP AREA FOR JOB WITH MIX=I; 00943000
DEFINE ACTLEN[ACTLEN1]=DAT[ACTLEN1],[33:15]#; COMMENT 00943900
ARRAY ACTLEN[*]; COMMENT ACTLEN[I] IS ACTUAL LENGTH OF STUFF SWAPPED 00944000
        FOR JOB WITH MIX=I; 00945000
        COMMENT MEM[I,*] IS AN ARRAY WHICH POINTS TO MEMORY 00954000
        SYSTEM VARIABLES (AVAIL,LEFTOFF,ETC) FOR JOB WITH 00955000
        MIX=I; 00956000
DEFINE BATCHED[BATCHED1]=CT[BATCHED1],[1:1]#, 00956200
        BATCHJOB[BATCHJOB1]=UVROW[BATCHJOB1],[7:1]#; 00956300
DEFINE MLINK1 = 0#, 00957000
        LEFTLIT = 1#, 00958000
        AVAIL = 2#, 00959000
        ADDRESSES = 3#; 00960000
        COMMENT SPACE ALLOCATED BELOW THE FENCE IS NOT SUBJECT 00961000
        TO SWAPPING, WHILE THAT ABOVE THE FENCE IS SUBJECT 00962000
        TO SWAPPING; 00963000
DEFINE UPOLAY(UPOLAY1)=CORE,[4:14]x40xUPOLAY1#, 00973000
        COMMENT = (FACTORx100)/400 x (UPOLAYx16000) 00973100
        = FACTOR/4 x (UPOLAY1 IN MICROSECS); 00973200
DOWNLAY(DOWNLAY1)=*P(DUP)=DOWNLAY1x80=20000#; %R3800974000
        COMMENT DECREASES BY AVERAGE MICROSECONDS REQUIRED 00974100
        FOR A DISK I/O OF DOWNLAY1 SEGMENTS; 00974200
SAVE PROCEDURE SWAP(STATE,B); VALUE STATE,B; REAL STATE,B; FORWARD; 00990000
PROCEDURE SHORTCOMMUNICATES;FORWARD; 00990501
PROCEDURE INITIALSWAP(N); VALUE N; REAL N; FORWARD; 00991000
PROCEDURE BRINGBACK(MIX); VALUE MIX; REAL MIX; FORWARD; 00992000
PROCEDURE REENTER(STUFF); VALUE STUFF; REAL STUFF; FORWARD; 00992100
PROCEDURE SWAPPER; FORWARD; 00992200
BOOLEAN PROCEDURE OUTWAIT(B); BOOLEAN B; FORWARD; 00993000
SAVE REAL PROCEDURE GETAREAC(N); VALUE N; REAL N; FORWARD; 00994000
SAVE PROCEDURE FORGETAREAC(N,A); VALUE N,A; REAL N,A; FORWARD; 00994500
SAVE INTEGER PROCEDURE SPACE(N); 00995000
        VALUE N; INTEGER N; 00996000
        BEGIN SPACE1=GETSPACE(N,0.5)+2; END; 00997000
$ SET OMIT = NOT(DFX) 00998000
ARRAY EUQ[10] * %DFX00998010
        @1777777777777777, 00998020
        @1777777777777777, 00998030
        @1777777777777777, 00998040
        @1777777777777777, 00998050
        @1777777777777777, 00998060
        @1777777777777777, 00998070
        @1777777777777777, 00998080
        @1777777777777777, 00998090
        @1777777777777777, 00998100
        @1777777777777777; 00998110
COMMENT EUQ IS USED WITH THE REST OF THE IO QUEUE STRUCTURE FOR %DFX00998120
        HANDLING AN EXCHANGE IN THE DISK-FILE SUBSYSTEM, IN THIS %DFX00998130
        CASE, DISK IO REQUESTS ARE HANDLED SPECIALLY, AFTER FINDING%DFX00998140
        AN IOQ SLOT, THE EU NUMBER IS RETRIEVED FROM THE DISK %DFX00998150
        ADDRESS WORD AND USED AS AN INDEX INTO EUQ, WHICH INDICATES%DFX00998160
        THE QUEUEING STATUS OF THE EU, %DFX00998170
%%%% EUQ[N] CONTAINS THE FOLLOWING FOR EU #N; %%% %DFX00998180
[33:15] = IF 1:1 = 1 THEN LU OF CONTROL CURRENTLY HANDLING EU 00998190
        IF 0 THEN TAIL OF QUEUE (IOQ INDEX) FOR THIS EU 00998200

```

```

[18:15] = HEAD OF QUEUE (IOQ INDEX),                                %DFX00998210
[ 3:15] = NEXT EU WAITING FOR A CU (EUQ INDEX),                    %DFX00998220
[ 2: 1] = IF 0, EU IS ACTIVE BUT FURTHER IO-S ARE BEING QUEUED 00998222
          SO THAT A DIFFERENT EU CAN USE THE CONTROL,              00998224
[ 1: 1] = TRUE IFF THIS EU IS IN OPERATION;                        %DFX00998230
REAL EUW; COMMENT EUW QUEUES UP EUS WAITING FOR CUS;              %DFX00998240
[33:15] = TAIL OF WAITING-EU QUEUE (EUQ INDEX)                    %DFX00998250
[18:15] = HEAD " " " " " " " " ;                                  %DFX00998260
INTEGER DISKOUNT; COMMENT NUMBER OF READY CUS;                    %DFX00998270
SAVE PROCEDURE LINKEU; FORWARD;                                    00998500
$ POP OMIT                                                         00998999
ARRAY FORKQUE[*];                                                 01000000
COMMENT THE FORKQUE IS A QUEUE OF REQUESTS TO RUN INDEPENDENT    01001000
      PROCESSES, ENTRIES ARE PUT IN THE QUEUE BY FORK, AND        01002000
      ARE INITIATED BY THE CONTROL SECTION OF THE MCP NEAR        01003000
      THE LABELS NOTHINGTODO AND SLATESTARTER, THE FORMAT        01004000
      OF THE FORKQUE IS:                                          01005000
                                                                    01006000
FORKQUE: @500777 xxxxx xxxxx                                       01007000
      THE F FIELD CONTAINS THE ADDRESS OF THE                     01008000
      FIRST WORD OF THE LAST REQUEST IN THE QUEUE,                01009000
      THE C FIELD CONTAINS THE ADDRESS OF THE                     01010000
      FIRST WORD OF THE FIRST REQUEST IN THE QUEUE,              01011000
      WHEN THERE ARE NO CURRENT REQUESTS, BOTH OF THESE          01012000
      FIELDS POINT AT THE FORKQUE,                                01013000
      THE FORMAT OF A FORKQUE ENTRY IS:                            01014000
                                                                    01015000
WORD1: @000PPP BBBB FFFF                                           01016000
      PPP IS PRIORITY +64 == USED IN LINKING IN NEW              01017000
      ENTRIES WITH A LINK-LIST-LOOKUP, IT                        01018000
      ALSO DEFINES THE INITIAL PRIORITY FOR                       01019000
      THE NEW PROCESS,                                           01020000
      BBBB IS THE LINK BACK TO THE PRECEDING ENTRY,              01021000
      FFFF IS THE LINK FORWARD TO THE NEXT ENTRY,                01022000
WORD 2: ADDRESS OF DESCRIPTOR OF ROUTINE TO BE RUN,              01023000
WORD 3: A PARAMETER TO THIS ROUTINE,                             01024000
WORD 4: SIZE OF STACK NEEDED, ZERO IF STACK                      01025000
      ALREADY OBTAINED, OR WILL USE ISTACK,                      01026000
WORD 5: ADDRESS OF STACK, IF ALREADY OBTAINED,                  01027000
      =0 => STACK SPACE NEEDED                                   01027100
      =1 => USE STACKQ TO GET SPACE IF AVAILABLE                 01027200
      (IN PARAMETER TO FORK ONLY)                                01027300
                                                                    01028000
END FORKQUE COMMENT;                                              01029000
                                                                    01030000
REAL BED;                                                         01031000
ARRAY BED1[*];                                                    01032000
COMMENT THE BED IS A LINKED LIST OF CONTROL LINES THAT ARE      01033000
      AWAITING SOMETHING, BED POINTS TO THE FIRST ENTRY,         01034000
      AND BED1 POINTS TO THE LAST ENTRY IN THE LIST,             01035000
      (NOTE: BED1 MUST BE THE NEXT PRT CELL AFTER BED,)          01036000
      THE LINKS ARE STRUCTURED SO THAT THEY MAY BE              01037000
      MANIPULATED AS ARRAYS, CONSIDERED AS AN ARRAY:           01038000
                                                                    01039000
WORD 0 CONTAINS THE LINK FORWARD TO THE NEXT ENTRY,            01040000
WORD 1 CONTAINS THE LINK BACK TO THIS ENTRY,                    01041000
WORD 2 CONTAINS THE MIX INDEX ASSOCIATED WITH THIS ENTRY      01042000
WORD 3 CONTAINS THE TIME OUT VALUE                               01043000
WORD 4 CONTAINS A LOGICAL LINE NUMBER (FOR DATACOM)            01044000
      WHICH IS ASSOCIATED WITH THE ENTRY,
WORD 5 CONTAINS THE F-REGISTER SETTING OF THE                   01045000

```

SLEEP ROUTINE (WHICH MADE THE ENTRY),	01046000
BED AND ALL FORWARD LINKS LOOK LIKE @100777PPPPPxxxxx	01047000
WHERE PPPP IS THE PRIORITY+64 FOR THIS ENTRY	01048000
AND THE BACK LINK LOOKS LIKE @500777xxxxxxxxxx,	01048100
END BED COMMENT;	01049000
REAL PRIORITY;	01050000
COMMENT PRIORITY IS THE PRIORITY ASSOCIATED WITH THE "JOB"	01051000
CURRENTLY BEING RUN ON PROCESSOR 1, IN THE SENSE	01055000
USED HERE, A "JOB" IS EITHER A USER PROGRAM OR AN	01056000
INDEPENDENT LINE OF CONTROL WITHIN THE MCP;	01057000
COMMENT P1MIX,P2MIX NOW DECLARED AT 00021700 ;	01058000
COMMENT P1MIX IS THE MIX INDEX ASSOCIATED WITH THE "JOB"	01059000
RUNNING ON PROCESSOR 1, P1MIX=0 MEANS THAT AN MCP	01060000
ROUTINE IS CURRENTLY RUNNING,	01061500
P2MIX, SIMILARLY IS THE MIX INDEX ASSOCIATED WITH THE	01062000
JOB RUNNING ON PROCESSOR 2, PROCESSOR 2 -- WHICH CAN	01063000
ONLY OPERATE IN NORMAL STATE -- IS IDLE IF P2MIX=0,	01064000
AND IS NOT PRESENT (OR SICK) IF P2MIX=-1;	01065000
REAL STACKQ;	01066000
DEFINE STANDARDSTACK =128#; % TO INCREASE FOR INDEP RUNNERS	01067000
SAVE PROCEDURE FORK(ROUTINE, PARAMETER, PRIORITY, SIZE, LOCATION);	01068000
VALUE ROUTINE, PARAMETER, PRIORITY, SIZE, LOCATION;	01069000
REAL ROUTINE, PRIORITY, SIZE;	01069100
ARRAY PARAMETER[*], LOCATION[*];	01070000
BEGIN COMMENT FORK QUEUES A REQUEST TO INITIATE AN INDEPENDENT	01071000
PROCESS IN THE LIST OF REQUESTS POINTED TO BY FORKQUE,	01072000
REQUESTS ARE ORDERED ACCORDING TO PRIORITY, AND ARE	01073000
FIRST-IN, FIRST-OUT AMONG EQUAL-PRIORITY REQUESTS,	01074000
SEE ALSO COMMENT AT FORKQUE DECLARATION;	01075000
REAL T, FRONTLINK, BACKLINK;	01076000
IF LOCATION=1 THEN	01077000
IF (LOCATION#STACKQ) NEG 0 THEN STACKQ:=M[STACKQ];	01078000
T ← GETAREA(0);	01079000
BACKLINK ← P(FORKQUE, 0&(PRIORITY+65)[9:39:9], LLL,	01079100
0, INX, ,FRONTLINK, +),[FF]);	01079200
M[T] ← FRONTLINK&BACKLINK[CTF]&(PRIORITY+64)[9:39:9];	01080000
M[FRONTLINK],[FF] ← T; M[BACKLINK],[CF] ← T;	01081000
M[T+1] ← ROUTINE; M[T+2] ← PARAMETER;	01082000
M[T+3] ← SIZE; M[T+4] ← LOCATION,[CF];	01083000
END INDEPENDENT RUNNER QUEUEING;	01084000
SAVE PROCEDURE KILL(STACK);	01085000
VALUE STACK;	01086000
ARRAY STACK[*];	01087000
BEGIN COMMENT KILL RETURNS THE STACK FOR A COMPLETED INDEPENDENT	01088000
PROCESS, AND REMOVES IT FROM THE SYSTEM;	01089000
REAL S=NT3;	01090000
P(64,STS);	01091000
IF (S+STACK,[CF])<PRT,[CF] THEN	01092000
FORGETSPACE(S)	01093000
ELSE BEGIN MIS=1]+STACKQ; STACKQ+S=1 END;	01093100
GO TO NOTHINGTODO;	01094000
END INDEPENDENT RUNNER TERMINATION;	01095000
SAVE PROCEDURE SLEEP(ADDRESS, MASK);	01095100
VALUE ADDRESS, MASK;	01095200
NAME ADDRESS;	01096000
	01097000
	01098000
	01099000
	01100000
	01101000

ARRAY MASK[*];	01102000
BEGIN COMMENT SLEEP MAKES AN ENTRY IN THE BED, IT IS USED TO WAIT	01103000
FOR AN EXOGENOUS EVENT TO OCCUR -- AN I=O OPERATION	01104000
TO FINISH, OR A TOGGLE TO BE RELEASED, FOR EXAMPLE, IT	01105000
CAN ALSO BE USED TO CLEAR INTERRUPTS (SLEEP(1,1)) OR TO	01106000
SUSPEND A PROCESS FOREVER (SLEEP(0,0)), THROUGH CALLS	01107000
ON COMPLEXSLEEP, IT IS POSSIBLE TO WAIT FOR ONE OR MORE	01108000
OF A VARIETY OF CONDITIONS TO BE SATISFIED;	01109000
REAL LINK;	01110000
ARRAY BACK = +2[*];	01111000
BAC = +2[*,*];	01112000
\$ SET OMIT = NOT(NEWLOGGING)	01112099
BOOLEAN LOGTURNEDOFF=+7; % TESTED IN NOTHINGTODO	01112100
\$ POP OMIT	01112101
P(BED,(PRIORITY+65)&P(@7777777700000,XCH)[CTF],LLL,XCH,DEL,	01113000
1,XCH,INX,LOD));	01113100
P(P1MIX,CLICK,LOGLINE);	01114000
P(O,RDF);	01115000
LINK ← BACK[0]&(PRIORITY+64)[CTF];	01116000
BAC[0,1] ← FLAG(BACK[0] ← (*P(DUP))&[LINK][CTC]);	01117000
CLICK ← @777777777777;	01118000
STOPLOG(P1MIX,1);	01119000
GO TO NOTHINGTODO;	01120000
END WAIT FOR EXTERNAL CONDITION;	01121000
SAVE PROCEDURE COMPLEXSLEEP(CODE);	01122000
REAL CODE;	01123000
BEGIN SLEEP(1,*P(.CODE)) END;	01124000
	01125000
	01126000
SAVE PROCEDURE RUN(MIX);	01127000
VALUE MIX;	01128000
REAL MIX;	01129000
BEGIN COMMENT RUN IS THE BUSINESS END OF SAVEMIX, IT RESTARTS	01130000
THE JOB WHICH SAVEMIX SAVED;	01131000
PRIORITY ← PRYOR[P1MIX ← MIX];	01132000
\$ SET OMIT = NEWLOGGING	01132999
STARTLOG(MIX,0);	01133000
\$ POP OMIT	01133001
TOGGLE ← TOGGLE OR STACKMASK;	01134000
GO TO EXTERNAL;	01135000
END REINITIATING SAVED JOB;	01136000
	01137000
SAVE PROCEDURE SAVEMIX(MIX,LOGLINE);	01138000
VALUE MIX, LOGLINE;	01139000
REAL MIX, LOGLINE;	01140000
BEGIN COMMENT SAVEMIX IS USED TO KEEP A NORMAL STATE JOB IN THE	01141000
WORKS, IT IS CALLED BY SELECTION TO INITIATE A NORMAL-	01142000
STATE JOB FOR THE FIRST TIME, AND IS USED BY HALT AND	01143000
THE P2BUSY SOFTWARE TO HANDLE THE PROBLEMS OF STOPPING	01144000
PROCESSOR 2, AND NON-EXISTENT PROCESSOR 2;	01145000
FORK(@100000XLOGLINE OR P(,RUN),MIX,-2,0,0);	01146000
\$ SET OMIT = NEWLOGGING	01146999
STOPLOG(MIX,0);	01147000
\$ POP OMIT	01147001
END MIX INDEX SAVER;	01148000
	01149000
SAVE PROCEDURE HALT;	01150000
BEGIN COMMENT IF ANYONE IS RUNNING ON PROCESSOR 2, HALT WILL	01151000
STOP HIM, HALT ALSO ALLOWS FOR THE POSSIBILITY OF	01152000
A NON-ACKNOWLEDGED PROCESSOR 2 INTERRUPT BEING FOUND,	01153000

```

        AND PERFORMS THE ESSENTIAL OPERATIONS IF NONE WAS;
        NOPROCESSTOG ← NOPROCESSTOG+1;
        IF P2MIX>0 THEN
        BEGIN P(HP2);
$ SET OMIT = NOT(NEWLOGGING)
        STOPLOG(P2MIX,0);
$ POP OMIT
        P(PRIORITY); PRIORITY ← -5;
        SLEEP(1,1); PRIORITY ← POLISH;
        IF P2MIX>0 THEN
        BEGIN SAVEMIX(P2MIX,LOGLINE2); P2MIX ← 0;
        TOGGLE ← TOGGLE AND NOT HP2MASK;
        END END END STOPPING SECOND PROCESSOR;
SAVE PROCEDURE ENTERLINEQ(ADR,LINE,PRIPTY);
        VALUE ADR,LINE,PRIPTY; REAL ADR,LINE,PRIPTY; FORWARD;
$ SET OMIT = TWXONLY
REAL
$ POP OMIT
PROCEDURE DCWRITE(ADR,LINE,SIZE);
        VALUE ADR,LINE,SIZE; REAL ADR,LINE,SIZE; FORWARD;
BOOLEAN PROCEDURE BLASTREAD(LINE,C);
        VALUE LINE,C; REAL LINE,C; FORWARD;
        SAVE PROCEDURE IOREQUEST(FINAL,IODESC,LOCATION);
        VALUE FINAL,IODESC,LOCATION;
        ARRAY FINAL,LOCATION[*]; REAL IODESC; FORWARD;
PROCEDURE OLDWIREDHAROLD; FORWARD;
PROCEDURE NOTIFYCANDE(MIX); VALUE MIX; REAL MIX; FORWARD;
PROCEDURE SYSDISKIO(IO,L,A); VALUE IO,L,A; ARRAY A[*]; REAL IO,L; %R6001164300
        FORWARD; %R6001164301
        SAVE PROCEDURE NEWIO; FORWARD;
SAVE PROCEDURE STARTIO(U); VALUE U; REAL U; FORWARD;
SAVE PROCEDURE INITIATEDCIO(IODESC,S);
        VALUE IODESC,S;
        REAL IODESC,S;
        BEGIN
$ SET OMIT = NOT(SAVERESULTS OR DEBUGGING)
        STORAWAY:=
$ POP OMIT
        IOQUE[S]:=IODESC;
        CHANNEL[P(TIO)] ← 30; %
        CHANIO[P(TIO)]:=CLOCK+P(RTR); %R5901170100
        P([IODESC],IIO);
        UNIT[30],[16:2] ← 3;
        END INITIATEDCIO;
REAL INTERROGATEMASK;
PROCEDURE NEXTDCIO;
        %IF AN INTERROGATE IS TO BE DONE IT IS HANDLED FIRST;ELSE CHECK FOR
        %READY QUEUE ENTRIES WAITING FOR DTTU AVAILABILITY, (DCREQUEST[0]
        % CONTAINS HEAD OF READY QUEUE IN CF) ALSO IF THERE IS NO SLOT IN
        % IOQUE FOR DC STUFF WILL GET ONE AND WAIT FOR A CHANNEL AS NECES-
        % SARY, CHANNEL AVAILABILITY IS GOTTEN BY OUTERBLOCK CODE WHEN THE
        % 1ST INTERROGATE IS DONE;IT MAY BE GOTTEN HERE ALSO WHEN A NEW IO
        % IS PLACED IN THE READYQUE, IOQUE SPACE IS GIVEN BACK BY IOFINISH
        % WHENEVER THERE IS NOTHING ELSE TO SEND TO DC,
BEGIN
        LABEL DOIT;
        REAL IO,D,V,S;
        P(P1MIX);
        P1MIX ← 0;
        IF UNIT[30],[15:3] < 3 THEN

```

```

01154000
01155000
01156000
01157000
01157099
01157100
01157101
01158000
01159000
01160000
01161000
01162000
01163000
01163100
01163200
01163290
01163300
01163310
01163350
01163400
01163410
01163420
01163500
01163600
01163700
01164000
01164200
01164300
01164301
01164500
01165000
01166000
01167000
01168000
01169000
01169499
01169500
01169501
01169600
01170000
01170100
01171000
01172000
01173000
01189000
01190000
01191000
01192000
01193000
01194000
01195000
01196000
01197000
01198000
01199000
01199600
01200000
01200100
01200200
01201000

```

```

IF INTRGATCTR > 0 THEN                                01202000
BEGIN %                                                01203000
    INTRGATCTR ← INTRGATCTR + 1;                       01204000
    LLNR ← 0;                                          01205000
    IOD:=INTERROGATEMASK;                             %R8601206000
    GO TO DOIT;                                       01206500
END ELSE                                              01207000
IF (LLNR+(S+STATABLE[0]),[CF])≠0 THEN                01208000
BEGIN                                                01208500
    STATABLE[0]←S&(S+STATABLE[LLNR])[40:25:8];      01209000
    STATABLE[LLNR]←S&0[25:40:8];                     01209500
    S:=LINETABLE[LLNR]:=(*P(DUP))&0[26:47:1]; % RESET READYQED 01210000
$ SET OMIT = TWXONLY                                01210495
    IF S,[21:5]=POLLING THEN %IF POLLING, KILL WITH ACT INT, 01210500
    IOD:=INTERROGATEMASK&S[9:9:9] ELSE               01211000
$ POP OMIT                                          01211005
    BEGIN                                           01211500
        IOD:=((V:=STATABLE[
$ SET OMIT = TWXONLY                                01212000
        IF S,[3:3]=MULTI THEN SEQARRAY[LLNR],[40:8] ELSE 01212995
$ POP OMIT                                          01213000
        LLNR) INX @04000000000000001)&S[9:9:9]      01213005
        &(V:=M[V])[18:5:1]&V[24:6:1];                01214000
        IF V,[5:2]=0 THEN LINETABLE[LLNR]:=S&V[21:5:5]; 01215000
    END;                                           01215500
DOIT:                                               01216000
    IF (S+UNIT[30],[FF]) GTR 1023 THEN                01216500
    IOREQUEST(0,IOD,0&30[12:42:6])                    01217000
    ELSE INITIATEDCIO(IOD,S);                         01217500
    END;                                           01218000
    PIMIX ← POLISH;                                  01218500
END NEXTDCIO;                                       01221500
SAVE PROCEDURE ENTERREADYQ(T);                       01222000
VALUE T; REAL T;                                    01223000
BEGIN                                               01223500
    LABEL ENDIT,BELOW;                                01224000
    REAL U,S;                                         01224100
$ SET OMIT = NOT TWXONLY                             01224200
    REAL LINE=T;                                      01224495
    IF (S:=LINETABLE[LINE]),[21:2]=1 THEN            01224500
    BEGIN                                           01224800
$ SET OMIT = TWXONLY                                01224900
    REAL LINE,X,V;                                    01224995
    IF (S:=LINETABLE[LINE]:=IF T GTR LMAX THEN        01225000
    STATABLE[T],LEENKER ELSE T]),[21:2]=1 THEN      01225500
    BEGIN                                           01226000
        IF S,LINEDIS = MULTI THEN                    01226500
        IF STATABLE[T]:=SEQARRAY[LINE],[40:8]),[CF]=0 THEN 01227000
        IF S,[21:5] ≠ WAITING THEN %STATION NOT WAITING TO DO MORE I/O 01227500
        BEGIN                                       01228000
            U:=T;                                     01228500
            WHILE (T:=SEQARRAY[T],LINELINK)≠U DO    01229000
            IF STATABLE[T],[CF]≠0 THEN %ANOTHER STAT,ON LINE HAS I/O, 01229500
            BEGIN                                   01230000
                SEQARRAY[LINE],[40:8]:=T;           01230500
                GO TO BELOW;                          01231000
            END;                                     01231500
            Xi=(V:=GETAREA(S,BUFSIZE))+1;           01232000
            P(1); %INITIALIZE TEST FOR ACTIVE STATIONS 01232500
            DO IF (U:=SEQARRAY[U]),[2:2]=0 THEN      01233000
            01233500

```

```

BEGIN
P(DEL,0); %INDICATE ACTIVE STATION ON LINE,
STREAM(X;U,S;X;=U,[24;1],W;=U,[25;1],V;=P(DUP)#X);
BEGIN
DI:=X;
DS:=LIT "$";S(DS:=LIT "#");
SI:=LOC U;SI:=SI+2;
DS:=CHR;W(DS:=LIT "#");
DS:=CHR;V(DS:=LIT "#");
DS:=2 LIT "P#";X:=DI; %SAVE DI FOR MOR STATIONS
DS:=2 LIT "#*";
END;
X:=P;
END UNTIL (U:=U,LINELINK)=T;
IF P THEN %NO ACTIVE STATIONS, DONT DO POLL
BEGIN
FORGETAREA(MEV],[2;2],V);
GO TO ENDIT;
END;
M[V]:=(#P(DUP))&POLLING[5;43;5]&(U:=STATABLE[T])[CTC];
STATABLE[T]:=U&V[CTC]; %LINK POLL INTO LINE QUEUE
END ELSE GO TO ENDIT;
$ POP OMIT OMIT
IF STATABLE[T],[CF] # 0 THEN
IF NOT S,READYQED THEN
BELOW: IF (U:=STATABLE[0]),[CF]#0 THEN %PUT LINE INTO READY QUEUE
BEGIN
STATABLE[0]:=U&LINE[CTF];
STATABLE[U,[FF]],LEENKER:=LINE;
END ELSE STATABLE[0]:=U&LINE[CTC]&LINE[CTF] ELSE % ALREADY IN Q
ELSE GO TO ENDIT; % NOTHING RDY
LINETABLE[LINE]:=S&(
$ SET OMIT = TWXONLY
(P(DUP),[21;5]=IDL POLLING)+P(DUP)+
$ POP OMIT
1)[21;42;6]; % SETS READYQED
END;
ENDIT;
NEXTDCIO;
END ENTERREADYQ;
PROCEDURE USASITAPE(AREA,TYPE,FROM,U,DIR); %RHR
VALUE AREA,FROM,U,DIR; REAL AREA,TYPE,FROM,U,DIR;
BEGIN REAL PTN,Y;
ARRAY ULAB[*];
LABEL EXIT,ERROR,VOL,BAD,WAIT,TIP,ETIP;
SUBROUTINE LABELSPACE;
BEGIN ULAB:=[M[SPACE(11)]]&10[8;38;10];
MOVE(10,ULAB,[CF]-1,ULAB,[CF]);
END LABELSPACE;
SUBROUTINE VOL1FILL;
BEGIN STREAM(AREA,ULAB);
BEGIN DS:=8 LIT " LABEL "; DI:=DI+1; SI:=AREA;
SI:=SI+11; IF SC=" " THEN DS:=7LIT"0" ELSE DS:=7CHR;
DI:=DI+37; %MID
SI:=AREA; SI:=SI+5; DS:=5 CHR; %PHYSICAL TAPE NO.
END;
END VOL1FILL;
SUBROUTINE HDR1CHK;
BEGIN STREAM(Y:=0;AREA,X:=0);
BEGIN DI:=LOC X; DS:=4 LIT "HDR1";

```

```

01234000
01234500
01235000
01235500
01236000
01236500
01237000
01237500
01238000
01238500
01239000
01239500
01240000
01240500
01241000
01241500
01242000
01242500
01243000
01243500
01244000
01244500
01244505
01245000
01245250
01245500
01246000
01246500
01247000
01247250
01247500
01248000
01248050
01248100
01248150
01248200
01248500
01249000
01249500
01250000
01250100
01250200
01250300
01250400
01250500
01250600
01250700
01250800
01250900
01251000
01251100
01251200
01251300
01251310
01251400
01251500
01251600
01251700
01251800
01251900

```



SI:=AREA; DI:=LOC X;	01252000
IF 4 SC=DC THEN TALLY:=1;	01252100
Y:=TALLY;	01252200
END;	01252300
Y:=P;	01252350
END HDR1CHK;	01252400
SUBROUTINE HDR1FILL;	01252500
BEGIN STREAM(AREA,ULAB);	01252600
BEGIN SI:=AREA; SI:=SI+4;	01252700
DI:=DI+17; DS:=7 CHR; %FID	01252800
SI:=SI+17; DS:=3 CHR; %REEL	01252900
SI:=SI+11; DS:=5 CHR; %C=DATE	01253000
SI:=SI-8; DS:=2 CHR; %CYCLE	01253100
SI:=SI+7; DS:=5 CHR; %P=DATE	01253200
DI:=DI+1; SI:=SI+2;	01253300
DS:=5 CHR; %BLOCK COUNT	01253400
DS:=7 CHR; %RECORD COUNT	01253500
END;	01253600
END HDR1FILL;	01253700
SUBROUTINE HARDFILL;	01253800
BEGIN PTN:=PRNTABLE[U],[30;18];	01253900
STREAM(PTN,AREA,ULAB);	01254000
BEGIN SI:=LOC PTN; DI:=DI+53;	01254100
DS:=5 DEC; DI:=ULAB; %PHYSICAL TAPE NO.	01254200
DS:=8 LIT " LABEL ";	01254300
END;	01254600
ULAB[1]:=MULTITABLE[U];	01254650
END HARDFILL;	01254700
LABELSPACE;	01254800
IF FROM=1 THEN	01254900
BEGIN VOL1FILL;	01255000
P(WAITIO(@140000005,@377,U),DEL);	01255100
P(WAITIO(AREA INX @120540000000,@377,U),DEL);	01255200
HDR1CHK;	01255300
IF Y THEN HDR1FILL ELSE GO TO ERROR;	01255400
P(WAITIO(@340000005,@55,U),DEL);	01255450
P(WAITIO(@340000005,@55,U),DEL);	01255500
GO TO WAIT;	01255600
END;	01255700
IF FROM =2 THEN	01255800
BEGIN IF TYPE=1 THEN	01255900
BEGIN VOL1FILL;	01256000
VOL: P(WAITIO(AREA INX @120540000000,@377,U),DEL);	01256100
HDR1CHK;	01256200
IF Y THEN HDR1FILL ELSE GO TO ERROR;	01256300
P(WAITIO(@340000005,@377,U),DEL);	01256400
GO TO WAIT;	01256500
END;	01256600
IF TYPE=2 THEN	01256700
BEGIN HDR1FILL;	01256800
HARDFILL;	01256900
GO TO EXIT;	01257000
END;	01257100
END;	01257200
IF FROM=3 OR FROM=4 THEN	01257300
BEGIN IF TYPE=1 THEN	01257400
BEGIN VOL1FILL;	01257500
GO TO VOL;	01257600
END;	01257700
IF TYPE=2 OR TYPE=4 THEN	01257800

BEGIN HDRIFILL;	01257900
HARDFILL;	01258000
GO TO EXIT;	01258100
END;	01258200
IF TYPE=3 OR TYPE=5 THEN	01258300
BEGIN IF DIR=0 THEN	01258400
BEGIN P(WAITIO(@340000005,@377,U),DEL);	01258500
P(WAITIO(@340000005,@377,U),DEL);	01258600
P(WAITIO(AREA INX @120540000000,@377,U),DEL);	01258700
END ELSE	01258800
P(WAITIO(AREA INX @120740000000,@377,U),DEL);	01258900
HDR1CHK;	01259000
IF Y THEN HDRIFILL ELSE GO TO ERROR;	01259100
HARDFILL;	01259200
GO TO WAIT;	01259300
END;	01259400
IF TYPE=6 THEN	01259500
BEGIN HDRIFILL;	01259600
HARDFILL;	01259700
STREAM(ULAB);	01259800
BEGIN DI:=ULAB; DI:=DI+39;	01259900
DS:=1 LIT "1";	01260000
END;	01260100
GO TO EXIT;	01260200
END;	01260300
END;	01260400
WAIT;	01260425
TIP;	01260450
IF((TWO(U) AND P(RRR)) #0) THEN	01260455
GO TO EXIT ELSE SLEEP([CLOCK], NOT CLOCK);	01260460
PTN:=PTN+1;	01260465
IF(PTN>120) THEN GO TO EXIT ELSE GO TO TIP;	01260500
P(WAITIO(@4200000000,@377,U),DEL);	01260600
STREAM(TI=TINU[U],ULAB);	01260700
BEGIN SI:=LOC T; SI:=SI+5;	01260800
DS:=LIT "#"; DS:=3 CHR;	01260900
DS:=22 LIT " INVALID USASI, RW/L*";	01261000
END;	01261100
SPOUTIT(ULAB,1); LABELTABLE[U]*@314;	01261150
TYPE*0; PTN*0;	01261160
ETIP;	01261170
IF((TWO(U) AND P(RRR)) #0) THEN	01261180
GO TO BAD ELSE SLEEP([CLOCK], NOT CLOCK);	01261200
PTN*PTN+1;	01261300
IF(PTN>120) THEN GO TO BAD ELSE GO TO ETIP;	01261400
EXIT;	01261450
MOVE(10,ULAB,[CF],AREA,[CF]);	01261500
FORGETSPACE(ULAB,[CF]);	01261999
BAD;	01262000
END USASITAPE;	01263000
\$ SET OMIT = NOT(AUXMEM)	01264000
ARRAY AUXDATA[*], AUXCODE[*];	01265000
COMMENT AUXDATA AND AUXCODE KEEP TRACK OF HOW MUCH	01266000
AUXILIARY MEMORY IS ASSIGNED TO A JOB, IN LEAGUE	01267000
WITH AUXLIMIT, THEY CONTROL THE AMOUNT USED, THEY	01268000
ALSO ALLOW THE RETURN OF THIS SPACE AFTER PROBLEMS;	01269000
INTEGER PROCEDURE AUXILIARYSPACE(SIZE);	01270000
VALUE SIZE;	01271000
INTEGER SIZE;	01272000
FORWARD;	01273000
PROCEDURE FORGETAUXILIARYSPACE(SIZE,LOC);	
VALUE SIZE, LOC;	
INTEGER SIZE, LOC;	

FORWARD;	01274000
PROCEDURE FILLORKILL(A, START, SIZE, TYPE);	01275000
VALUE A, START, SIZE, TYPE;	01275100
ARRAY A[*];	01275150
INTEGER START, SIZE;	01275200
BOOLEAN TYPE;	01275300
FORWARD;	01275400
\$ POP OMIT	01275401
REAL SCHEDULEIDS;	02015000
PROCEDURE REPORTBACK(WHY,P1,P2);	02016000
VALUE WHY,P1,P2;	02017000
REAL WHY,P1,P2;	02018000
FORWARD;	02019000
PROCEDURE MAKELOG(M,T); VALUE M,T; REAL M,T; FORWARD;	02020000
REAL KEYBOARDCOUNTER;	02020500
REAL PROCEDURE KEYIN(B); VALUE B; BOOLEAN B; FORWARD;%	02021000
DEFINE % KEYIN TABLE DEFINE VALUES FOR "REPLY"	02021200
VAX = 01#;	02021210
VIL = 02#;	02021220
VUL = 03#;	02021230
VQT = 04#;	02021240
VOU = 05#;	02021250
VWY = 06#;	02021260
VOK = 21#;	02021270
VFM = 22#;	02021280
VFR = 23#;	02021290
VOF = 24#;	02021300
VCC = 10#;	02021310
VIF = 25#;	02021320
VCT = 32#;	02021330
VTL = 34#;	02021340
BOOLEAN PROCEDURE WHYSLEEP(MASK); VALUE MASK; REAL MASK; FORWARD;%	02022000
LABEL P1PROCESS;	02023000
LABEL P2PROCESS;	02023100
REAL ONEOHONE = @101, ONEOHTWO = @102;%	02024000
REAL NUMESS;	02052100
REAL PBCOUNT;	02052200
BOOLEAN PROCEDURE OLAY(LOC,MIXX);	02052500
VALUE LOC,MIXX; REAL LOC,MIXX; FORWARD;	02052600
PROCEDURE SEEKNAM(A,B,C,D,E,N); VALUE A,B; REAL A,B,C,D,E,N; FORWARD;	02052700
PROCEDURE UNHOOQUE(MIX);%	02053000
VALUE MIX;%	02054000
INTEGER MIX;%	02055000
BEGIN%	02056000
REAL U,S,SN,T,X,I,PROCE;%	02057000
NAME OLDQ=X;	02057500
LABEL DOLP,DELINKIT;	02058000
FOR U=0 STEP 1 UNTIL 31 DO%	02059000
BEGIN%	02060000
IF(S+UNIT[U],[FF])#@77777 THEN	02061000
BEGIN%	02062000
WHILE (SN+LOCATQUE[S],[FF])#@77777 DO%	02063000
BEGIN IF (T+NFLAG(LOCATQUE[SN]),[3:5]) =%	02064000
MIX THEN%	02065000
IF LOCATQUE[SN].[11:1] THEN S+SN ELSE	02065100
BEGIN%	02066000
LOCATQUE[S]+LOCATQUE[S]&T[FTF];%	02067000
IF UNIT[U],[CF] = SN THEN	02067100
UNIT[U],[CF]+S;	02067200
RETURNIOSPACE(SN);	02068000

```

                                END ELSE% 02070000
                                S+SN;% 02071000
                                END% 02072000
                                END 02072100
                                END; 02072200
$ SET OMIT = NOT DFX; 02072490
FOR U:=0 STEP 1 UNTIL 9 DO 02072500
  IF EUQ[U]>0 THEN % EU IS NOT ACTIVE 02072600
  BEGIN % DELETE THIS JOBS IO=502072700
    OLDQ:=[EUQ[U]]; 02072800
    WHILE (S:=OLDQ[0],[FF])<1023 DO 02072900
      IF LOCATQUE[S],[3:5]=MIX THEN 02073000
      BEGIN OLDQ[0]:=(+P(DUP))&LOCATQUE[S][FTF]; 02073100
        RETURN IOSPACE(S); 02073200
      END ELSE 02073400
        OLDQ:=[LOCATQUE[S]]; 02073500
    END; 02073600
  IF (U:=T:=S:=EUW,[FF])<1023 THEN % CHECK FOR QUE EMPTIED 02073700
  DO % BY IO-S DELETED ABOVE 02073800
    IF (SN:=EUQ[S]),[FF]>1023 THEN 02073900
    BEGIN 02074000
      EUQ[S]:=SN&(NOT 0)[3:33:15]; 02074100
      IF U=T THEN EUW:=EUW&SN[18:3:15] ELSE 02074200
        EUQ[U]:=(+P(DUP))&SN[3:3:15]; 02074300
    END ELSE 02074400
      U:=S 02074500
    UNTIL (S:=SN,[3:15]) > 1023; 02074600
$ POP OMIT 02074610
DOLP; FOR U+0 STEP 1 UNTIL 31 DO% 02075000
  BEGIN% 02076000
    IF (S+(T+UNIT[U]),[FF])#077777 THEN 02077000
    BEGIN% 02078000
      IF LOCATQUE[S],[3:5]=MIX THEN% 02079000
      BEGIN% 02080000
        IF (X+T,[13:5])=0 OR X=16 OR X=31 THEN 02081000
        GO DELINKIT; 02082000
        IF X=4 THEN% 02087000
        BEGIN% 02088000
          IF LOCATQUE[S],[FF]=077777 THEN% 02089000
          BEGIN% 02090000
            I+FIRSTWAIT;% 02091000
            WHILE WAITQUE[I]#UX 02092000
            DO I + I+1 AND 31;% 02093000
            WAITQUE[I]+% 02094000
            WAITQUE[NEXTWAIT+NEXTWAIT% 02095000
              +31 AND 31];% 02096000
            UNIT[U]+T&077777[13:28:20]; 02097000
          END ELSE 02097200
            UNIT[U]:=T&LOCATQUE[S][FTF]; 02097400
          IF (U AND 076)#18 THEN 02097590
          IF UNIT[U],[FF]>1023 THEN 02097600
          BEGIN 02097800
            IF EUQ[NT4:=M[IOQUE[S]],[6:6]]<0 02098000
            THEN EUQ[NT4]:=ABS(NOT 0) ELSE 02098200
          BEGIN 02098400
            IF EUQ[NT4],[FF]<1023 THEN 02098600
            LINKEU; 02098800
            P([EUQ[NT4]],IOR); 02099000
          END; 02099200
          02099400

```

```

DELINKIT;
$ SET OMIT = NOT DFX

```

```

STARTIO(U);
END;
$ POP OMIT
RETURNIOSPACE(S);
END ELSE
PROCE*((U#23 AND U#24) OR X=3)
AND X#25 OR PROCE;
END%
END%
END ;%
IF PROCE THEN%
BEGIN%
SLEEP(1,1);PROCE*0;GO TO DOLP%
END;%
END UNHOOQUE;%
DEFINE PSF = 3:4#;
DEFINE TERMSET(TERMSET1)=PRTRW[TERMSET1],[PSF]=1#;
DEFINE NOTERMSET(NOTERMSET1)=PRTRW[NOTERMSET1],[PSF]#1#;
REAL PROCEDURE GETESPDISK;FORWARD;%
PROCEDURE DIRECTORYBUILDER(A,DDD);
VALUE A,DDD; REAL A; ARRAY DDD[*]; FORWARD;
REAL READERA,READERB;
REAL CCTBLWORD = DIRECTORYBUILDER;
DEFINE CCCOUNT = CCTBLWORD,[FF]#;
CCTBLADDR = CCTBLWORD,[CF]#;
$ SET OMIT = NOT(PACKETS)
ARRAY PSEUDO[*]; %PSEUDOMAX1
ARRAY PSEUDOMIX[*], NYLONZIPPER[*]; %MIXMAX
DEFINE PACKETPAGE[PACKETPAGE1]=PSEUDO[PACKETPAGE1],[22:26]#;
DEFINE PACKETREC[PACKETREC1]=PSEUDO[PACKETREC1],[18:3]#;
DEFINE PACKETPBD[PACKETPBD1]=PSEUDO[PACKETPBD1],[8:10]#;
DEFINE PACKETACT[PACKETACT1]=PSEUDO[PACKETACT1],[2:6]#;
DEFINE PACKETERR[PACKETERR1]=PSEUDO[PACKETERR1],[1:1]#;
DEFINE PAGESIZE=900#; % SAME AS PBDROWSZ AT 08699100
DEFINE PAGEFULL=(PAGESIZE DIV 3)*5=40#; % ALLOW FOR 8 INFO RECORDS
$ POP OMIT
REAL SPOWORD;
PROCEDURE TWXOUT(A,B,C,D); VALUE A,B,C,D; REAL A,B,C,D; FORWARD;
PROCEDURE MESSAGEWRITER;
BEGIN REAL RCW=+0;%
REAL T;
LABEL L;%

L:
IF REMOTE AND SPOWORD#0 THEN
BEGIN
TWXOUT(MESSAGEHOLDER INX 1,230,1 OR M,ABS(SPOWORD));
END;
IF SPOWORD>0 THEN
BEGIN
P(WAITIO(MESSAGEHOLDER INX 1,0,25));
P(DEL);%
END ;
NUMESS ← NUMESS-1;%
T ← MMESSAGEHOLDERJ,[18:15];
FORGETSPACE(MESSAGEHOLDER INX 1);
IF T ≠ 0 THEN%
BEGIN MESSAGEHOLDER,[33:15] ← T;%
GO TO L%

```

```

02099600
02099800
02099810
02100000
02100400
02101000
02101100
02102000
02103000
02104000
02105000
02106000
02107000
02108000
02109000
02110000
02110100
02110200
02111000
02112000
02112100
02112200
02112300
02112400
02112500
02113079
02113080
02113085
02113086
02113087
02113088
02113089
02113090
02113091
02113092
02113099
02113100
02113200
02114000
02115000
02116000
02117000
02118000
02119000
02119010
02119015
02119020
02119200
02119300
02119400
02119500
02120000
02121000
02121010
02122000
02123000
02124500
02125000
02126000
02127000

```

```

                END;%
                MESSAGEHOLDER + 0;%
                KILL([RCW] INX NOT 2);
        END;%
$ SET OMIT = PACKETS
PROCEDURE SPOUTIT(MESSAGE,TYPE);
    VALUE MESSAGE,TYPE;
    REAL MESSAGE,TYPE;
$ POP OMIT
$ SET OMIT = NOT(PACKETS)
PROCEDURE SPOUTER(MESSAGE,UNITNO,TYPE);
    VALUE MESSAGE,UNITNO,TYPE;
    REAL MESSAGE,UNITNO,TYPE;
$ POP OMIT
    BEGIN REAL MKSCH=MESSAGE-1;
    REAL S,T,MIX;
$ SET OMIT = NOT(PACKETS)
    DEFINE PACKETFREE=PSEUDO[UNITNO],[21:1]#,
        PACKETMASK=@400000000#;
    REAL PSD,PSW,Y,Z,BB;
    INTEGER NT1,R; ARRAY BUF[*];
    R:=UNITNO; UNITNO:=0;
    IF R=0 THEN IF P1MIX#0 THEN R:=PSEUDOMIX[P1MIX];
    IF R>31 AND R<36 THEN UNITNO:=R;
$ POP OMIT
    MESSAGE:=P(,MESSAGE,LOD),[33:15]-1;
$ SET OMIT = NOT(PACKETS)
    IF TYPE#64 THEN
    BEGIN
$ POP OMIT
        STREAM(A:=0;B:=MESSAGE+1);           % CHECK FOR + IN
        BEGIN S1:=B; IF SC#"+" THEN TALLY:=1; % FIRST POSITION
            A:=TALLY;
        END;
        IF P THEN MAKELOG(MESSAGE,TYPE);
$ SET OMIT = NOT(PACKETS)
    END;
    IF TYPE OR UNITNO#0 THEN
$ POP OMIT
$ SET OMIT = PACKETS
    IF TYPE THEN
$ POP OMIT
    BEGIN
        MIX := M[MESSAGE-1],[9:6];
        IF MESSAGE>FENCE THEN
        BEGIN S+M[MESSAGE-1],[CF]=MESSAGE;
            T1=GETSPACE(S-1,64,5)+1;
            MOVE(S,MESSAGE,T);
            FORGETSPACE(MESSAGE+1);
            MESSAGE+T;
        END;
$ SET OMIT = NOT(PACKETS)
    IF TYPE THEN
    BEGIN
$ POP OMIT
        IF MESSAGEHOLDER = 0 THEN%
            BEGIN MESSAGEHOLDER + MESSAGE;%
                FORK(P(,MESSAGEWRITER),0,0,90,1);
            END%
        ELSE M[MESSAGEHOLDER,[18:15]],[18:15] + MESSAGE;

```

```

02128000
02129000
02130000
02131000
02131999
02132000
02132100
02132200
02132201
02132299
02132300
02132400
02132500
02132501
02133000
02133010
02133129
02133130
02133140
02133150
02133200
02133300
02133350
02133380
02133381
02133500
02133509
02133510
02133520
02133521
02133600
02133700
02133800
02133900
02134000
02134004
02134005
02134006
02134007
02134009
02134010
02134011
02134020
02134030
02134100
02134200
02134300
02134400
02134500
02134600
02134700
02134799
02134800
02134900
02134901
02135000
02136000
02137000
02138000
02139000

```

```

M[MESSAGE] := O&MIX[4:43:5];
MESSAGEHOLDER.[18:15] ← MESSAGE;
$ SET OMIT = NOT(PACKETS)
END;
$ POP OMIT
M[MESSAGE-1].[9:6] ← 0;
IF P(MKSCW.[33:15],DUP) = 0 THEN%
BEGIN
    ;
    STREAM(N←0;X←MESSAGE+1);
    BEGIN SI ← X;
L: IF SC ≠ "+" THEN%
    BEGIN IF SC = " " THEN%
        B: BEGIN SI ← SI+1;
            IF SC = " " THEN GO TO B;
            IF SC = ALPHA THEN%
                BEGIN SI ← SI-1;
                    DS ← CHR;
            END ELSE GO TO L;
        END;
        IF SC = @14 THEN%
            BEGIN DS ← CHR;
                Q: IF SC = @14 THEN%
                    BEGIN SI ← SI+1;
                        GO TO Q;
                    END;
                GO TO L;
            END;
            DS ← CHR;
        GO TO L;
    END;
    DS ← CHR;
    N ← DI;
END;
$ SET OMIT = NOT(PACKETS)
NT1←P; NT1←((NT1.[CF]-(MESSAGE+1))×8+NT1.[30:3])×6;
END ELSE NT1←P×6;
IF UNITNO≠0 THEN IF PACKETPAGE[UNITNO=32]>1 THEN
BEGIN UNITNO:=UNITNO-32;
IF NOT PACKETFREE THEN SLEEP([PSEUDO[UNITNO]],PACKETMASK);
IF (PSD:=PACKETPAGE[UNITNO])>1 THEN
BEGIN % JUST TO BE SURE
PACKETFREE:=FALSE;
ZI:=IF (PSWI=PACKETREC[UNITNO]) THEN 60 ELSE 30;
SI:=((Y:=IF NT1>725 THEN 120 ELSE NT1 DIV 6)+7) DIV 8;
BUF:=[M[T:=GETSPACE(Z+S,64,5)+2]]&Z[8:38:10];
M[BUF-2].[9:6]:=0;
STREAM(N:=S,AA:=MESSAGE+1,BUF:=BUF INX Z);
BEGIN SI:=AA; DS:=N WDS END;
DISKWAIT(-T,Z,PSD+PSW DIV 2);
R:=(PSW×18) MOD 30;
IF (BB:=BUF[R+17],[CF]) GEQ PAGEFULL THEN
    BEGIN STREAM(BUF:=BUF[R]);
        BEGIN DS:=12LIT" ";
            DS:=28LIT"ALL FURTHER MESSAGES LOST ";
            2(DI:=DI+48); DS:=6LIT"×5908";
        END;
    END;

```

```

02140000
02141000
02141099
02141100
02141101
02142000
02143000
02143050
02144500
02145000
02146000
02147000
02148000
02149000
02150000
02151000
02152000
02153000
02154000
02155000
02156000
02157000
02158000
02159000
02160000
02161000
02162000
02163000
02164000
02165000
02166000
02167000
02168000
02169000
02170000
02171000
02172000
02173069
02173070
02173072
02173075
02173080
02173085
02173087
02173088
02173090
02173095
02173100
02173110
02173120
02173150
02173160
02173210
02173220
02173230
02173240
02173245
02173250
02173255
02173260

```

```

        PACKETPAGE[UNITNO]:=1; % TO MARK OVERFLOW          02173265
    END                                                    02173270
ELSE BEGIN P(@1540005000100000&(BB+1)[CTCJ]); % PBDSTOPPER 02173275
    IF PSW=0 THEN                                         02173280
    BEGIN P(BUF[29],XCH);                                  02173282
        P([BUF[29]],STD);                                  02173284
        DISKWAIT(T,30,PSD+5);                             02173286
        P([BUF[29]],STD);                                  02173288
    END ELSE                                              02173290
    P([BUF[R-1]],STD);                                     02173292
    BUF[R+17]:=@1540000104000000&BB[CTCJ]&              02173294
        (S+1+(M[BUF INX Z],[1:5]#">"))[8:38:10];         02173296
    STREAM(N:=S-1,CL:=S*8-Y,AA:=BUF INX Z,BUF:=[BUF[R]]); 02173300
    BEGIN 16(DS:=LIT" "); SI:=AA;                          02173305
    IF SC#">" THEN DS:=8 CHR ELSE                          02173310
    BEGIN DI:=DI-8; 8(IF SC#">" THEN DS:=CHR ELSE         02173315
        BEGIN DI:=DI+1; SI:=SI+1; END);                   02173320
    END; N(DS:=8 CHR); DI:=DI-CL; AA:=DI;                 02173325
    SI:=AA; SI:=SI-1;                                     02173330
    IF SC="+" THEN BEGIN DI:=DI-1; DS:=LIT" "; END;       02173335
    CL(DS:=LIT" ");                                       02173340
    END;END;                                              02173345
    DISKWAIT(T,Z,PSD+PSW DIV 2);                          02173350
    IF PACKETPAGE[UNITNO]>1 THEN                          02173360
    IF PSW=0 THEN                                         02173362
    BEGIN PACKETPAGE[UNITNO]:=PSD+3;                      02173364
        PACKETREC[UNITNO]:=4;                             02173366
    END ELSE                                              02173368
    PACKETREC[UNITNO]:=PSW-1;                             02173370
    PACKETFREE:=TRUE;                                     02173375
    FORGETSPACE(BUF);                                     02173380
    END; % JUST TO BE SURE                                02173383
    END;                                                  02173385
    IF NOT TYPE THEN BEGIN FORGETSPACE(MESSAGE INX 1);P(XIT); 02173389
        END;                                              02173390
$ POP OMIT                                              02173391
$ SET OMIT = PACKETS                                    02173999
    END;                                                  02174000
$ POP OMIT                                              02174001
    IF (NUMESS + NUMESS+1) > 0 THEN%                     02175000
        SLEEP([NUMESS],-0);%                             02176000
    END ELSE FORGETSPACE(MESSAGE+1);                     02176100
    END;%                                                 02177000
PROCEDURE ENDOFDECK(R); VALUE R; REAL R; FORWARD;       02177100
PROCEDURE PBIO(A,B); VALUE A; REAL A,B; FORWARD;       02178500
PROCEDURE TERMINATE(MIX); VALUE MIX; REAL MIX;%        02180000
    BEGIN IF MIX LEQ 0 THEN BYBY("MCP DS=ED+",10);     02181000
        IF PRTRW[MIX] # 0 THEN                          02181500
        BEGIN                                           02182000
            IF NOTERMSET(MIX) THEN                      02183000
            BEGIN                                       02184000
                PRTRW[MIX],[FF]*MIX,[FF];              02185000
            END;%                                       02186000
            PRTRW[MIX],[PSF]+1;                          02186100
            BRINGBACK(MIX,[CF]);                        02186200
        END;                                           02186300
    END;%                                               02187000
REAL PROCEDURE PLACEFINDER(S,A,L);                     02187100
    VALUE S,A;                                          02187200
    REAL S,A,L;                                        02187300

```



```

FORWARD;
ARRAY CIDROW[*],CIDTABLE=CIDROW[*,*];
PROCEDURE STOPCANDY; FORWARD;
PROCEDURE TERMINALMESSAG(N); VALUE N; REAL N;
  BEGIN LABEL FOUND,DOIT,OWT,TOIT;
    REAL A,T,S,ADR;%
    NAME B;%
    REAL MSTART;
    ARRAY FIB[*];
    REAL BLEN,NBUF;
  REAL MIXER,TOPIO,LUN,L;%
    INTEGER I=S; LABEL QZ;%
  LABEL STT;%
  SUBROUTINE SLAPITOFF;%
    IF LUN GEQ 32 THEN
$ SET OMIT = PACKETS
    ENDOFDECK(LUN=32)
$ POP OMIT
    ELSE
  BEGIN SLEEP([TOGGLE],STATUSMASK);
    READY + NOT (I + TWO(LUN)) AND READY;%
    RRRMECH + NOT I AND RRRMECH OR I AND SAVEWORD;%
    LABELTABLE[LUN] + @114;%
    MULTITABLE[LUN] + RDCTABLE[LUN] + 0;%
  END;%
  LABEL LB,LBI;%
    LABEL SK1,SK2,SK3;
    BOOLEAN FROMPUNT = -5;
    REAL MIXX=P1MIX,P1MIX;
$ SET OMIT = NOT(NEWLOGGING)
    STARTLOG(MIXX,0);
$ POP OMIT
    P1MIX:=MIXX;
    IF FROMPUNT THEN MIXX:=0; % GET SPACE BELOW THE FENCE
    UNLOCKTOG(TAR[P1MIX]);
    REPLY[P1MIX]+INTABLEROW[P1MIX]+0;
    PRTROW[P1MIX],[PSF]+1;
    PRIORITY+PRYOR[P1MIX]+-1;
    MSTART+MEM[P1MIX,MLINK1],[CF];
    IF N#35 THEN
    IF JAR[P1MIX,0]="CANDE " AND JAR[P1MIX,1]="TSHARER" THEN
      STOPCANDY;
    IF FROMPUNT THEN % PICK UP PUNT MESSAGE
    BEGIN STREAM(N, A:=A:=SPACE(15));
      BEGIN DS:=LIT""; SI:=N;
        63(IF SC="+" THEN JUMP OUT ELSE DS:=CHR);
        DS:=2 LIT"+";
      END;
      N:=0;
      GO TO SK1;
    END;
    A + IF N < 0 THEN ABS(N) ELSE SPACE(10);%
    B + PRT[P1MIX,4];%
    IF P(M[L+PRT[P1MIX,8],[CF]],TOP,XCH,DEL)THEN %TR02208000
      S+ADR+0 ELSE %TR02209000
DO BEGIN IF P(M[L],TOP,XCH,0,INX,ADR,+) THEN% OVERLAID RCWTR02210000
  BEGIN IF NOT M[L],[3311] THEN%NOT TYPE 13 INT 02211000
    BEGIN S+ADR; %SEGNO IN RCW 02211010
      T+0;ADR+M[M[L],MOM],[CF]; % AND THE MSCW %TR02212000
      END ELSE S+1; 02212100

```

```

END ELSE % ITS PRESENT; WEVE GOT TO WORK %TRO2213000
BEGIN T:=IF ADR>FENCE THEN MSTART ELSE 0; 02214000
      WHILE (S:=M[T],[CF]) LSS ADR DO 02215000
      IF S GTR T THEN T:=S ELSE PUNT([PUNTER[4]]); 02215500
      S+IF M[T],[3:6]=1 THEN M[T+1],[CF] ELSE 0; %TRO2216000
T+T+2; END; %TRO2216100
      IF PRT[P1MIX,8],[CF]≠L OR MEL=1],MSFF%STACK IS MARKED02216200
      THEN DO L+M[L],MOM UNTIL NOT M[L],MSFF;%GET LAST MSCW02216300
      L+M[L],MOM;%POINT L TO NEXT RCW,JUST IN CASE. %TRO2216400
END UNTIL (IF S≠0 THEN IF S=(-1) THEN 0 ELSE 02216500
              (B[0]<S OR NOT B[S],PB[1]) 02216510
              ELSE P(M[T-2],[3:6],DUP)≠7 AND P(XCH)≠13 02216600
              ) OR L=0; 02216610
FOUND; ADR ← ADR-T;% 02217000
T+PLACEFINDER(S,ADR,S); 02217100
SK1; IF N GTR 0 THEN 02217200
      BEGIN 02217300
      B ← [M[SPACE(TERMSGSZ)]]; 02218000
      DISKWAIT(=(B INX 0),TERMSGSZ,MESSAGETABLE[1],[22:26]); 02219000
      END ELSE N:=0; 02220000
      STREAM(Z:=N≠0,X:=T,T:=6,J:=[JAR[P1MIX,0]], 02221000
      P1MIX,INDX+PRT[P1MIX,8] INX NOT 2 INX 0, 02222000
      DSZE+IF P(M[P(DUP)+1],TOP) THEN P ELSE P,[8:10], 02222200
      TOG:=(N=7), Q:=[B[N]], PF:=(NOT FROMPUNT),[47:1], A);02223000
      BEGIN CI ← CI+Z; GO TO L1;% 02224000
      DS:=LIT "="; SI:=Q; 02225000
L1; SI:=SI+1; 02226000
      IF SC = "8" THEN SI:=SI+1 ELSE 02227000
      BEGIN AI=DI; DI:=LOC T; 02228000
      DS:=OCT; DI:=A; 02229000
      END; 02230000
      DS:=T CHR; 02231000
      IF TOGGLE THEN GO TO L; 02232000
      DS:=LIT"!"; GO TO L2; 02234000
L1; SI ← A;% 02235000
      IF SC ≠ "8" THEN% 02236000
      BEGIN SI ← SI+1; A ← SI;% 02237000
      GO TO L1;% 02238000
      END;% 02239000
      DI ← A;% 02240000
L2;% 02241000
      SI ← J; SI ← SI+1; DS ← 7 CHR; DS ← LIT "/";% 02242000
      SI ← SI+1; DS ← 7 CHR; DS ← LIT "=";% 02243000
      SI+LOC P1MIX; DS+2DEC; A+DI; 02244000
      DI+DI-2; DS+FILL; DI+A; 02244500
      PF(SI:=X; DS:=20 CHR; AI=DI); 02245000
      TOG(DI+A; DS+2 LIT ", "; A+DI; SI+INDX; 02251010
      SKIP SB; IF SB THEN BEGIN DI+INDX; 02251020
      SKIP DB; DS+RESET; DI+A; TOG+TALLY; 02251030
      DS+12 LIT "EFF INX IS "; END; 02251040
      A+DI; SI+INDX; DI+LOC Q; DS+8 DEC; 02251050
      SI+LOC Q; 7(IF SC>"0" THEN JUMP OUT; 02251060
      TALLY+TALLY+1; SI+SI+1); DI+A; 02251070
      T+TALLY; DS+8 CHR; DI+DI-T; 02251080
      T(DS+LIT " "); DI+DI-T; A+DI); 02251090
      TOG(SI+LOC DSZE; DI+LOC Q; DS+4 DEC; 02251100
      DI+A; DS+5 LIT " GEQ "; SI+LOC Q; 02251110
      TALLY+0; 3(IF SC>"0" THEN JUMP OUT; 02251120
      TALLY+TALLY+1; SI+SI+1); 02251130
      T+TALLY; DS+4 CHR; DI+DI-T; 02251140

```

T(DS+LIT " "); DI+DI-T; A+DI);	02251150
DI + A; DS + LIT "+";%	02252000
END;%	02253000
IF N#0 THEN FORGETSPACE(B);	02253050
IF S≥0 THEN S+S&ADR(CTF);	02253100
IF LOGLINE,[33:7] NEQ 0 THEN	02253150
BEGIN BREAK[LOGLINE,[40:8]]:=0;	02253200
REPORTBACK(DSW,S&N(8:38:10),A);	02253300
END;	02253400
S+A;	02254000
IF FROMPUNT THEN GO TO SK2;	02254050
FORGETSPACE(T);	02254100
STREAM(B:=S,A:=A:=SPACE(17));	02255000
BEGIN 17(DS+8 LIT"#"); SI+B;DI+A;DI+DI+8;DS+2 LIT" ";%	02255100
17(8(IF SC#" THEN DS+CHR ELSE JUMP OUT 2 TO L1)) ;	02255200
L1: DS+2 LIT" ";%	02255500
END;%	02256000
SK2: SPOUT(S);	02256500
IF NOT TERMGO THEN BEGIN HALT;%	02257000
COMPLEXSLEEP(-100=NUMESS);%	02258000
DO UNTIL KEYIN(0)=1;	02258100
NOPROCESSTOG + NOPROCESSTOG=1; END;%	02258200
JAR[P1MIX,1] ←-JAR[P1MIX,1];%	02259000
IF (LOGARRAY[31] AND IOMASK)=0 THEN	02259100
SLEEP([LOGARRAY[31]], IOMASK);	02259200
UNHOOQUE(P1MIX);%	02260000
IF FROMPUNT THEN BEGIN MIXX:=P1MIX; GO TO SK3 END;	02260050
MIXER ← @300+P1MIX;%	02261000
IF N=35 THEN	02261050
% ES=ED	02261100
IF (JAR[P1MIX,0] EQV "PRNPBT ") = NOT 0 THEN	02261150
IF (JAR[P1MIX,1] EQV ("DISK "))) = NOT 0 THEN	02261200
IF (L:=PRT[P1MIX,@25]) ≠ 0 THEN	02261200
BEGIN IF (LUN:=L,[38:5])<16 THEN SLAPITOFF;	02261300
LUN:=L,[43:5];	02261400
SLAPITOFF;	02261500
END; % PRNPBT/DISK ESED: TO CLEAR UNITS,	02261750
STT: T:=M[MSTART];	02262000
WHILE (L+T,[CF])≠MSTART DO	02263000
IF (T+M[L],[3:12])=MIXER AND T>0%	02264000
THEN%	02265000
BEGIN LUN ← (TOPIO ← NFLAG(M[L+2]),[12:6]);	02266000
IF LUN ≥32 THEN	02266100
BEGIN	02266200
FILECLOSE(TOPIO INX 0);	02266300
GO TO STT;%	02266400
END;	02266500
IF UNIT[LUN],[13:5] = @20	02267000
THEN BEGIN%	02268000
QZ:%	02269000
SLAPITOFF;	02270000
UNIT[LUN],[13:5]=@20;% MARK IT NOT READY ANYWAYS	02270500
FORGETSPACE(L INX 2);%	02271000
GO TO STT;%	02272000
END ELSE	02273000
BEGIN T ← 0;	02274000
FIB ← M[TOPIO INX NOT 2];	02275000
ADR ← NBUF ← FIB[13],[119]=1;	02275100
IF P(M[TOPIO],[315],DUP)=22 OR P(XCH)=26 THEN	02275150
BEGIN FOR S ← 1 STEP 1 UNTIL ADR DO	02275200
TOIT: IF NOT M[TOPIO INX S],[19:1] THEN	02275250

```

DOIT: IF LUN≤18 THEN                                02275300
      BEGIN M[TOPIO INX S],[20:11] ← 0;              02275350
            M[M[TOPIO INX S] INX 17] ← M[TOPIO INX S] 02275400
              & FIB[5] [FTC];                          02275450
            FIB[5] ← P(DUP,L0D,0,1,CFX,+);          02275500
            IF NOT PR1ROW[P1MIX],[7:1] THEN          02275550
              IF FIB[14],[CF]=FIB[14],[FF] THEN    02275600
                BEGIN PB10(TOPIO INX S,FIB[14]);    02275650
                  SLEEP([M[TOPIO INX S]],IOMASK); 02275700
                END ELSE                              02275750
                BEGIN STREAM(C+M[TOPIO INX S],      02275800
                              Z+FIB[14],[FF]);    02275850
                  BEGIN SI ← C; DS ← 18 WDS; END; 02275900
                  FIB[14],[FF] ← P(DUP),[FF]=18; 02275950
                END;                                  02276000
              END ELSE                                02276050
              BEGIN IF WAITIO(M[TOPIO INX S],@357,LUN),[45:1] 02276100
                    THEN GO OWT;                    02276150
                    FIB[6] ← *P(DUP)+1;             02276200
              END;                                    02276250
            IF ADR<0 THEN                             02276260
              BEGIN IF ADR THEN FIB[17] ← BLEN; GO OWT; 02276270
                END;                                  02276280
            S ← 0;                                     02276290
            IF FIB[17] < (BLEN+FIB[18],[3:15]) THEN 02276300
              BEGIN IF NOT FIB[13] THEN              02276350
                FIB[17] ← *P(DUP)-(FIB[5],[46:2]=3); 02276360
                M[TOPIO] ← FLAG(FIB[16]);            02276370
                STREAM(N+FIB[17],D+M[TOPIO],[CF]); 02276400
                BEGIN N(DS ← 8 LIT " "); END;        02276450
                ADR ← -1; GO DOIT;                   02276500
              END ELSE ADR ← -2;                       02276550
            GO TOIT;                                   02276600
          END ELSE                                    02276700
OWT: FOR NT1 ← 0 STEP 1 UNTIL NBUF DO                 02276750
      M[TOPIO INX NT1] ← *P(DUP) OR IOMASK;%          02277000
      IF LUN≤22 AND LUN≥20 OR (LUN≤18 AND           02278000
      P(M[TOPIO],[3:5],DUP)=22 OR P(XCH)=10))       02278100
      THEN                                            02278500
        BEGIN IF LUN ≤ 18 THEN % UNIT IS BACKUP     02279000
          BEGIN S+17;%                                02280000
            STREAM(A,D+L+4);                          02281000
            BEGIN SI+A; DS+17 WDS END;%              02282000
            NT4+M[TOPIO INX NOT 2] INX 0;%           02283000
            NT1+M[NT4+14];%                           02284000
            NT2+NT1,[FF]; NT1+NT1,[CF];%            02285000
            IF M[TOPIO],[3:5]=22 THEN                 02285100
              IF NT1=NT2=72 THEN%                     02286000
                BEGIN NT1+M[NT4+5],[FF];%            02287000
                  M[NT4+5],[FF]←NT1+1;%             02287100
                  M[NT2+17]← @1540004002000000 &NT1[CTC];% 02287110
                  M[NT4+14],[FF]←NT2-18;%           02287120
                END ELSE%                              02287130
                  IF M[NT2+35],[27:6]=0 THEN M[NT2+35],[28:1]←1; 02287140
                  FIB[17] ← -1;                       02287200
                  M[TOPIO] ← FLAG(FIB[16]&0[20:47:1]&S[8:38:10]); 02287210
                END ELSE %                             02287230
                BEGIN T+(A INX @540000000000000) &(LUN#22)[32:47:1]% 02287240
                  &17[8:38:10];%                     02287250
              IF LUN#22 THEN %IF PUNCH FILE, IGNORE 02287254

```

```

                IF WAITIO(@4002000000,@357,LUN),[45:1] THEN GO QZ;
                T←WAITIO(T,@357,LUN);%
                IF T,[45:1] THEN GO TO QZ;%
            END;
        END ELSE%
            IF LUN=23 OR LUN=24 THEN%
                BEGIN ADR←L+4;%
                LB: IF(T←UNIT[LUN]),[13:5]=25 THEN%
                    BEGIN ADR←T,[CF];
                    STREAM (A←"END";ADR); BEGIN SI ← ADR;%
                    L←SI ← SI +1; IF SC = " " THEN GO TO L;%
                $ SET OMIT = PACKETS
                    DI ← LOC A; DI ← DI+5; IF 3 SC ≠ DC THEN TALLY ← 1; A ←%
                $ POP OMIT
                $ SET OMIT = NOT(PACKETS)
                    DI←LOC A;DI←DI+5; IF 3SC=DC THEN TALLY←0 ELSE
                    BEGIN DI←LOC A; DS←4 LIT "PACK"; DI←LOC A;
                    SI←SI-3; IF 4SC=DC THEN TALLY←0 ELSE
                    TALLY←1 END; A:=
                $ POP OMIT
                TALLY END; IF P THEN BEGIN%
                    UNIT[LUN]←@7777777777%
                END
            ELSE BEGIN M[TOPIO]←M[TOPIO]OR@2004000000; T←0;%
                M[M[TOPIO]]←"END, "&@14[1:43:5]; END;%
                END;
                IF T≠0 THEN%
                    BEGIN%
                        LBI←T←WAITIO(@400000000+ADR,@367,LUN);%
                        IF T,[45:1] THEN GO TO QZ;%
                        IF T,[42:1] THEN GO TO LB ELSE%
                            GO TO LBI%
                    END
                END;%
                IF T=0 THEN
                    IF FIB[5],[42:1]
                        THEN FORGETSPACE(L INX 2)
                        ELSE FILECLOSE(TOPIO INX 0);
                GO TO STT
            END; END;
            FORGETSPACE(A);%
            TI←M[MSTART];
            MIXER←@400+P1MIX;
            WHILE (L←T,[CF])≠MSTART DO
                IF(T←M[L]),[3:12]=MIXER AND T>0 THEN%
                    IF M[M[L+4],[CF]+5],[41:1] THEN FILECLOSE(L+7);
                T←M[MSTART];
                MIXER←@600+P1MIX;
                WHILE (L←T,[CF])≠MSTART DO
                    IF(T←M[L]),[3:12]=MIXER AND T>0 THEN%
                        IF M[L+7],[41:1] THEN FILECLOSE(M[L+1] INX 3);%
                FOR LUN←0 STEP 1 UNTIL 31 DO
                    IF RDCTABLE[LUN],[8:6] = P1MIX THEN%
                        SLAPITOFF;%
                PRT[P1MIX,9]←5; %SET FOR GETSPACE
                P(,COM5); GO TO DIFFCOM;
            END;%
        SAVE PROCEDURE TERMINALMESSAGE(N); VALUE N; REAL N;
        BEGIN NT1 ← N;

```

```

02287255
02287260
02287270
02287280
02290000
02291000
02292000
02293000
02294000
02295000
02296000
02296999
02297000
02297001
02297009
02297010
02297100
02297200
02297300
02297301
02298000
02299000
02300000
02301000
02302000
02303000
02304000
02305000
02306000
02307000
02308000
02309000
02310000
02311000
02312000
02313000
02313500
02313600
02314000
02315000
02316000
02317000
02318000
02318050
02319000
02320000
02321000
02322000
02322050
02323000
02324000
02325000
02326000
02327000
02328000
02328500
02329000
02330000
02330100
02330200

```

SK31

P(0,STF);	02330300
TERMINALMESSAGA(NT1);	02330400
END;	02330500
ARRAY UNITCODE[*];	02347100
BOOLEAN PROCEDURE READEMFROMDISK(H,IB);	02347150
VALUE H,IB; ARRAY H[*],IB[*]; FORWARD;	02347160
\$ SET OMIT = NOT(PACKETS)	02347199
PROCEDURE DRAIN0(UNIT,BUMP,ERROR);	02347200
VALUE UNIT,BUMP,ERROR; REAL UNIT; BOOLEAN BUMP,ERROR;	02347210
BEGIN REAL T;	02347220
LABEL NEXT;	02347222
UNIT←UNIT-32;	02347230
IF BUMP THEN	02347240
PACKETACT[UNIT]:=PACKETACT[UNIT]-1;	02347250
IF ERROR THEN PACKETERR[UNIT]:=TRUE;	02347260
IF PACKETACT[UNIT]=0 THEN	02347280
IF LABELTABLE[UNIT+32]≥0 THEN	02347290
IF CIDTABLE[UNIT,3]<CIDTABLE[UNIT,7] THEN	02347300
BEGIN	02347310
LABELTABLE[UNIT+32]←-@14;	02347315
T:=GETSPACE(13,64,5)+4;	02347320
NEXT:  DO UNTIL READEMFROMDISK(CIDROW[UNIT],	02347330
[M[T]]&10[8:38:10]);	02347335
IF PACKETERR[UNIT] THEN BEGIN;	02347340
STREAM(E+"END"; Q=@14,D+T);	02347350
BEGIN SI←LOC Q; SI←SI+7; IF SC≠DC THEN DI←DI+1;	02347360
Q←DI; SI←Q;	02347370
L:  IF SC=" " THEN BEGIN SI←SI+1; GO TO L END;	02347380
DI←LOC E; DI←DI+5; IF 3 SC≠DC THEN TALLY+1;	02347390
E←TALLY; END;	02347400
IF P THEN GO TO NEXT; END;	02347410
M[T INX 10]←UNITCODE[UNIT+9];	02347420
FREECARD(T&(UNIT+32)[2:42:6]&ERROR[1:1:1]);	02347430
END ELSE	02347440
ENDOFDECK(UNIT&ERROR[1:1:1]);	02347450
END DRAIN0;	02347460
\$ POP OMIT	02347461
REAL STREAM PROCEDURE UNITIN(TINU,WHAT); VALUE WHAT;%	02348000
BEGIN%	02349000
SI ← WHAT;%	02350000
L:  IF SC = " " THEN%	02351000
BEGIN SI ← SI+1; GO TO L END;%	02352000
DI ← TINU;%	02353000
\$ SET OMIT = SHAREDISK	02353999
37(DI ← DI+5)	02354000
\$ POP OMIT	02354001
\$ SET OMIT = NOT(SHAREDISK)	02354099
41(DI:=DI+5)	02354100
\$ POP OMIT	02354101
IF 3 SC = DC THEN JUMP OUT;%	02355000
TALLY ← TALLY+1;%	02356000
SI ← SI-3));%	02357000
UNITIN ← TALLY;%	02358000
END;%	02359000
DEFINE ENTERUSERFILE(ENTERUSERFILE1,ENTERUSERFILE2,ENTERUSERFILE3)=	02378000
P(EUF(ENTERUSERFILE1,ENTERUSERFILE2,ENTERUSERFILE3),DEL)#;	02378500
REAL PROCEDURE EUF(A,B,L); VALUE A,B,L; REAL A,B,L; FORWARD;	02379000
INTEGER PROCEDURE CALCULATEPURGE(PURGE);%	02380000
VALUE PURGE; REAL PURGE;%	02381000
BEGIN REAL Y,D;%	02382000

REAL J; %	02383000
REAL C=+1; %	02384000
STREAM(A+[DATE],B+[Y]); %	02385000
BEGIN SI=A; SI=SI+3; DS + 2 OCT; DS + 3 OCT END; %	02386000
J + (D + (Y+3) DIV 4x1461+(Y+3) MOD 4 x 365 +D+PURGE=	02387000
1) DIV 1461; %	02388000
IF (Y + (D + D MOD 1461) DIV 365) = 4 THEN %	02389000
BEGIN Y + 3; D + 365 END ELSE D + D MOD 365; %	02390000
CALCULATEPURGE + (4xJ+Y=3)x1000+D+1; %	02391000
STREAM(C+[C]); BEGIN SI=C; DS + 8 DEC END; %	02392000
END; %	02393000
PROCEDURE CHANGEDATE(BUFF); VALUE BUFF; REAL BUFF; FORWARD;	02393100
DEFINE MIDNIGHT = BEGIN XCLOCK:=XCLOCK=WITCHINGHOUR;	02393200
DATE:=CALCULATEPURGE(1);	02393225
CHANGEDATE(SPACE(10));	02393250
END#;	02393300
REAL PROCEDURE TAPELABEL(M,F,R,C,P); VALUE M,F,R,C,P;	%A102393400
REAL M,F,R,C,P; FORWARD;	%A102393500
\$ SET OMIT = NOT(DUMP OR DEBUGGING)	02393999
PROCEDURE PRINTCORE(X); VALUE X; REAL X;	%10302394000
BEGIN REAL B,S,N,I,K; ARRAY T[*]; %	02395000
LABEL L1,L2;	%A102395050
REAL U,MS;	%10302395100
LABEL NW,PR,SK;	02395150
DO UNTIL ( U +	02395200
IF LABELTABLE[20] = 0 THEN 20 ELSE	02395300
IF LABELTABLE[21] = 0 THEN 21 ELSE	02395400
IF P(RRR),[27:1] THEN 20 ELSE	02395500
IF P(RRR),[26:1] THEN 21 ELSE 0) # 0;	02395600
S + 0;	02396000
HALT;	02396100
WAITSTORE(0);	02396110
STOREDY[0]=0;	02396120
WHILE (S + M[S]),[33:15] # 0 DO	02396130
IF M[S],[1:17] = @1000 THEN	02396200
B + OLAY(S,[33:15],0);	02396300
STOREDY[0]=1;	02396400
\$ SET OMIT = NOT(DEBUGGING) OR OMIT	02396500
B+(DBARRAY INX 2)&15[8:38:10];	02396599
\$ POP OMIT	02397000
\$ SET OMIT = DEBUGGING OR OMIT	02397001
B:=SPACE(30)&15[8:38:10];	02397499
\$ POP OMIT	02397500
L1: IF P(WAITIO(@4000100000,4,U)),[45:1] THEN GO TO L1;	02397501
IF X GTR MIXMAX OR X LSS 0 THEN X:=0;	%A102397800
MS:=SI=MEMROW[X],[CF];	%10302398000
IF STATUS[X]=RUNNING OR S=0 THEN	%10302398100
	%10302398200
	02398500
DO BEGIN N:=IF (T:=M[S]) GTR 0 AND T,[CF] NEQ MS	%10302399000
THEN	%10302399100
NOT S INX T INX 1 ELSE 3; %	02400000
IF (S AND @7777) = @7777 AND (T AND @7777) = 0 THEN %	02401000
N + 1; %	02402000
NW: STREAM(N+IF N>6 THEN 6 ELSE N,S,B); %	02403000
BEGIN 60(DS + 2 LIT " "); %	02404000
DI + B; SI + LOC S; SKIP 33 SB; %	02405000
5(DS+3RESET; 3(IF SB THEN %	02406000
DS + SET ELSE DS + RESET; SKIP SB)); %	02407000
DS + LIT " "; %	02408000

```

SI * S;% 02409000
N(CDS * LIT " "; 2(CDS*LIT" "); 8(CDS*3RESET;% 02410000
3(IF SB THEN DS*SET ELSE DS * RESET;% 02411000
SKIP SB))));% 02412000
END;% 02413000
K * 1;% 02414000
PRI I * WAITIO(B*((N>6)+1)[27:46:2],@64,U); 02415000
IF I.[42:1] THEN% 02416000
I * WAITIO(@4000100000,4,U); 02417000
IF I.[45:1] THEN GO TO PR;% 02418000
SKI S * S+6; N * N+6;% 02419000
IF N ≥ 6 THEN% 02420000
BEGIN; STREAM(A+S-6;S);% 02421000
BEGIN SI * A; IF 48 SC = DC THEN TALLY * 1; 02422000
A * TALLY;% 02423000
END;% 02424000
IF P THEN% 02425000
BEGIN IF K THEN BEGIN;% 02426000
STREAM(B); 60(CDS*2LIT"*");% 02427000
K * 0;% 02428000
GO TO PR;% 02429000
END ELSE GO TO SK; END;END;% 02430000
IF N > 0 THEN GO TO NW;% 02431000
S * T.[33:15];% 02432000
END UNTIL S=MS; %10302433000
NOPROCESSTOG * NOPROCESSTOG-1; 02433100
L2: IF P(WAITIO(@4000100000,4,U)).[45:1] THEN GO TO L2; %AI02433500
END;% 02434000
REAL MEMOD; %AI02434050
PROCEDURE DUMPCORE(BUFF); %AI02434100
VALUE BUFF; REAL BUFF; %AI02434110
BEGIN REAL B,S,N,TM,TA,U,D; %AI02434120
INTEGER I; %AI02434125
ARRAY TP[*]; ARRAY TL[*]; %AI02434130
LABEL X,L1; %AI02434135
BOOLEAN SUBROUTINE CHKMOD; %AI02434140
BEGIN; %AI02434142
STREAM(N;MM*MEMOD); %AI02434144
BEGIN SI*LOC MM; SKIP 40 SB; SKIP N SB; 02434146
IF SB THEN TALLY*1; N*TALLY; %AI02434148
END; %AI02434150
CHKMOD * P; %AI02434155
END; %AI02434160
FOR U:=0 STEP 1 UNTIL 15 DO %AI02434170
IF (MULTITABLE[U]="MEMORY ") AND %AI02434180
(LABELTABLE[U],[5:25]="1DUMP") THEN GO TO L1; %AI02434190
FOR U=0 STEP 1 UNTIL 15 DO IF LABELTABLE[U]=0 %AI02434200
AND PRNTABLE[U],[1:1] THEN GO TO L1; %AI02434210
BUFF:=BUFF.[15:15]-1; %AI02434215
STREAM(BUFF); %AI02434220
DS:=17LIT"#NO MEMDUMP TAPE*"; %AI02434230
GO TO X; %AI02434240
L1: MULTITABLE[U]="MEMORY "; %AI02434250
LABELTABLE[U],[1:29]:=@1024644447; %AI02434260
STREAM(A:"001",B:[LABELTABLE[U]]); %AI02434270
BEGIN SI := LOC A; SI := SI + 5; %AI02434280
DI:=DI+5; DS:=3ADD; %AI02434290
END; %AI02434300
RRRMECH := TWO(U) OR RRRMECH; %AI02434310
BI:=GETSPACE(20,0,0)+2; %AI02434320

```



```

STREAM(LTT+BUFF,[33:15]<100,BUFF+BUFF,[33:15],B); 02434330
BEGIN %AI02434340
  DS:=8LIT" "; SI:=B; DS:=19WDS; %AI02434350
  DI + B; 02434360
  LTT(SI + LOC BUFF; DS + 2 DEC; JUMP OUT 1 TO L); 02434365
  SI + BUFF; 02434367
  20(8(IF SC#"+ " THEN DS+CHR ELSE JUMP OUT 2 %AI02434370
    TO L)); L; %AI02434380
END; %AI02434390
LABELTABLE[U],[1:5]:=020; %AI02434400
TL:=[M[TAPELABEL("MEMORY ",LABELTABLE[U],[6:42], %AI02434410
  1,1,10)]]&10[8:38:10]; %AI02434420
STREAM(A+PRNTABLE[U],[30:18],TL); 02434424
  BEGIN SI+LOC A; DI+DI+53; DS+5 DEC END; %AI02434426
TP:=[M[TA:=GETSPACE(513,0,0)+2]]&513[8:38:10]; %AI02434430
TM:=0&@1737[1:37:11]; %AI02434440
P(WAITIO(TL&@05000[CTF],0,U),DEL); %AI02434450
P(WAITIO(TM,@40,U),DEL); %AI02434460
S:=0; %AI02434470
HALT; WAITSTORE(0); %AI02434480
STOREDY[0]+0; %AI02434490
WHILE (S:=M[S]),[33:15] NEQ 0 DO %AI02434500
  IF M[S],[1:17]=@1000 THEN %AI02434510
    D:=OLAY(S,[33:15],0); %AI02434520
STOREDY[0]+1; %AI02434530
S:=0; %AI02434540
DO BEGIN %AI02434550
  N:=S,[33:3]; %AI02434560
  IF CHKMOD THEN S := =S %AI02434570
    ELSE MOVE(512,S,TA+1); %AI02434580
  TP[0] := S; %AI02434590
  P(WAITIO(TP&@05000[CTF],0,U),DEL); %AI02434600
  IF S LSS 0 THEN S := 3584 - S; %AI02434610
  END UNTIL (S:=S+512),[18:15]; %AI02434620
P(WAITIO(B&20[8:38:10]&5[21:45:13],0,U),DEL); %AI02434630
P(WAITIO(TM,@40,U),DEL); %AI02434640
P(WAITIO(TL&@05000[CTF],0,U),DEL); %AI02434650
P(WAITIO(TM,@40,U),DEL); 02434652
P(WAITIO(@4740000020,@377,U),DEL); 02434654
FORGETSPACE(TP); %AI02434660
FORGETSPACE(TL); %AI02434670
FORGETSPACE(B); %AI02434680
LABELTABLE[U],[1:5]+@01; %AI02434690
BUFF:=BUFF,[15:15]-1; %AI02434695
STREAM(U+TINU[U],L+LABELTABLE[U],BUFF); %AI02434700
BEGIN %AI02434710
  SI:=LOC U; SI := SI + 5; %AI02434720
  DS:=1LIT" "; DS:=3CHR; %AI02434730
  SI+LOC L; SI+SI+1; DS+ 1 LIT " "; DS+7 CHR; %AI02434735
  DS:=7LIT" DP=ED+"; %AI02434740
END; %AI02434750
NOPROCESSTOG:=NOPROCESSTOG-1; %AI02434760
X; SPOUT(BUFF); %AI02434770
END DUMPCORE; %AI02434780
$ POP OMIT 02434781
PROCEDURE NAMEID(A,KTR);% 02603000
  REAL A,KTR;% 02604000
  BEGIN;% 02605000
    STREAM(A+[A];KTR);% 02606000
    BEGIN DI + A; DS + 8 LIT "0 ";% 02607000

```

DI ← DI-7; SI ← KTR;%	02608000
L1: IF SC = " " THEN%	02609000
BEGIN SI ← SI+1; GO TO L END;%	02610000
IF SC = "" THEN%	02611000
BEGIN SI ← SI+1;%	02612000
7(IF SC = "+" THEN JUMP OUT TO EXIT;%	02613000
DS ← CHR;%	02614000
IF SC = "" THEN JUMP OUT TO LQ;);%	02615000
LQ:    SI ← SI+1;%	02616000
GO TO EXIT;%	02617000
END;%	02618000
IF SC = ALPHA THEN%	02619000
BEGIN 7(DS ← CHR;%	02620000
IF SC = ALPHA THEN GO TO LA;%	02621000
JUMP OUT TO EXIT;%	02622000
);%	02623000
LA:  );%	02623500
LE:  IF SC = ALPHA THEN %	02623501
BEGIN SI←SI+1; GO TO LE; END; %	02624000
GO TO EXIT;%	02625000
END;%	02626000
IF SC = "+" THEN%	02627000
BEGIN DS ← CHR; SI ← SI-1; GO TO EXIT END;%	02628000
IF SC = "=" THEN%	02629000
BEGIN DS←2 LIT"←"; SI←SI+1; GO TO EXIT END;	02630000
DS ← CHR;%	02631000
EXIT: A ← SI;%	02632000
END;%	02633000
KTR ← P(XCH);%	02634000
END;%	02635000
REAL PROCEDURE TAPELABEL(MULFID,FID,REELNO,CYCLE,PURGE);%	02636000
VALUE MULFID,FID,REELNO,CYCLE,PURGE;%	02637000
REAL MULFID,FID,REELNO,CYCLE,PURGE;%	02638000
REAL LBL;%	02639000
LBL := SPACE(10);	02640000
STREAM(%	02641000
DATE, MULFID,FID,REELNO,CYCLE,PU←CALCULATEPURGE(PURGE),%	02642000
LBL);%	02643000
BEGIN%	02644000
DS←8 LIT" LABEL " ;%	02645000
SI←LOC MULFID;%	02646000
DS←WDS;%	02647000
DS←WDS;%	02648000
DS←3 DEC;%	02649000
SI ← LOC DATE; SI ← SI+3;%	02650000
DS ← 5 CHR;%	02651000
SI←LOC CYCLE;%	02652000
DS← 2 DEC;%	02653000
SI←LOC PU; SI←SI+3;%	02654000
DS←5 CHR; DS←1 LIT"0";%	02655000
5(DS←8 LIT"00000000");%	02656000
END;%	02657000
TAPELABEL←LBL;%	02658000
END;%	02659000
REAL PROCEDURE LABELSCRATCH(LBL); VALUE LBL;REAL LBL;%	02660000
BEGIN%	02661000
REAL LUN, TM, REEL, T;	02662000
LBL ← P(,LBL,LOD),[33,15];%	02662100
STREAM(L←LBL+3,R←[REEL]);	02662200
BEGIN SI←L; DS←3 OCT END;	02663000
LUN←FINDOUTPUT(M[LBL+1],M[LBL+2],2,0,REEL,0,0, TM);	

```

IF LUN ≥ 0 THEN
BEGIN;
STREAM(A+PRNTABLE[LUN],[30:18],T+[T],L+LBL+6);
BEGIN DI+DI+5; SI+LOC A; DS+5DEC; SI+SI+8; DI+T;
DS+8DEC; DI+DI+7; DS+6FILL; END;
RDCTABLE[LUN],[8:6]+P1MIX;
MULTITABLE[LUN]+M[LBL+1];
RRRMECH+TWO(LUN) OR RRRMECH;
P(WAITIO(LBL OR @12050000000,0,LUN),DEL);%
TM+0&"2+"[1:37:11];%
P(WAITIO([TM],0,LUN),DEL);%
FILEMESSAGE(" OUT"&TINU[LUN][6:30:18],T,
M[LBL+1],M[LBL+2],REEL,0,0,OPNMESS);
END;
LABELASCRATCH=LUN%
END LABELASCRATCH;%
PROCEDURE NSECOND;FORWARD;%
BOOLEAN PROCEDURE SYSTEMFILE(A,B); VALUE A,B;REAL A,B; FORWARD;
DEFINE CHECKSTACKSPACE=IF P(PRT[P1MIX,*] INX 0)=P(0,RDS) LSS
63&(CANDYINX NEQ P1MIX)[41:47:1]
THEN BEGIN P(64,STS); GO TO STACKOVERFLOW; END#; %WF
REAL PROCEDURE PUTORTAKE(MIX,WHERE,IO,WHAT);
VALUE MIX,WHERE,IO,WHAT;
REAL MIX,WHERE,IO,WHAT;
COMMENT THIS ROUTINE IS USED TO GET OR RETURN ONE OR MORE
WORDS FROM A (PERHAPS) SWAPPED AREA. THIS
ROUTINE IS SMART ENOUGH TO NOT DO DISK ACCESSES
UNLESS IT NEEDS TO, TOO. IF ONE WORD IS GOTTEN,
IT IS RETURNED AS THE VALUE. IF MORE THAN ONE
WORD IS GOTTEN, THE VALUE RETURNED IS A POINTER
TO THE AREA CONTAINING THE WORDS. THE ROUTINE
DOES GETSPACE/FORGETSPACE ACTION ON MULTIPLE=WORD
AREAS, OF COURSE INPUT VERSUS OUTPUT DECIDES;
BEGIN INTEGER S,COUNT,SIZE;
REAL DA,T,NORTH;
IF (STATUS[MIX] AND STABLE) = 0 THEN
BEGIN IF IO,[2:1] THEN CLICK:=CLOCK+P(RTR)+180;
SLEEP([SQ[MIX]],0&STABLE[18:42:6]);
IF (STATUS[MIX] AND STABLE)=0 THEN P(0,NOT,RTN);
END;
SQ[MIX]+=*P(DUP);
COUNT + ABS(IO)+(IO=0);
IO + IO>0;
IF NORTH+(S+[WHERE],[CF]) ≥ FENCE AND((T+STATUS[MIX])=
READYSTATE OR T=RDYRPT OR T=WAITSTATE)
THEN BEGIN T := SPACE(SIZE := 30*((COUNT+58)DIV 30));
DA+DISKSTORE[MIX]+(S+([MEM[MIX,MLINK1]]
INX 1)+S) DIV 30;
S+S MOD 30;
DISKWAIT(-T,SIZE,DA);
WHERE+[M[T+S]];
END;
IF COUNT#1 THEN
BEGIN PUTORTAKE+NFLAG(WHERE);
IF NOT IO THEN M[WHERE] + WHAT;
END ELSE BEGIN
IF IO THEN WHAT := SPACE(COUNT);
P([WHERE],WHAT);
IF NOT IO THEN P(XCH);
STREAM(S+P,D+P:SIZE+COUNT);

```

```

02663100
02663200
02664000
02664100
02665000
02665100
02665150
02665200
02666000
02667000
02668000
02668500
02668600
02668800
02669000
02670000
02692000
02692500
02693000
02693500
02694000
02700000
02701000
02702000
02702100
02702200
02702300
02702400
02702500
02702600
02702700
02702800
02702900
02703000
02704000
02704100
02704120
02704140
02704160
02704180
02704200
02704300
02704400
02705000
02706000
02707000
02708000
02709000
02710000
02711000
02712000
02713000
02714000
02714100
02714200
02714300
02715000
02715100
02715200
02716000

```

```

                BEGIN SI+S; DI+D; DS+SIZE WDS; END;          02716100
                P(DEL,DEL);                                  02716200
                PUTORTAKE+WHAT;                               02716300
            END;                                             02717000
            IF NORTH THEN                                    02718000
            BEGIN IF NOT IO THEN DISKWAIT(T,SIZE,DA);        02719000
                FORGETSPACE(T);                               02719100
            END;                                             02719200
            IF NOT IO THEN IF COUNT#1 THEN FORGETSPACE(WHAT); 02720000
            SQ[MIX]+=*P(DUP);                                 02720100
        END;                                                 02721000
    PROCEDURE DIRECTORYFULL(PASSBY); VALUE PASSBY; REAL PASSBY; 02722000
    BEGIN REAL T;                                           02723000
        STREAM(LOST:=PASSBY#0, P1MIX, T:=T:=SPACE(10));     02724000
        BEGIN DS+34 LIT "*** SHOULD H/L; DISK DIRCTRY FULL="; 02725000
            SI+LOC P1MIX; DS+2 DEC;                           02726000
            P1MIX+DI; DI+DI-2; DS+FILL; DI+P1MIX;           02727000
            LOST(DI+DI-3;DS+17 LIT",SOME FILES LOST=");     02728000
            DS+LIT "+";                                       02729000
        END;                                                 02730000
        SPOUT(T);                                           02731000
        T:=SPACE(30);                                       %R7102731100
        M[T+28]+@114;                                       02732000
        IF PASSBY#0 THEN DISKWAIT(T,30,PASSBY,[CF]);        02733000
        FORGETSPACE(T);                                       02734000
        IF [MEM[P1MIX,0]], [CF]#FENCE THEN                  02735000
            SWAP(WAITSWAP,1)                                  02736000
        ELSE SLEEP(0,0);                                     02737000
    END;                                                     02738000
    PROCEDURE DCERR(R);                                     02740000
    VALUE R; REAL R;                                       02741000
    BEGIN REAL RCW#+0;                                       02742000
        LABEL E0,E1,E2,E3,E4,NEXT,SEARCH,FOUND,ENDIT;      02742500
        REAL A,B,C;                                          02743000
        SWITCH ERR:=E0,E1,E2,E3,E4;                         02743500
        IF (B:=R,[4:4])=15 THEN GO SEARCH;                   02743900
        A:=SPACE(10);                                        02744000
        STREAM(R:=R&0[4:4:4];Y:=R,[9:4];Z:=R,[14:4],A);    02745000
        BEGIN SI:=LOC R;                                     02746000
            DS:=14 LIT"#DCA ERR = RD=";                       02747000
            16(DS:=3 RESET; 3(IF SB THEN DS:=SET ELSE DS:=RESET; 02748000
                SKIP 1 SB)); SI:=LOC Y;                      02749000
            DS:=8LIT",TU/BUF="; DS:=2DEC;DS:=LIT"/";DS:=2DEC; %R9502749100
            R:=DI;                                           02750000
        END; C:=P;                                           02751000
        GO TO ERR[B];                                        02751100
    E0: STREAM(R:=C);                                       02751200
        DS:=15 LIT "-BAD TU/BU NUM=";                        02751300
        GO TO NEXT;                                         02751400
    E1: STREAM(R:=C);                                       02751500
        DS:=14 LIT "-BAD RES DESC=";                         02751600
        GO TO NEXT;                                         02751700
    E2: STREAM(R:=C);                                       02751800
        DS:=15 LIT "-BAD INPUT ADR=";                        02751900
        GO TO NEXT;                                         02752000
    E3: STREAM(R:=C);                                       02752100
        DS:=19 LIT "-READ BOUNCED BACK=";                   02752200
        GO TO NEXT;                                         02752210
    E4: STREAM(R:=C);                                       02752220
        DS:=29 LIT"-BUSY INTERRUPT DURING WRITE=";          02752230

```

```

NEXT:
      SPOUTIT(A,35);
SEARCH: A:=R,[9:9];
        IF B#0 AND B#4 THEN
          FOR B:=1 STEP 1 UNTIL LMAX DO
            IF A=LINETABLE[B],[9:9] THEN GO FOUND;
          GO ENDIT;
FOUND:  IF BLASTREAD(B,3) THEN
        $ SET OMIT = TWXONLY
        IF LINEDISC[B]=MULTI THEN
          BEGIN A:=SEQARRAY[B],[40:8];
              STABLE[A]:=(*P(DUP))&P(DUP)[23:22:11]&1[24:47:1]
                &1[21:47:1];
              IF TANKLINE[A]=0 THEN
                IF TAILOUT#A THEN
                  BEGIN TANKLINE[TAILOUT]:=A;
                      TAILOUT:=A;
                      STARTWORKING;
                  END;
                ENTEREADYQ(B);
          END;
        $ POP OMIT
        ;
ENDIT:  KILL([RCW] INX NOT 2);
        END OF FUNNY DC RESULT DESCRIPTOR HANDLING;
        PROCEDURE DCBUSY(V);
        VALUE V;
        REAL V;
        BEGIN
          REAL RCW:=+0;
          REAL A;
          A:=SPACE(5);
          IF V,[CF] = 0 THEN
            BEGIN
              STREAM(A); DS:=29 LIT"=*#PREVIOUS INPUT(S) LOST#=#*";
            END
          ELSE
            BEGIN
              STREAM(V:=V,[CF]+1,A);
            BEGIN S1:=V;
                DS:=17 LIT"=*#LINE STARTING "; DS:=LIT"";
                DS:=10 CHR;
                DS:=LIT""; DS:=8 LIT " LOST#=#*";
            END;
          FORGETAREA(M[V],[2:2],V);
        END;
        TWXOUT(A,37,0&3[1:46:2],V,[FF]);
        FORGETSPACE(A);
        KILL([RCW] INX NOT 2);
        END BUSY INTERRUPT HANDLER;
        ARRAY SPACER[*];
        REAL BIGUNS;
        DEFINE BIGUNMIN = 4#;
        PROCEDURE MOREAREAS;
          BEGIN REAL I,J;
              REAL RCW:=+0;
              I:=(J:=GETSPACE(162.5,5)+1)+3 AND NOT 3;
              MOVE(162,J,J+1);
              J:=I+159;

```

```

02752300
02752400
02753000
02754000
02755000
02756000
02757000
02758000
02758090
02758100
02758200
02758300
02758400
02758500
02758600
02758700
02758800
02758900
02759000
02759100
02759200
02759210
02759400
02759500
02760000
02761000
02762000
02763000
02764000
02765000
02766000
02767000
02768000
02769000
02770000
02771000
02772000
02773000
02774000
02775000
02776000
02777000
02778000
02779000
02780000
02781000
02783000
02784000
02785000
02786000
03000000
03001000
03001100
03002000
03003000
03003100
03004000
03005000
03006000
03007000

```

```

FOR I←I STEP 20 UNTIL J DO
    FORGETAREA(2,I);
AREARDY←TRUE;
KILL([RCW] INX NOT 2);
END;
SAVE PROCEDURE FORGETAREA(N,T); VALUE N,T; REAL N,T;
BEGIN REAL R,S;
$ SET OMIT = NOT(CHECKLINK OR DEBUGGING)
REAL SP,ME,MS; LABEL PUN;
$ POP OMIT
T←P(,T,LOD),[CF];
DO UNTIL T≠0;
WHILE N≠2 AND (S←M[R←(IF N THEN 10&T[1:46:1] ELSE
    5&T[1:47:1]) +T]),[1:3]=N+4 DO
BEGIN M[S,[FF]],[CF]←S;
M[S],[FF]←S,[FF];
IF T>R THEN T←R;
N←N+1;
END;
IF N=2 THEN BIGUNS←BIGUNS+1;
M[S←M[R←P(M[SPACER[N]],O&T[CTF],LLL,DEL),[FF]],[FF]]
, [CF]←T;
M[M[T]←R&S[CTF]&N[2:46:2]], [FF]←T;
$ SET OMIT = NOT(CHECKLINK OR DEBUGGING)
IF CHECK THEN
FOR N1=0 STEP 1 UNTIL 2 DO
BEGIN T1=R1=SPACER[N],[CF];
SP1=S1=[SPACER[N]], [CF];
MS1=M[0],[CF];
ME1=PRT,[CF];
DO BEGIN
IF M[R],[FF]≠S THEN GO TO PUN;
IF R≠SP THEN
IF M[R],[2:2]≠N OR M[R] GEQ 0 THEN GO TO PUN;
S1=R;
IF (R1=M[R],[CF]) LEQ MS OR (R GEQ ME AND R≠SP) THEN
PUN: PUNT(3); % DATACOM INVALID LINK
END UNTIL R=T;
END;
$ POP OMIT
END;
SAVE REAL PROCEDURE GETAREA(N); VALUE N; REAL N;
BEGIN REAL T←+1,I;
I←N;
WHILE (T←SPACER[I]),[CF]≠[SPACER[I]], [CF] DO
I←I+1;
DO UNTIL I≤2;
M[(SPACER[I])←T&M[T1←T,[FF]][FTF]),[FF]]
, [CF]←[SPACER[I]]; % USE LIL AREAS FRM END
M[T]←O&N[2:46:2];%
IF I=2 THEN
IF (BIGUNS1←BIGUNS-1) LSS BIGUNMIN THEN
IF AREARDY THEN TOGLE,[25:2]←2;
WHILE I≠N DO
FORGETAREA(I←I-1,(IF I THEN 10 ELSE 5)+T);
END;
SAVE PROCEDURE QUEVENT(T,MIX);
VALUE T,MIX; REAL T,MIX;

```

```

03008000
03009000
03010000
03011000
03012000
03013000
03014000
03015000
03016000
03016009
03016010
03016011
03016500
03016600
03017000
03018000
03019000
03021000
03022000
03023000
03024000
03025000
03026000
03026100
03027000
03027004
03027005
03027010
03027020
03027030
03027040
03027050
03027060
03027070
03027080
03027090
03027100
03027110
03027120
03027130
03027140
03027141
03028000
03029000
03030000
03031000
03032000
03033000
03033100
%R7703034000
%R7703035000
%R7703036000
03037000
03038000
03039000
03043000
03044000
03046000
03048000
03049000

```

BEGIN M[EVENT[MIX],[FF]], [CF]+T;	03050000
EVENT[MIX],[FF]+T;	03051000
M[T],[CF]+0;	03051100
END;	03052000
PROCEDURE COMM1; % DISK I/O COMMUNICATE	03053000
BEGIN REAL DA=-5, SZ=-6, F=-7, IO=-9, RES=-10;	03054000
ARRAY A=-8[*];	03055000
REAL S,T;	03056000
M[T + GETAREA(0)], [FF] + RES;	03056500
IF IO THEN	03057000
BEGIN IF NOT A,[2:1] THEN MAKEPRESENT([IO] INX 1);	03057100
M[T+2] + M[S + M[A,[FF]], [CF]-2]&P(S, XCH)[2:2:1];	03057200
M[T+3]+M[S+1]; P([M[S]], IOR); S+-(A INX F);	03057300
END ELSE	03057500
BEGIN S1=SPACE(SZ);	03058000
MOVE(SZ,A,S); M[T+2] + NABS(S);	03058100
END;	03058200
T1=[M[T+1]];	03058500
DISKIO(T, NOT 0 INX S, SZ&1[2:47:1], DA);	03058600
GO TO RETURN	03059000
END;	03060000
PROCEDURE COMM2; % COMMAND LANGUAGE WAIT COMMUNICATE	03061000
BEGIN ARRAY A=-6[*];	03062000
NAME CLK=-7, WT=-8;	03062100
REAL T,S;	03062200
LABEL X;	03062300
S+A,[FF];	03062400
IF (T+A,[CF])#0 THEN	03063000
BEGIN M[S],[CF]+0;	03063500
FORGETAREA(M[T],[2:2],T);	03064000
END;	03064500
WHILE EVENT[P1MIX],[CF]=0 DO	03065000
BEGIN IF NOT WT[0] THEN GO TO X;	03065500
CLICK=CLOCK+P(RTR)+256;	03065600
SLEEP([EVENT[P1MIX]], (NOT 0), [CF]);	03066000
IF TERMSET(P1MIX) THEN GO TO RETURN;	03066200
END;	03066500
IF (T+(S+(M[S]+(A&(T+EVENT[P1MIX])[CTC]) OR M))&T[FTF]),	03067000
[CF]=0 THEN T,[FF]+[EVENT[P1MIX]];	03067100
EVENT[P1MIX]+T;	03067200
LOGLINE+S,[FF];	03067300
WT[0]+1;	03067400
X: CLK[0]+PROCTIME[P1MIX]+CLOCK+P(RTR);	03067500
GO TO RETURN	03068000
END;	03069000
PROCEDURE LOGOUT; FORWARD;	03070000
REAL PROCEDURE INPUTSCAN(MODE, SOURCE, DEST, NUM, FLAGS);	03071000
VALUE    NUM, FLAGS;	03072000
REAL    MODE, SOURCE, DEST, NUM, FLAGS;	03073000
FORWARD;	03074000
REAL PROCEDURE OUTRAN980(ADR, NUM, TYPE, LCC, B);	03075000
VALUE    ADR, NUM, TYPE, LCC;	03076000
REAL    ADR, NUM, TYPE, LCC, B;	03077000
FORWARD;	03078000
PROCEDURE GIVEAWAY(A); VALUE A; REAL A; FORWARD;	03079000
% SET OMIT = TWXONLY	03079990
REAL PROCEDURE OUTRANBIDS(ADR, NUM, TYPE, B, C);	03080000
VALUE ADR, NUM, TYPE; REAL ADR, NUM, TYPE, B, C; FORWARD;	03080100
REAL PROCEDURE OUTRANTC(ADR, NUM, TYPE, B, C);	03080200
VALUE ADR, NUM, TYPE; REAL ADR, NUM, TYPE, B, C; FORWARD;	03080300

```

$ POP OMIT
$ SET OMIT = NOT SEPTICTANK
PROCEDURE RUNSEPTIC(BUFF); VALUE BUFF; REAL BUFF; FORWARD;
SAVE PROCEDURE DISPOSAL(L,I,R); VALUE L,I,R; REAL L,I,R; FORWARD;
ARRAY ARGH[*];
DEFINE SEPTICSIZE = 500#; % SEPTICSIZE MUST BE A MULTIPLE OF 2
SEPTICEOF = 249#; % AND SEPTICEOF = SEPTICSIZE/2 = 1
$ POP OMIT
PROCEDURE MAKEPRESENT(A); VALUE A; REAL A; FORWARD;
PROCEDURE COMM9;
BEGIN REAL I=-5;
REAL L=-7;
REAL ARRAY A=-6[*];
LOGLINE=L;
I:=GETSPACE((L:=A,[8:10])+4,64,5)+4;
IF NOT A,[2:1] THEN MAKEPRESENT([L] INX 1);
MOVE(L,A,I);
CCARD(I&26[3:43:5]);
GO TO RETURN;
END;
PROCEDURE REPORTBACK(WHY,P1,P2);
VALUE WHY,P1,P2;
REAL WHY,P1,P2;
BEGIN REAL T;
IF LOGLINE,[33:7]#0 THEN
BEGIN M[T+GETAREA((P2>0)*2)],[FF]+LOGLINE;
M[T+1]+WHY;
M[T+2]+XCLOCK+P(RTR);
M[T+3]+P1;
IF P2>0 THEN MOVE(10,P2,T+4) ELSE M[T+4]+P2;
QUEVENT(T,CANDEMIX[LOGLINE,[40:8]]);
END;
END;
PROCEDURE TWXOUT(ADR,NUM,TYPE,LL);
VALUE ADR,NUM,TYPE,LL;
REAL ADR,NUM,TYPE,LL;
BEGIN LABEL START,EXIT;
$ SET OMIT = TWXONLY
LABEL BYEBYE;
REAL OOP;
$ POP OMIT
REAL A,B,L,N,T,C,D;
DEFINE TANKING=STABLE[LL].OUTPUTTANKING#;
SUBROUTINE RITE;
BEGIN DISKWAIT(A,30,GLOMSIZE*LL+((L+N) AND NOT GLOMSIZE)
+TANKADDRESS);
N:=N+1;
$ SET OMIT = NOT(STATISTICS)
COUNTUP(16,1);
$ POP OMIT
M[A]:=0;
END;
IF (LL:=LL AND (=255)) LEQ 0 THEN
BEGIN B=SPACE(NUM,[39:16]+2);
STREAM(N1+NUM,[36:16],N2+NUM,ADR,B);
BEGIN SI+ADR; N1(DS+32 CHR; DS+32 CHR);
DS+N2 CHR; DS=LIT "+";
END;
SPOUT(B);
P(EXIT);

```

```

03080301
03080990
03081000
03081100
03081200
03081300
03081400
03081910
03099000
03100000
03101000
03102000
03103000
03104000
03105000
03106000
03107000
03108000
03109000
03110000
03111000
03112000
03113000
03114000
03115000
03116000
03117000
03118000
03119000
03120000
03121000
03122000
03123000
03124000
03125000
03126000
03127000
03127099
03127100
03127200
03127201
03128000
03128100
03129000
03130000
03131000
03131100
03131199
03131200
03131201
03131500
03132000
03133200
03133300
03133400
03133500
03133600
03133700
03133800
03133900

```

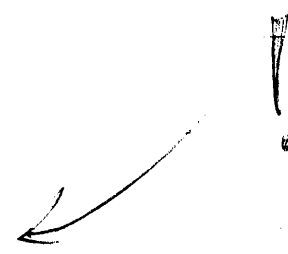
TWXOUT



```

END; 03134000
START: IF TERMSET(P1MIX) AND LL#ABS(SPOWORD) THEN GO TO RETURN; 03134100
IF BREAK[LL] OR DISCONNECTING[LL] OR 03134110
NOT STATABLE[LL],DIALEDUP THEN P(XIT); 03134120
IF NUM GTR 184 THEN NUM:=184; % = 1 SEG AFTER LINE FOLDING 03134130
IF LL LEQ LMAX THEN 03134140
IF SCHEDULE[LL] THEN 03134150
BEGIN IF STATABLE[LL],DIALEDUP THEN 03134200
SCHEDIO(NUM,TYPE,(ADR&LL[CTF])); 03134250
P(XIT); 03134300
END; 03134400
IF NOT REMOTE THEN P(XIT); 03134500
IF NOT(TANKS[LL]),[1:1] THEN 03135000
SLEEP([TANKS[LL]],0); 03136000
T:=TANKS[LL]:=ABS(*P(DUP)); 03136300
A=T,[CF]; 03137000
DI=T,[SOUSE]; 03140000
LI=T,TANKL+D; 03141000
$ SET OMIT = TWXONLY 03141099
IF STATABLE[LL],STATIONTYPE NEQ TWX THEN 03141100
BEGIN DI=T,[14:5]=TNAOG[SEQARRAY[LL],[26:6]],[1:5]; 03141200
DI=(IF D < 0 THEN GLOMSIZE ELSE 0)+D+T,[SOUSE]+N; 03141300
END; 03142000
$ POP OMIT 03142001
IF D+9 GEQ GLOMSIZE THEN 03142100
BEGIN IF D+6 GEQ GLOMSIZE THEN 03142200
IF NOT T,[10:1] THEN 03143000
BEGIN M((B*GETAREA(0))+1)*10; 03143100
M[B]*0&10[18:41:7]&LL[25:40:8]; 03143200
T:=TANKS[LL]:=T&1[10:47:1]; 03143300
QUEVENT(B,CANDEMIX[LINE]); 03143400
END; 03143500
IF P(0,RDS)>FENCE THEN 03143600
BEGIN TANKS[LL]:=NABS(T)&1[11:47:1]; 03144000
SWAP(WAITSWAP,1); 03145000
GO TO START; 03146000
END; 03147000
IF D GEQ GLOMSIZE THEN 03148000
BEGIN CLICK:=CLOCK+1023; TANKS[LL]:=NABS(*P(DUP)); 03148100
SLEEP([TANKS[LL]],0&(NOT T)[14:14:5]); 03148110
IF TANKS[LL],[14:19]=T,[14:19] THEN 03148120
BEGIN TANKS[LL]:=NABS(*P(DUP))&0[19:19:14]; 03148130
$ SET OMIT = TWXONLY 03148139
STATABLE[LL]:=(*P(DUP))&0[23:23:1]; 03148140
$ POP OMIT 03148141
P(XIT); 03148150
END; 03148160
GO TO START; 03148200
END; 03148300
END; 03149000
% AT THIS POINT DATA=COMM DROPPINGS COHERE TO FORM A MIGHTY EXTRUSION 03150000
$ SET OMIT = TWXONLY 03169999
IF(DI=STATABLE[LL],STATIONTYPE) = TC500 THEN 03170000
IF(NUM:=OUTRANTC(ADR,NUM,TYPE,B,C)) LSS 2 THEN 03170100
GO BYEBYE 03170200
ELSE ELSE 03170300
IF D NEQ TWX THEN 03170400
IF(NUM:=OUTRANBIDS(ADR,NUM,TYPE,B,C)) LSS 2 THEN 03170500
GO BYEBYE 03170600
ELSE ELSE 03170700

```



\$ POP OMIT	NUM:=OUTRAN980(ADR,NUM,TYPE,LONGCARRIAGE[LL],B);	03170701
	IF NOT TANKING THEN	03170800
\$ SET OMIT = TWXONLY	IF (STABLE[LL],[CF])=0	03178000
	AND D=TWX) OR (D≠TWX AND NOT(SEQARRAY[LL],SELECTED)	03179000
\$ POP OMIT	) THEN	03179090
	BEGIN IF NOT AREARDY THEN	03179100
	SLEEP([TOGGLE],AREARDYMASK);	03179101
\$ SET OMIT = NOT(TWXONLY)	DCWRITE(B,LL,NUM);	03179900
	GO TO EXIT;	03180000
	END;	03180100
\$ SET OMIT = TWXONLY	TANKING:=TRUE;	03180190
	OOP:=DCWRITE(B,LL,NUM=(D≠TWX));	03180200
	IF D≠TWX THEN C.[47:1]:=1 ELSE GO TO EXIT;	03180225
	END;	03180250
	TANKING:=NOT(OOP);	03180275
\$ POP OMIT OMIT	IF A=0 OR STABLE[LL],SWAPPED THEN	03180290
	BEGIN M[A+B-1]+NUM=1; RITE END ELSE	03180300
	BEGIN IF (T=M[A],[CF])+NUM GTR 232 THEN	03180350
	BEGIN	03180400
\$ SET OMIT = TWXONLY	M[A]:=ABS(*P(DUP));	03181000
\$ POP OMIT	RITE;	03181010
	T:=0;	03182000
	END;	03183000
\$ SET OMIT = TWXONLY	STREAM(N1:=T,[45:3],B,N3:=NUM,N2:=P(DUP),[40:2],	03184000
	X:=D:=(M[A],[1:1] AND C.[2:1]);	03185000
\$ POP OMIT	A:=T,[39:6]+A+1);	03185090
\$ SET OMIT = TWXONLY	BEGIN DI:=D1+N1; SI:=B;	03185100
	X(SI:=SI+1; DI:=DI-1);	03185101
\$ POP OMIT	N2(DS1=32 CHR; DS1=32 CHR);	03185500
	DS1=N3 CHR;	03186000
	END STREAM;	03187000
\$ SET OMIT = NOT(TWXONLY)	M[A]:=T+NUM-1;	03188000
\$ POP OMIT		03188390
\$ SET OMIT = TWXONLY	M[A]:=T+NUM-1-D-D;	03188400
	IF C OR (C.[45:1] AND M[A],[CF] NEQ 0) THEN	03188401
	BEGIN RITE;	03188500
	END ELSE M[A]:=(*P(DUP))&C[1:1:1];	03189000
\$ POP OMIT		03189090
	END;	03190000
\$ SET OMIT = TWXONLY	IF C.[45:1] THEN %WE R GION DISCONNECT U	03190001
	BEGIN IF A=0 OR STABLE[LL],SWAPPED THEN A:=B-1;	03191000
	STABLE[LL],[16:1]:=0;	03192000
	STREAM(A:=A+1); DS:=2 LIT LEFTARROW;	03193000
	M[A]:=1; RITE;	03193490
	TANKING:=TRUE;	03193500
		03193501
		03193990
		03194000
		03194100
		03194200
		03194500
		03194501
		03195000
		03195090
		03195100
		03195150
		03195200
		03195300
		03195400
		03195500

```

                IF STABLE[LL],STATIONTYPE THEN
                    TNAOG[SEQARRAY[LL],[26:6]]:=(P(DUP))&1[42:42:6];
                END;
$ POP OMIT
EXIT:   FORGETSPACE(B=1);
        TANKS[LL]:=NABS(P(DUP))&(P(DUP),[SOUSE]+N
$ SET OMIT = TWXONLY
        =00P
$ POP OMIT
        ) [27:42:6];
        END TWXOUT;
PROCEDURE CLEAR TANK(LL,MCP);
    VALUE LL,MCP; REAL LL,MCP;
    BEGIN REAL T;
        IF (LL:=LL.[40:8]) LEQ LMAX THEN
            IF LL=0 OR SCHEDULE[LL] THEN P(XIT);
$ SET OMIT = TWXONLY
            IF STABLE[LL],STATIONTYPE=TWX THEN
$ POP OMIT
                IF MCP THEN
                    IF TANKS[LL],[SOUSE] = 0 THEN
                        BEGIN STREAM(A:=7=CLOCK,[40:2],T:=(T:=GETAREA(0))+1);
                            BEGIN A(DS:=LIT "<"); DS:=LIT "+"; END;
                            M[T]:=(P(DUP))&LL[10:40:8];
                            ENTERLINEQ(T,LL,0);
                        END;
                        IF NOT TANKS[LL],[1:1] THEN
                            SLEEP([TANKS[LL]],-0);
                            T:=TANKS[LL]:=ABS(P(DUP));
                            IF T,[CF]#0 THEN
                                IF M[T]#0 THEN
                                    BEGIN
$ SET OMIT = TWXONLY
                                        M[T]:=ABS(P(DUP));
$ POP OMIT
                                        DISKWAIT(T,[CF],30,GLOMSIZE*LL+((T,TANKL+T,TANKN)
                                            AND NOT GLOMSIZE)+TANKADDRESS);
$ SET OMIT = NOT(STATISTICS)
                                        COUNTUP(16,1);
$ POP OMIT
                                        M[T]:=0;
                                        STABLE[LL],OUTPUTANKING:=1;
                                        T:=1;
                                    END ELSE T:=0;
                                        TANKS[LL]:=NABS(P(DUP))&(P(DUP),[SOUSE]+T)[TANKNDIAL];
                    END;
REAL PROCEDURE DCWRITE(ADR,LINE,SIZE);
    VALUE ADR, LINE, SIZE;
    REAL ADR, LINE, SIZE;
    BEGIN
$ SET OMIT = NOT TWXONLY
        REAL LBUF,FBUF;
        REAL BUF=NT1,BUFSZ=NT2,A=NT3;
        IF NOT (PAPER TAPE[LINE] OR LOCKED[LINE]) AND
            STABLE[LINE],DIALEDUP THEN
            BEGIN
                BUF:=IF SIZE<28 THEN 0 ELSE
                    (LINETABLE[LINE],[6:3]+1),[45:2];
                BUFSZ:=(BUF OR P(DUP),[46:1]) * 28 + 28;
                DO

```

```

03195600
03195700
03195900
03195901
03196000
03197000
03197090
03197100
03197110
03197200
03198000
03200000
03201000
03202000
03202600
03202700
03202799
03202800
03202801
03203100
03203200
03203300
03203400
03203450
03203500
03203600
03203900
03204000
03205000
03206000
03207000
03207500
03207599
03207600
03207601
03208000
03209000
03209099
03209100
03209101
03210000
03210100
03210200
03211000
03212000
03213000
03214000
03214100
03214200
03214300
03214900
03215000
03215100
03215200
03215250
03215300
03215400
03215500
03215600
03215700

```

DCWRITE

```

BEGIN
    STREAM(ADR; X:=IF SIZE>BUFSZ THEN BUFSZ ELSE SIZE,
          X1:=P(DUP)>63, Z:=(A:=GETAREA(BUF))+1);
    BEGIN SI:=ADR;
          X1(DS:=32 CHR; DS:=32 CHR); DS:=X CHR;
          ADR:=SI;
    END;
    ADR:=P;
    IF LBUF#0 THEN M[LBUF]:=(*P(DUP))&A[CTC] ELSE FBUF:=A;
    M[LBUF:=A]:=(*P(DUP))&LINE[10:40:8]&LBUF[CTF];
END UNTIL (SIZE:=SIZE-BUFSZ) LEQ 0;
M[LBUF]:=(*P(DUP))&1[4:47:1];
ENTERLINEQ(FBUF,LINE,0);
END;
$ SET OMIT = TWXONLY
REAL BUF=NT1,NR=NT2,A=NT3,D=NT4,DISC=NT5,CCNT=NT6,LTIN=NT7,LBUF,ABC;
REAL STYP,T,TN,PM,PC,LM,LC,TEXTMD,LFC,NFC=NT7,S=PM;
LABEL LINKEMIN,GNC,PNC,DOTC,POP,POL,SEND;
LABEL IGNOREOUTPUT,GETANOTHERBUF;
IF PAPER TAPE[LINE] OR LOCKED[LINE] OR
  NOT STABLE[LINE],DIALEDUP THEN GO IGNOREOUTPUT;
BUF:=(LTIN:=LINETABLE[IF LINE GTR LMAX THEN STABLE[LINE],LEENKER
  ELSE LINE]),[6:3] + 1],[45:2];
IF SIZE=1 THEN BUF:=0;
BUF:=BUF&((P(DUP) OR P(DUP),[46:1]) * 28 + 28)[CTF];
IF (STYP:=STABLE[LINE],STATIONTYPE) NEQ TWX THEN
  BEGIN
    TN:=TNAOG[(T:=TANKS[LINE]&(S:=SEQARRAY[LINE])[42:26:6]),[42:6]];
    CCNT:=T,[19:8];
  END ELSE
  CCNT:=0; DISC:=1;
DO
  BEGIN
GETANOTHERBUF: IF DISC GTR 2 THEN BUF:=0&28[CTF];
D:=(A:=GETAREA(BUF,[CF]))+1;
IF(NR:=ABC:=BUF,[FF]) GTR SIZE THEN NR:=SIZE;
IF STYP = TWX THEN %SPITWX IT OUT
  BEGIN;
    STREAM(A:=ADR; X:=NR, NR:=P(DUP),[41:1], D);
    BEGIN SI:=A; NR(DS:=32 CHR; DS:=32 CHR);
          DS:=X CHR; A:=SI;
    END STREAM;
    ADR:=P;
    GO LINKEMIN;
  END ELSE NR:=0;
  IF DISC#7 THEN GO POP;
  IF SIZE = 1 THEN
    BEGIN;
      STREAM(DISC:=0;ADR,HBIT:=T,[12:1],
            EOTBIT:=(P(DUP)=0) AND (DISC LSS 29),D);
      BEGIN SI:=ADR; IF SC=LEFTARROW THEN
            SI:=SI+1; IF SC=LEFTARROW THEN
            BEGIN TALLY:=28;
                  EOTBIT(DS:=LIT "$"; TALLY:=29);
                  HBIT(DS:=LIT "#"; TALLY:=5);
                  DS:=LIT LEFTARROW;
            END;
            DISC:=TALLY;
      END STREAM;
    IF(DISC:=P) GTR 0 THEN

```

```

03215800
03215900
03216000
03216100
03216200
03216300
03216400
03216500
03216600
03216700
03216800
03216900
03217000
03217100
03217300
03217400
03217500
03217600
03217700
03217800
03217820
03217900
03218000
03218100
03218200
03218300
03218400
03218500
03218600
03218700
03218800
03218900
03219000
03219050
03219100
03219200
03219300
03219400
03219500
03219600
03219700
03219800
03219900
03219950
03220000
03220100
03220200
03220300
03220400
03220450
03220500
03220600
03220700
03220750
03220800
03220900
03221000
03221100
03221200
03221300

```

	BEGIN T.[12:1]:=FALSE;	03221350
	GO LINKEMIN;	03221400
	END END;	03221500
	IF DISC = 3 THEN	03221600
	BEGIN;	03221700
	STREAM(D);	03221800
	DS:=2 LIT "#*";	03222000
	DISC:=2;	03222100
	GO LINKEMIN;	03222200
	END;	03222300
	IF NOT T.[12:1] THEN     %TIME TOO	03222400
	BEGIN;                 %GET A=HEAD	03222500
	IF NOT T.[13:1] THEN	03222510
	BEGIN M[LM:=GETAREA(0)]:=0&LINE[10:40:8]&2[5:43:5];	03222520
	LBUF:=LM&LM[CTF];	03222530
	END;	03222540
	STREAM(TN:=TN,[38:10],BOP:=(TEXTMD:=	03222600
	(TN,[28:8] OR TN,[42:6])=0);	03222620
	STAD:=S,C1:=P(DUP),[24:1],C2:=S,[25:1],	03222700
	C3:=(P(DUP)#S,[24:1]);	03222750
	MFH:=(LTIN,LINEDIS=MULTI);	03222800
	EOL:=(TN,[28:8]+1=TN,[20:8]);	03222850
	ERAS:=T,[13:1],BIDI:=(STYP=BIDS);	03222900
	TC:=(STYP=TC500);	03223000
	TXT:=TEXTMD:=(STABLE[LINE],[23:1]),D;	03223200
	SEL:=LM+1);	03223250
	BEGIN	03223300
	ERAS(JUMP OUT TO NOSEL);	03223350
	MFH(SI:=LOC STAD; SI:=SI+2;	03223400
	DS:=LIT "S"; C1(DS:=LIT "#"); DS:=CHR;	03223500
	C2(DS:=LIT "#"); DS:=CHR; C3(DS:=LIT "#");	03223600
	DS:=LIT "Q");             % SELECT MULTI	03223700
	DS:=2 LIT "#*";         % OR CONTENTION	03223800
NOSEL:	DI:=D;	03223850
	MFH(SI:=LOC STAD; SI:=SI+2;	03223900
	DS:=LIT ">"; C1(DS:=LIT "#"); DS:=CHR;	03224000
	C2(DS:=LIT "#"); DS:=CHR;	03224100
	C3(TC(JUMP OUT 2 TO ITN); DS:=LIT "#");	03224200
ITN:	TC(SI:=LOC TN; DS:=3 DEC;	03224300
	IF TOGGLE THEN TALLY:=0 ELSE TALLY:=1;	03224400
	TN:=TALLY; DS:=LIT "#");	03224500
	DS:=LIT " ";	03224600
	TC(JUMP OUT TO PYLON);	03224800
	ERAS(BID(DS:=3 LIT " ", " ; JUMP OUT 2 TO NEX);	03224900
	BOP(JUMP OUT 2 TO NEX); DS:=3 LIT "1 " );	03224950
NEX:	BOP(DS:=2 LIT "#*"; JUMP OUT TO NEWPG);	03225000
	BID(JUMP OUT TO NEWPG); %DONT DO BACKSPACE FOR BIDS	03225040
	DS:=3 LIT "( ";	03225050
	EOL(DS:=3 LIT "3 " );	03225060
NEWPG:	TXT(DS:=LIT "#");	03225100
PYLON:	BOP:=DI;	03225200
	END HEADING STREAM;	03225300
	DI:=P; PM:=P;	03225400
	IF STYP=TC500 THEN %UP TRANSMISSION NUMBER	03225500
	TNAOG[T,[42:6]]:=(P(DUP))&(IF PM THEN 1 ELSE	03225600
	P(DUP),[38:10]+1)[38:38:10];	03225700
	T:=T&2[12:46:2];	03225800
	ABC:=ABC"(D,[30:3]&(D=A=1)[30:33:15]);	03226000
	END;                 %HEADINGOUT FOR RETAILNG	03226200
	IF STYP NEQ TC500 THEN   % BID FOR CONRAC	03226300

	BEGIN	03226400
	NR:=0;	03226450
	IF(PM:=TN,[14:6])=0 THEN GO DOTC; %MAX LINES/PAGE	03226500
	IF(LM:=TN,[20:8])=0 THEN GO DOTC; %MAX CHAR/LINE	03226600
	IF(PCI:=TN,[42:6]) GEQ PM THEN GO POP; %LINES/PAGE	03226700
	IF(LCI:=TN,[28:8]) GEQ LM THEN GO POL; %CHAR/LINE	03226800
	IF DISCONNECTING[LINE] THEN PCI:=1;	03227000
GNC:	IF CCNT + NR GEQ SIZE THEN GO SEND;	03227100
	IF TEXTMD THEN	03227200
	BEGIN	03227300
	STREAM(NC:=0,A1:=ADR,	03227400
	MC:=(IF LM=LC LEQ ABC=NR THEN LM=LC ELSE ABC=NR)	03227500
	-1,	03227550
	MW:=P(DUP),[41:1] x 2,FCT:=LFC,D1:=D1);	03227600
	BEGIN D1:=D1; S1:=A1;	03227650
	FCT(IF SC="#" THEN JUMP OUT TO LASTC);	03227700
	TALLY:=0; FCT:=TALLY;	03227800
	MW(32(IF SC="#" THEN	03227900
	BEGIN MC:=TALLY; TALLY:=0; MW:=TALLY;	03228000
	JUMP OUT 2 TO LASTC;	03228100
	END ELSE DS:=CHR; TALLY:=TALLY+1));	03228200
	MC(IF SC="#" THEN JUMP OUT ELSE	03228300
	DS:=CHR; TALLY:=TALLY+1);	03228400
	MC:=TALLY;	03228500
LASTC:	D1:=D1; D1:=LOC NC; D1:=D1+7;	03228550
	DS:=CHR; A1:=S1;	03228600
	END STREAM;	03228650
	D:=P; IF(LFC:=P) THEN P(DEL,DEL)	03228700
	ELSE NR:=P(32,x,+,DUP,LC,+,LC,STD,NR,+);	03228750
	END ELSE	03228800
	STREAM(NC:=0,A1:=ADR);	03228900
	BEGIN S1:=A1; D1:=LOC NC;	03229000
	D1:=D1+7; DS:=CHR; A1:=S1;	03229100
	END STREAM;	03229200
	ADR:=P;	03229300
	IF (NFC:=P) = 31 THEN GO SEND; %GROUP MARK	03229400
	IF TEXTMD THEN	03229500
	BEGIN	03229600
	IF NFC=60 THEN TEXTMD:=FALSE; %CHANGE MODE	03229700
	GO PNC;	03229800
	END ELSE	03229900
	IF(NFC=43 OR NFC=44) THEN %LF OR CRLF	03230000
	IF DISCONNECTING[LINE] THEN	03230100
	BEGIN IF NFC=43 THEN NFC:=48; GO PNC END	03230200
	ELSE BEGIN IF STYP=BIDS THEN	03230500
	BEGIN LC:=0; NFC:=43; END ELSE	03230600
	IF NFC=44 THEN LC:=0;	03230700
	PC:=PC+1;	03230800
	END	03230900
	ELSE IF NFC=60 THEN TEXTMD:=TRUE %CHANGE MODE	03231000
	ELSE IF(NFC=4 OR NFC=58) THEN LC:=PC:=0 %HOMER	03231100
	ELSE IF NFC=29 THEN LC:=LC-1 %BACKSPACE	03231200
	ELSE IF NFC=3 THEN PC:=PC-1 %REV LF	03231300
	ELSE IF NFC=2 THEN LC:=LC+1; %NON DES SPACE	03231400
	STREAM(NC:=NFC:D);	03231500
	BEGIN S1:=LOC NC; S1:=S1+7;	03231600
	DS:=CHR; NC:=D1;	03231700
	END STREAM;	03231800
	D:=P;	03231900
	IF NFC=60 THEN LC:=LC+LFC	03232000
PNC:		

	ELSE LC:=LC+TEXTMD;	03232100
	NR:=NR+1;	03232200
	IF LC GEQ LM OR PC GEQ PM THEN	03232300
	BEGIN %THERE IS A LINE FULLOFIT	03232400
	IF DISCONNECTING(LINE) THEN PC:=1;	03232500
	IF(PC:=PC+1)	03232600
	GEQ PM THEN %ND A PAGE	03232700
	BEGIN PC:=PM; %FULLSOFIT	03232800
	IF NR=ABC THEN GO LINKEMIN	03232900
	ELSE	03233000
POP:	BEGIN; %STUFF A BUFF IE,(ETXIT)	03233100
	STREAM(F1:=(ABC-NR GEQ (TEXTMD+5)):	03233200
	F2:=TEXTMD;	03233300
	F4:=(PC GEQ PM)+D);	03233400
	BEGIN TALLY:=7;	03233450
	F1( F2(DS:=LIT "%");	03233500
	F4(DS:=3 LIT "4 ";	03233600
	JUMP OUT TO NEWPG);	03233700
	DS:=3 LIT "1 ";	03233800
NEWPG:	DS:=LIT "%"; TALLY:=2);	03233900
	DS:=LIT LEFTARROW; F1:=TALLY;	03234000
	END STREAM;	03234100
	IF(DISC:=P)=2 THEN %STUFFED AND BUFFED	03234200
	IF LC GEQ LM THEN	03234300
	BEGIN; STREAM(A1:=ADR,FRY:=TEXTMD);	03234400
	BEGIN SI:=A1; TALLY:=1;	03234500
	IF SC=LEFTARROW THEN	03234600
FIRE:	BEGIN A1:=TALLY; TALLY:=0; FRY:=TALLY END;	03234700
	FRY(IF SC="%" THEN	03234800
	BEGIN SI:=SI+1; TALLY:=2;	03234900
	IF SC=LEFTARROW THEN JUMP OUT TO FIRE	03235000
	ELSE	03235100
PAN:	IF SC="%" THEN	03235200
	BEGIN TALLY:=1; FRY(TALLY:=0);	03235300
	FRY:=TALLY; SI:=SI+1;	03235400
	GO TO PAN;	03235500
	END ELSE TALLY:=1;	03235600
	END ELSE BEGIN TALLY:=0; FRY:=TALLY END;	03235700
	A1:=TALLY; TALLY:=1;	03235800
	FRY(TALLY:=0);	03235900
	FRY:=TALLY);	03236000
	END STREAM;	03236100
	TEXTMD:=P;	03236200
	CCNT:=CCNT+P;	03236300
	END ELSE IF(SIZE=NR=CCNT) LEQ 3 THEN	03236400
	CCNT:=SIZE=NR;	03236500
	GO LINKEMIN;	03236800
	END END ELSE %PAGE NOT FULL	03236900
POL:	LC:=LC-LM;	03237200
	END;	03237300
	LFC:=(IF LFC THEN 0 ELSE (NFC=60));	03237400
	IF NR LSS ABC THEN GO GNC	03237500
	ELSE IF CCNT+NR GEQ SIZE AND NFC=60 THEN	03237600
	DISC:=7; GO LINKEMIN;	03237650
SEND:	CCNT:=SIZE=NR;	03237700
	IF T.[SOUSE] GTR 1 THEN	03237800
	BEGIN; STREAM(D); DS:=LIT LEFTARROW;	03237900
	END ELSE	03238000
	GO POP;	03238100
	END ELSE %TC500	03238300

```

DOTC:      BEGIN IF(NR:=ABC) GTR SIZE=CCNT THEN NR:=SIZE=CCNT;      03238400
           STREAM(DISC:=1,A1:=ADR;      03238500
                 X:=NR,NR:=P(DUP),[41:1];      03238600
                 LASTB:=((CCNT+NR) GEQ SIZE);      03238700
                 BUNGIT:=(ABC=1 GTR NR);      03238750
                 CRAMIT:=(ABC GTR NR)*D);      03238800
           BEGIN SI:=A1;      03238850
                 NR(DS:=32 CHR; DS:=32 CHR);      03238900
                 DS:=X CHR;      03239000
                 LASTB(TALLY:=3;      03239100
                       CRAMIT(DS:=LIT "#");      03239200
                       BUNGIT(DS:=LIT LEFTARROW); TALLY:=2);      03239300
                       DISC:=TALLY);      03239400
                 A1:=SI;      03239450
           END STREAM;      03239500
           ADR:=P; DISC:=P;      03239600
           END TC500;      03239700
LINKEMIN:  03239900
           M[A]:=(*P(DUP))&LINE[10:40:8]&LBUF[CF];      03240000
           IF LBUF NEQ 0 THEN M[LBUF]:=(*P(DUP))&A[CTC]      03240100
           ELSE LBUF,[FF]:=A;      03240200
           CCNT:=CCNT+NR;      03240250
           LBUF,[CF]:=A;      03240300
           IF STYP THEN      03240340
               TN:=TN&LC[28:40:8]&PC[42:42:6];      03240360
               IF DISC GEQ 12 THEN M[LBUF]:=(*P(DUP))&DISC[7:43:3];      03240380
               IF(DISC GEQ 3 AND DISC) THEN GO GETANOTHERBUF;      03240400
           END UNTIL CCNT GEQ SIZE OR DISC GTR 1;      03240600
           M[LBUF]:=(*P(DUP))&DISC[4:47:1];      03240650
           IF STYP NEQ TWX THEN      03240700
               BEGIN      03240800
                   IF CCNT LSS SIZE THEN      03240900
                       T:=T&CCNT[19:40:8]      03241000
                   ELSE      03241100
                       T:=T&(T,[SOUSE]=P(DUP)≠0)[19:34:14]&      03241200
                       (T,[14:5]+1)[14:43:5];      03241300
                   IF M[LBUF,[FF]][5:5]=2 THEN SEQARRAY[LINE]:=NABS(*P(DUP));      03241350
                   TANKS[LINE]:=(*P(DUP))&T[13:13:20]&      03241400
                   DISC[12:47:1];      03241500
                   IF STYP THEN      03241530
                       BEGIN TNAOG[T,[42:6]]:=(*P(DUP))&LC[28:40:8]&PC[42:42:6];      03241540
                               STABLE[LINE]:=(*P(DUP))&TEXTMD[23:47:1];      03241550
                       END;      03241560
                   END;      03241600
                   ENTERLINEQ(LBUF,[FF],LINE,0);      03241700
           IGNOREOUTPUT;      03241800
           DCWRITE:=TANKS[LINE],[19:14]=0 AND STYP≠TWX;      03241870
           $ POP OMIT OMIT      03241890
           END DCWRITING;      03241900
           PROCEDURE SPOSET(TYPE,BUFH); VALUE TYPE,BUFH; REAL TYPE,BUFH; FORWARD;      03245900
           PROCEDURE QUITTER(LINE);      03246000
           VALUE LINE; REAL LINE;      03246050
           BEGIN REAL RCW=+0;      03246100
               REAL MIX,T,I,STA,S;      03246120
               ARRAY A[*];      03246140
               LABEL EXIT,BSB,NRDONE;      03246160
               STA=LINE,[FF];      03247050
               IF LOCKED[LINE:=LINE,[CF]] THEN COMPLEXSLEEP(NOT LOCKED[LINE]);      03247200
               IF ($*STABLE[STA])[17:1] THEN GO TO EXIT;      03247350
               %SET QUITN = QUITTER INTERLOCK      03247500

```



S←S&1[17:47:1];	03247650
IF (I←S.[CF])≠0 THEN	03247800
BEGIN %THROWAWAY THE LINE QUEUE	03247950
DO FORGETAREA(T←M[I]),[2:2],I)	03248100
UNTIL (I←T,[CF])=0;	03248250
S←S&0[CTC];	03248400
END;	03248550
IF (I≠LINETABLE[LINE]),[CF] ≠ 0 THEN	03248700
BEGIN %THROWAWAY INPUT QUEUE	03248850
LINETABLE[LINE]←I&0[CTC];	03249000
DO FORGETAREA(T←M[I]),[2:2],I)	03249150
UNTIL (I←T,[FF])=0;	03249300
END;	03249450
M[T←GETAREA(0)]←(*P(DUP))&STA[10:40:8]&1[#:47:1];	03249500
GIVEAWAY(T); %QUEUE INPUT TANK DELETE	03249550
IF (I≠LINETABLE[LINE]),[21:5] = DISCON THEN	03249600
IF CANDEMIX[STA] = 0 THEN ELSE	03249750
BEGIN	03249900
M[T←GETAREA(0)]←(*P(DUP))&STA[10:40:8]&3[5:46:2];	03250050
STREAMC	03250200
\$ SET OMIT = TWXONLY	03250345
A←I,LINEDIS≠TWX,	03250350
\$ POP OMIT	03250355
T←T+1);	03250500
BEGIN %QUEUE A BYE TO CANDE	03250650
\$ SET OMIT = TWXONLY	03250795
A(DS←LIT"");DS←LIT"≠");	03250800
\$ POP OMIT	03250805
DS←3 LIT "BYE";	03250950
\$ SET OMIT = TWXONLY	03251095
A(DS←2 LIT "≠");	03251100
\$ POP OMIT	03251105
DS←LIT "≠";	03251250
END;	03251400
GIVEAWAY(T);	03251550
END ELSE	03251700
\$ SET OMIT = TWXONLY	03251845
IF (I←I,LINEDIS) ≠ MULTI THEN	03251850
\$ POP OMIT	03251855
BEGIN	03252000
S←S&(T←GETAREA(0))[CTC];	03252150
\$ SET OMIT = TWXONLY	03252295
IF I = TWX THEN	03252300
BEGIN %SEND OUT CARR, RET AND LINE FEED	03252450
\$ POP OMIT	03252455
STREAM(T←T+1); DS←4 LIT "≤≠";	03252600
M[T],[4:1]←STA ≠ ABS(SPOWORD);	03252750
\$ SET OMIT = TWXONLY	03252895
END ELSE	03252900
BEGIN % CONTENTION SEND OUT EOT	03253050
STREAM(T←T+1); DS←2 LIT "s=";	03253200
M[T],[5:5]←POLLING;	03253350
END;	03253500
\$ POP OMIT	03253505
END;	03253650
MIX←(STABLE[STA]←S),MIXNR;	03253800
IF DAT[MIX],NDSABLE OR (T←LSTATUS[LINE]≠BROKEN) THEN	03253950
BEGIN	03254100
IF CANDEMIX[STA]≠MIX AND MIX≠0 THEN	03254250
BEGIN BREAK[STA]←1;	03254300

IF TANKFUL[STA] OR INPUTANK[STA].INPUTREADY THEN	03254400
BRINGBACK(MIX);	03254550
IF T THEN	03254700
BEGIN	03254850
IF NOT INPUTANK[STA],[1:1] THEN	03255000
SLEEP([INPUTANK[STA]],-0);	03255050
INPUTANK[STA]:=(*P(DUP))&0[10:40:8]&0[40:40:8];	03255150
END ELSE	03255200
IF PAPERTAPE[STA] THEN     % QUEUE QUEST MARK	03255300
BEGIN M[T:=GETAREA(0)]:=(*P(DUP))&STA[10:40:8]	03255450
&1[5:47:1];	03255600
STREAMC	03255750
\$ SET OMIT = TWXONLY	03255895
A:=LINEDISC[LINE]*TWX;	03255900
\$ POP OMIT	03255905
T:=T+1);	03256050
BEGIN	03256200
\$ SET OMIT = TWXONLY	03256345
A(DS:=LIT"");DS:=LIT"#");	03256350
\$ POP OMIT	03256355
DS:=LIT MARK;	03256500
\$ SET OMIT = TWXONLY	03256645
A(DS:=2 LIT "#");	03256650
\$ POP OMIT	03256655
DS:=LIT "+";	03256800
END;	03256950
GIVEAWAY(T);	03257100
END;	03257250
END	03257400
END ELSE	03257550
TERMINATE(MIX & 61[CTF]);	03257700
IF STA=ABS(SPOWORD) AND LSTATUS[LINE]=BROKEN THEN	03257720
BEGIN	03257740
GO TO BSB;	03257780
END;	03257800
IF STABLE[STA],OUTPUTANKING THEN	03257850
BEGIN	03258000
WHILE NOT TANKS[STA],[1:1] DO	03258150
SLEEP([TANKS[STA]],-0);	03258300
STABLE[STA],OUTPUTANKING := 0;	03258450
IF TANKS[STA],[2:8] NEQ 0 OR TAILOUT = STA THEN	03258600
BEGIN	03258750
T + 0;	03258900
DO I:=T UNTIL (T:=TANKS[T],[2:8])=STA;	03259050
TANKS[I],[2:8]:=TANKS[STA],[2:8];	03259200
IF STA=TAILOUT THEN TAILOUT:=I;	03259350
END;	03259500
IF TANKS[STA],[10:1] THEN	03259650
BEGIN M[(I+GETAREA(0))+1]+0;	03259800
M[I]:=0&STA[CTF]&10[18:41:7];	03259950
QUEVENT(I,CANDEMIX[STA]);	03260100
END;	03260250
END;	03260400
END;	03260550
T:=TANKS[STA]:=P(DUP,LOD,0,SSN,INX);	03260600
IF T#0 AND NOT (STABLE[STA],SWAPPED) THEN M[T]:=0;	03260695
\$ SET OMIT = TWXONLY	03260700
IF STABLE[STA],STATIONTYPE = TC500 THEN	03260850
TNAOG[SEQARRAY[STA],[26:6]]:=(*P(DUP))&0[1:1:13]	03261000
ELSE IF (STABLE[STA],STATIONTYPE = CONRAC OR	03261150
STABLE[STA],STATIONTYPE = BIDS) THEN	

```

BEGIN
    TNAOG[SEQARRAY[STA],[26:6]]:=(P(DUP))&P(0,XCH)[14:14:14];
    STABLE[STA]:=(P(DUP))&0[22:46:2];
END;
$ POP OMIT
BSB:
$ SET OMIT = TWXONLY
    IF LINEDISC[LINE]#TWX THEN SEQARRAY[STA],[1:1]:=0 ELSE
$ POP OMIT
    IF LSTATUS[LINE]#BROKEN THEN SEQARRAY[STA]:=0;
    IF LSTATUS[LINE] # DISCON THEN
BEGIN
    STABLE[STA]:=(P(DUP))&1[16:47:1]; %DISCONNECTING
    QUITN[STA]:=0;
    IF ABS(SPOWORD) = STA THEN GO TO NRDONE;
    M[T:=GETAREA(0)]:=(P(DUP))&(STA&
    ((LSTATUS[LINE]=BROKEN)+1)[36:46:2])[CTF];
    QUEVENT(T,CANDEMIX[LINE]);
END ELSE
BEGIN
    A + [M[GETAREA(1)+1]]&SYSDISKRL[8:38:10]; %<9 WORDS LONG
    SYSDISKIO(3,-STA,A);
    A[0],DIALEDUP+A[1]+0;
    SYSDISKIO(0,-STA,A);
    FORGETAREA(1,A,[CF]-1);
$ SET OMIT = TWXONLY
    IF LINEDISC[LINE]=MULTI THEN SEQARRAY[STA],[3:1]:=1;
$ POP OMIT
    STABLE[STA]:=(P(DUP))&0[14:47:1]&
    % RESET BREAK AND DISCONNECTING
    ((MIX:=P(DUP),MIXNR)=0 OR MIX=CANDYINX)[16:47:1];
    IF STA=ABS(SPOWORD) THEN SPOSET(NABS(0),STA);
END;
NRDONE:
    LINETABLE[LINE]:=(P(DUP))&(
$ SET OMIT = TWXONLY
    IF P(DUP),LINEDIS=CONTENTION THEN WAITINGENQ ELSE
$ POP OMIT
    IDL)[21:43:5];
    ENTEREADYQ(STA); % XMIT NEW DATA, START A POLL FOR MULTIPPOINT
    QUITN[STA]:=0;
EXIT:
    KILL([RCW] INX NOT 2);
END QUITTER;
PROCEDURE DCWAIT(ADR,LINE,R,MASK,CLCK);
VALUE ADR,LINE,MASK,CLCK;
REAL ADR,LINE,MASK,CLCK;
REAL R;
%
% MASK,[47:1] WAIT FOR AN INTERRUPT AFTER I/O COMPLETE.
% [46:1] LOCK THE LINE FIRST.
% [45:1] WAIT FOR AN IDLE STATUS BEFORE LOCKING.
% [1:1] UNLOCK THE LINE WHEN DONE,
%
BEGIN REAL A,B;
    LABEL TRYAGAIN;
    IF LINE GTR LMAX THEN P(XIT);
    IF MASK.[46:1] THEN IF LOCKED[LINE] THEN
TRYAGAIN; COMPLEXSLEEP(NOT LOCKED[LINE]);
    IF NOT LOCKED[LINE] THEN

```

```

03261300
03261450
03261600
03261750
03261755
03261800
03261895
03261900
03261905
03262050
03262200
03262350
03262500
03262580
03262600
03262650
03262800
03262950
03263100
03263250
03263400
03263550
03263700
03263850
03264000
03264149
03264150
03264151
03264300
03264450
03264600
03265400
03265450
03265500
03265550
03265599
03265600
03265601
03265650
03265700
03265750
03265800
03265950
03266000
03267000
03267200
03267400
03267600
03267700
03267710
03267720
03267730
03267740
03267750
03267800
03268000
03268200
03268300
03268400
03268500

```

DCWAIT

```

BEGIN
    IF MASK,[45:1] THEN
        IF LINETABLE[LINE],[21:2]#1 THEN
            BEGIN CLICK:=CLOCK+P(RTR)+320;    % WAIT FOR IDL
                COMPLEXSLEEP(LINETABLE[LINE],[21:2]=1);
                IF LOCKED[LINE] THEN GO TRYAGAIN;
            END;
            LOCKED[LINE]#1;
        $ SET OMIT = TWXONLY
        IF LINEDISC[LINE]#TTY THEN SEQARRAY[LINE],SELECTED:=1;
        $ POP OMIT
        IF (B:=STABLE[LINE]),[CF]#0 THEN    % THROW AWAY OUTPUT QUEUE
            BEGIN STABLE[LINE]:=B&0[CTC];
                DO FORGETAREA((A:=M[B]),[2:2],B) UNTIL (B:=A,[CF])=0;
            END;
            M[B:=GETAREA(0)]:=(*P(DUP))&1[4:47:1]&LINE[10:40:8];
            GIVEAWAY(B); %QUEUE INPUT DELETE TO OLDWIERDHAROLD
            IF (B:=LINETABLE[LINE]),[CF]#0 THEN    % THROW AWAY INPUT QUEUE
                BEGIN LINETABLE[LINE]:=B&0[CTC];
                    DO FORGETAREA((A:=M[B]),[2:2],B) UNTIL (B:=A,[FF])=0;
                END;
            END;
        $
        R:=0;
        LSTATUS[LINE]#IDL;
        LINETABLE[LINE],[CF]:=P(R,L0D);
        ENTERLINEQ(ADR,LINE,1);
        CLICK:=CLOCK+P(RTR)+60;    % SET 1 SECOND I/O MAX
        SLEEP(P(R,L0D),=0);
        IF MASK THEN                % WAIT FOR INTERRUPT
            IF R,[CF]#ADR+1 THEN    % UNLESS I/O BOUNCED OR
                IF R,[CF] # 0 THEN % IT IS ALREADY BACK
                    BEGIN R:=0;
                        IF CLCK#0 THEN CLICK:=CLOCK+P(RTR)+CLCK;
                        SLEEP(P(R,L0D),=0);
                    END;
                $
                IF MASK,[1:1] THEN                % UNLOCK THE LINE
                    BEGIN % RESET LOCKED, LSTATUS, INPUT QUEUE
                        $ SET OMIT = NOT(TWXONLY)
                        LINETABLE[LINE]:=ABS(*P(DUP))&0[CTC]&IDL[21:43:5];
                        $ POP OMIT
                        $ SET OMIT = TWXONLY
                        IF (LINETABLE[LINE]:=ABS(*P(DUP))&0[CTC]&(IF P(DUP),LINEDIS
                            =CONTENTION THEN WAITINGENQ ELSE IDL)[21:43:5])
                            ,LINEDIS#TWX THEN SEQARRAY[LINE],[1:4]#0;
                        $ POP OMIT
                        END;
                        STABLE[LINE],[CF]#0;
                    END;
                REAL PROCEDURE INPUTSCAN(MODE,SOURCE,DEST,NUM,FLAGS);
                VALUE NUM,FLAGS;
                REAL MODE,SOURCE,DEST,NUM,FLAGS;
                BEGIN INTEGER I,J,N;
                    REAL U;
                    REAL DATA=U+1,NOTEQ=DATA+1,ASCII=NOTEQ+1,BS=DATA;
                    REAL INPTSC=#+1;
                    LABEL AUT;
                $
                INPUTSCAN:=DEST;
                J:=NUM MOD 63;

```

```

03268600
03268700
03268800
03269000
03269200
03269400
03270000
03270200
03270399
03270400
03270401
03270600
03270800
03271000
03271100
03271200
03271300
03271400
03271600
03271800
03272000
03272200
03272400
03272600
03272800
03273000
03273100
03273200
03273400
03273600
03273800
03274000
03274200
03274400
03274600
03274800
03275000
03275200
03275399
03275400
03275401
03275599
03275600
03275800
03276000
03276001
03276200
03276400
03276600
03276800
03277000
03277200
03277400
03277600
03277800
03278000
03278200
03278400
03278600
03278800

```

\$ SET OMIT = TWXONLY	03278999
IF FLAGS,[33:3]#TWX THEN	03279000
P(MODE,P(DUP),[34:1],P(XCH),[33:1],FLAGS,[47:1]=1) ELSE	03279200
\$ POP OMIT	03279201
P(FLAGS,[FF]);	03279400
U:=FLAGS,[46:1];	03279600
FOR I:=NUM DIV 63 STEP -1 UNTIL 0 DO	03279800
BEGIN	03280000
IF (N:=IF I=0 THEN J ELSE 63) = 0 THEN GO AUT;	03280200
\$ SET OMIT = TWXONLY	03280399
IF FLAGS,[33:3]=TWX THEN	03280400
\$ POP OMIT	03280401
STREAM(SOURCE, DEST, MODE:=4; N, BS, U);	03280600
BEGIN SI:=SOURCE; DI:=DEST;	03280800
N(IF SC = "≤" THEN	03281000
BEGIN DEST:=DI; DI:=LOC BS;	03281200
SOURCE:=SI; SI:=LOC DEST;	03281400
IF 8 SC=DC THEN DI:=DEST ELSE	03281600
BEGIN DI:=DEST; DI:=DI-1 END;	03281800
SI:=SOURCE; SI:=SI+1;	03282000
END ELSE	03282200
IF SC = "+" THEN	03282400
BEGIN SI:=SI+1; U(DS:=LIT"+");	03282600
TALLY:=1; MODE:=TALLY;	03282800
JUMP OUT TO EXIT;	03283000
END ELSE DS:=CHR);	03283200
EXIT: DEST:=DI; SOURCE:=SI;	03283400
\$ SET OMIT = TWXONLY	03283599
END	03283600
ELSE	03283800
BEGIN	03284000
STREAM(SOURCE, DEST, MODE:=4, DATA, NOTEQ: ASCII, U, N);	03284200
BEGIN SI:=SOURCE; DI:=DEST;	03284400
N(NOTEQ(TALLY:=0; NOTEQ:=TALLY);	03284600
IF SC="#" THEN JUMP OUT TO DSIT;	03284800
DATA(JUMP OUT TO L1); TALLY:=1;	03285000
L1: DATA:=TALLY; JUMP OUT TO L2);	03285200
IF SC = "#" THEN	03285400
BEGIN TALLY:=1; NOTEQ:=TALLY; SI:=SI+1;	03285600
ASCII(DS:=LIT"#"); GO TO L;	03285800
END;	03286000
L2: DATA(JUMP OUT TO DSIT);	03286200
IF SC = "#" THEN TALLY := 1 ELSE	03286400
IF SC = "+" THEN TALLY := 1 ELSE	03286600
BEGIN ASCII(JUMP OUT TO DSIT);	03286800
IF SC#"=" THEN BEGIN SI:=SI+1; GO TO L END	03287000
ELSE TALLY:=2;	03287200
END;	03287400
JUMP OUT TO SETALLY;	03287600
DSIT: DS:=CHR;	03287800
L: ); GO TO L3;	03288000
SETALLY: SI:=SI+1; MODE:=TALLY; U(DS:=LIT"+");	03288200
L3: SOURCE:=SI; DEST:=DI;	03288400
END;	03288600
NOTEQ:=P;	03288800
P(DATA:=P,P&P[34:47:1]&NOTEQ[33:47:1]);	03289000
\$ POP OMIT	03289001
END;	03289200
P([MODE],STD);	03289400
P([DEST],STD);	03289600

```

P([SOURCE],STD);
IF MODE,[45:1] THEN ELSE GO AUT;
END;
AUT:
INPUTSCAN:=(DEST,[30:3]&DEST[30:33:15]) = (INPTSC,[30:3]&
INPTSC[30:33:15]);
END;
PROCEDURE COMM13;
BEGIN
% INPUT TO NORMAL STATE JOBS ABOVE THE FENCE
ARRAY A:=5[*];
REAL N:=6,D,T,X,A1,A2,A3,SORC,N1,N2;
REAL MODE,FLAGS;
LABEL START,NEXTSEG,READIT,EXIT;
DEFINE COUNTEND = [45:1]#;
%
% INPUTTANKS IS NOT LOCKED = DONT LOSE CONTROL WHILE UPDATING IT
%
SUBROUTINE UPDATEINPUTANK;
BEGIN
WHILE NOT (X:=INPUTANK[T]),[1:1] DO SLEEP([INPUTANK[T]],-0);
IF M[A1],[FF] LEQ (A3:=SORC,[30:3]&(SORC=A1-1)[30:33:15])
OR (MODE AND NOT M[A1],[2:1]) THEN % SEG IS EMPTY
X:=X&0[2:2:8]&(X,INPUTN-1)[40:40:8]&((X,INPUTL+1)
AND NOT CLUMPSIZE)[10:40:8]
ELSE X:=X&A3[2:40:8];
$ SET OMIT = TWXONLY
INPUTANK[T]:=X&MODE[38:33:2];
$ POP OMIT
$ SET OMIT = NOT(TWXONLY)
INPUTANK[T]:=X;
$ POP OMIT
END;
%
START:
IF TERMSET(P1MIX) THEN GO TO RETURN;
IF LINEDISC[
$ SET OMIT = TWXONLY
DI:=IF (T:=LOGLINE,[40:8])>LMAX THEN STABLE[T],LEENKER ELSE T
$ SET OMIT = NOT TWXONLY
T:=LOGLINE,[40:8]
$ POP OMIT OMIT
J = SCHED THEN
BEGIN SCHEDIO(N,0,(-0&T[CTF]&(A)[CTC])); P(XIT); END;
IF (X:=INPUTANK[T]),INPUTN = 0 THEN % NOTHING IN TANK
BEGIN BREAK[T]:=0;
IF N GTR 0 THEN % SEND OUT QM IF NOT PT OR READIN
$ SET OMIT = TWXONLY
IF NOT(PAPERTAPE[T] OR P(LINETABLE[D],DUP).LINEDIS#MULTI AND
P(XCH),[21:1]) THEN
$ POP OMIT
$ SET OMIT = NOT(TWXONLY)
IF NOT (PAPERTAPE[T] OR LINETABLE[T],[21:1]) THEN
$ POP OMIT
BEGIN
STREAM(X:=GETAREA(0))+1);
BEGIN DS:=LIT"#"; DS:=LIT MARK; END;
TWXOUT(X+1,2,-0,LOGLINE);
FORGETAREA(0,X);
IF INPUTANK[T],INPUTN#0 THEN GO TO START;
END;

```

```

03289800
03290000
03290200
03290400
03290600
03290800
03291000
03292000
03292200
03292600
03292800
03292900
03293000
03293200
03293400
03293600
03293800
03294000
03294200
03294400
03294600
03294700
03295000
03295200
03295400
03295599
03295600
03295601
03295799
03295800
03295801
03296000
03296200
03296400
03296600
03296750
03296775
03296800
03296825
03296850
03296875
03296900
03297000
03297200
03297400
03297600
03297799
03297800
03298000
03298001
03298199
03298200
03298201
03298400
03298600
03298700
03298800
03299000
03299200
03299400

```

```

        INPUTANK[T],INPUTREADY:=1;
        SWAP(WAITSWAP,1);
        GO TO START;
    END;
%
    N2:=(N1=ABS(N))x8;
    DI=A,[CF];
    STREAM(N:=N-1,A);
    BEGIN DS:=8 LIT" "; SI:=A; DS:=N WDS; END;
    SORC:=((A1:=SPACE(30))+X,[2:5])&X[30:7:3]+1;
    A3:=A2:=SPACE(30);
% SET OMIT = TWXONLY
    FLAGS:=1&STABLE[T][33:10:3]&DAT[P1MIX][46:7:1]&A2[CTF];
    MODE:=0&X[33:38:2];
% POP OMIT
% SET OMIT = NOT(TWXONLY)
    FLAGS:=1&DAT[P1MIX][46:7:1]&A2[CTF];
% POP OMIT
NEXTSEG:
% SET OMIT = NOT(STATISTICS)
    COUNTUP(16,1);
% POP OMIT
    DISKWAIT(=A1,30,PROGTANK+CLUMPSIZExT+X,INPUTL);
    N1:=INPUTSCAN(MODE,SORC,A3,(M[A1],[FF]=X,[2:8]),FLAGS);
    XI:=IF N1 GTR N2 THEN N2 ELSE N1;
    STREAM(M:=X,[37:6];N:=X,[43:5],A2,D);
    BEGIN SI:=A2;
        M(DS:=32 CHR); DS:=N CHR;
        MI=DI;
    END;
    DI=P;
    UPDATEINPUTANK;
    IF MODE,COUNTEND THEN
    IF (N2:=N2-N1) GTR 0 THEN
    BEGIN SORC:=A1+1;
        A3:=A2;
        GO TO NEXTSEG;
    END ELSE
    BEGIN
        % SCAN TO END OF MESSAGE
    READIT: DISKWAIT(=A1,30,PROGTANK+CLUMPSIZExT+X,INPUTL);
    % SET OMIT = NOT(STATISTICS)
        COUNTUP(16,1);
    % POP OMIT
    % SET OMIT = TWXONLY
        IF STABLE[T],STATIONTYPE=TWX THEN
    % POP OMIT
        BEGIN SORC:=A1+100;
            IF NOT M[A1],MESSEND THEN
            BEGIN UPDATEINPUTANK;
                GO READIT;
            END;
            IF NOT M[A1],[2:1] THEN
            BEGIN UPDATEINPUTANK;
                GO TO EXIT;
            END;
        END;
        SORC:=A1+1;
        A3:=A2;
        N1:=INPUTSCAN(MODE,SORC,A3,M[A1],[FF],FLAGS);
        UPDATEINPUTANK;
        IF MODE,COUNTEND THEN GO READIT;
    END;

```

```

03299600
03299800
03300000
03300200
03300400
03300600
03300800
03301000
03301200
03301400
03301600
03301799
03301800
03302000
03302004
03302199
03302200
03302201
03302400
03302599
03302600
03302601
03302800
03307000
03307200
03307400
03307600
03307800
03308000
03308200
03308400
03308600
03308800
03309000
03309200
03309400
03309600
03309800
03310000
03310200
03310399
03310400
03310401
03310599
03310600
03310601
03310800
03311000
03311200
03311400
03311600
03312200
03312400
03312600
03312800
03313000
03313200
03313400
03313600
03313800

```

END OF HANDLING COUNTEND;	03314000
EXIT;	03314200
IF X,SLOWDOWN THEN	03314400
IF X,INPUTN LSS ,5*CLUMPSIZE THEN	03314600
\$ SET OMIT = TWXONLY	03314799
BEGIN	03314800
IF STABLE[T],STATIONTYPE = TWX THEN	03315000
\$ POP OMIT	03315001
BEGIN STREAM(A2);	03315200
DS:=15 LIT"CONTINUE TYPING";	03315400
TWXOUT(A2,15,1,LOGLINE);	03315600
\$ SET OMIT = TWXONLY	03315999
END ELSE TANKOK[T]:=0;	03316000
\$ POP OMIT	03316001
INPUTANK[T],SLOWDOWN:=0;	03316800
END;	03317000
FORGETSPACE(A1);	03317200
FORGETSPACE(A2);	03317400
END COMM13;	03317600
%	03333950
PROCEDURE COMM5;	03375000
%DATACOMM INPUT COMMUNICATE FOR C&E,	03375050
%FORMAT OF DISK BUFFER WHICH OLDWIERDHAROLD WRITES ON DISK WHEN FUL	03375100
% LASTSEG = 2 = WORD SET BY IOFINISH WHEN DISK WRITE COMPLETE,	03375200
% LASTSEG = 1 = DISK ADDRESS (DEKIMAL)	03375300
% LASTSEG[0] = INFORMATION WORD WRITTEN ON DISK	03375400
% [3 :15] = RELATIVE DISK ADDRESS WITHIN THIS DISK CHUNK	03375600
% [18:15] = LINK TO NEXT BUFFER	03375700
% [40:8] = NUMBER OF VALID WORDS IN THIS BUFFER SEGMENT	03375900
% LASTSEG[1] THRU LASTSEG[29] CONTAIN INPUT MESSAGES,	03375950
BEGIN	03377000
ARRAY A=-6[*];	03378000
REAL T,U;	03379000
ARRAY S = T[*];	03380000
DEFINE FIOADR=T*2*;%THIS IS DIFFERENT FROM GLOBAL ONE OF SAME NAM	03380100
LABEL LOOPEND;	03380200
IF (T*FIRSTSEG,[CF]) = LASTSEG,[CF] THEN	03381000
BEGIN	03382000
CANDEINPUTREADY + FIRSTOFFSET = LASTOFFSET;	03383000
IF CANDEINPUTREADY THEN	03384000
BEGIN	03385000
FOR U:=1 STEP 1 UNTIL LMAX DO	03385015
IF SCHEDULE[U] THEN	03385020
BEGIN IF STABLE[U].DIALEDUP THEN	03385025
IF SEQARRAY[U] LSS 0 THEN	03385030
BEGIN SEQARRAY[U]:=ABS(*P(DUP));	03385040
FORK(P(.SCHEDIO),(-U),0,128,1);	03385050
GO TO LOOPEND;	03385060
END;	03385070
END ELSE GO TO LOOPEND;	03385080
LOOPEND;	03385090
IF(U+A,[CF])=T THEN GO TO RETURN;	03385100
IF U > 1023 THEN FORGETSPACE(U-2);	03385200
M[A,[FF]]+(*P(DUP))&T[CTC] OR M;	03385300
GO TO RETURN;	03385400
END;	03385500
END;	03385600
IF T # (T+A INX 0) THEN	03393000
BEGIN	03394000
IF A.[2:1] THEN	03395000



```

BEGIN
  IF (M[FIOADR] AND IOMASK) = 0 THEN
    SLEEP([M[FIOADR]], IOMASK);
    FORGETSPACE(T=2);
  END;
  M[A,[FF]] ← (A & P(,FIRSTSEG,LOD)[CTC]) OR M;
END;
FORK(P(,NOTIFYCANDE),P1MIX,0,128,1);
GO TO RETURN;
END COMM5;
PROCEDURE ENTERLINEQ(ADR,LINE,PRIRTY);
VALUE ADR,LINE,PRIRTY;
REAL ADR,LINE,PRIRTY;
BEGIN
  REAL A,B,S;
  LABEL TOIT;
  S←STATABLE[LINE];
  A←ADR;
  IF PRIRTY THEN
    BEGIN %LINKINFRONT
      WHILE P(M[ADR],DUP),[CF]≠0 DO ADR←P;
      M[ADR]←P(XCH)&S[CTC];
      GO TOIT;
    END; %NEXT IS LINK AT END
    IF S.[CF]≠0 THEN
      BEGIN
        WHILE P(M[S],DUP),[CF]≠0 DO S←P;
        M[S]←P(XCH)&A[CTC];
      END ELSE
    TOIT:
      STATABLE[LINE]←S&A[CTC];
      ENTERREADYQ(LINE);
    END ENTERLINEQ;
    % SET OMIT = TWXONLY
  PROCEDURE NAKQUE;
  BEGIN % THIS STACK IS 64 WORDS LONG BE CAREFUL
    REAL E,I,S,T,V,RCH←+0; LABEL LOOP;
    WHILE (T:=SEQARRAY[0]),[18:30] ≠ 0 DO
      BEGIN
        WHILE (I:=T,[FF]) ≠ 0 DO
          BEGIN %PLACE ENTRIES IN QUEUE IN WAKEUP TIME ORDER
            T:=T&M[I][CTF];
            IF STATABLE[V:=M[I+1]],STATIONTYPE=TC500 THEN
              BEGIN %TC500 NAK RETRANSMIT TIME IS NOW 1/2 SECOND
                M[I+2]←(*P(DUP))+30;
                E←(TNAOG[SEQARRAY[V],[26:6]]←(*P(DUP))&(IF
                  (S:=P(DUP),[38:10])≠0 THEN S=1 ELSE 999)[38:38:10]);
                END ELSE %FOR B9352 NAK RETRANSMIT TIME INCREASES IN
              BEGIN %INCREMENTS OF 3 SEC. FOR EACH SUCCESSION NAK,
                M[I+2]←(*P(DUP))+(E←(S:=SEQARRAY[V])
                  ,[9:3] + 1)×180;
                IF E≠8 THEN SEQARRAY[V]←S&E[9:45:3];
                E←(TNAOG[S,[26:6]]←(*P(DUR))&0[28:28:20]);
              END; %NEXT = RESET TANK TO POINT TO THE NAKKED MESSAGE
            E:=M[I+2]; S:=T,[CF]; %INSERT STA IN TIME LIST
            WHILE (V:=M[S],[CF])≠0 AND M[V+2] LSS E DO S:=V;
            M[I],[CF]←V; M[S],[CF]←I;
          END; %END PUTTING IN LIST NOW TAKE OUT
          S←CLOCK + P(RTR);
          SEQARRAY[0]←T;

```

ENTERLINEQ

NAKQUE

```

03396000
03397000
03398000
03399000
03400000
03401000
03403000
03414000
03415000
03416000
03417000
03418000
03419000
03420000
03421000
03422000
03423000
03424000
03425000
03426000
03427000
03428000
03429000
03430000
03431000
03432000
03433000
03434000
03435000
03436000
03437000
03438000
03439000
03439995
03440000
03441000
03441500
03442000
03442500
03443000
03443500
03444000
03444500
03445000
03445300
03445400
03445500
03445600
03445700
03445800
03445900
03446000
03446100
03446200
03448000
03448500
03449000
03449500
03450000
03450500

```

```

LOOP:  IF (E:=M[(I:=T,[CF])+2]) LEQ S THEN %IS IT TIME TO TRY AGAIN 03451000
      BEGIN %YES, DELINK IT FROM THE TIMER QUEUE 03451500
        V:=M[I+1]; 03452000
        T:=SEQARRAY[0]:=T&M[I][CTC]; 03452500
        FORGETAREA(0,I); %NOW DETANK THE I/O AND TRY AGAIN, 03453000
        IF TANKLINE[V]=0 AND TAILOUT#V THEN 03453500
          BEGIN 03454000
            TANKLINE[TAILOUT]:=V; 03454500
            TAILOUT:=V; 03455000
            STARTWORKING; 03455500
          END; 03455800
          IF T,[CF] # 0 THEN GO LOOP; 03455900
        END ELSE %NOW SLEEP EITHER TIME FOR NEXT STA OR 1/2 SEC (MIN) 03456000
        BEGIN CLICK:=IF E GTR (I:=S+30) THEN I ELSE E; 03456100
          SLEEP(0,0); %WHEN SLEEP IS DONE GO BACK AND PLACE NEWLY 03456200
        END; END; %NAKED STATIONS IN LIST AND WAKE UP THE ONES THAT CAN BE 03456300
        KILL([RCW] INX NOT 2); 03456500
      END REMOVING STATIONS FROM NAKQUE; 03457000
      PROCEDURE DCTIMEOUT(R); VALUE R; REAL R; 03458000
      BEGIN %THIS STACK IS 64 WORDS LONG BE CAREFUL 03458200
        LABEL LEAV,LEAV1,TRY,FORGET,ENDIT; 03459000
        REAL L,C,S,T,RCW#+0; 03460000
        REAL SELTIM; 03460500
        LI:=S:=R,[FF]; 03461000
        RI:=R,[CF]; 03462000
        IF (I:=LINETABLE[L]),LINEDIS= MULTI THEN 03463000
          BEGIN 03464000
            IF SELTIM:=T,[21:5] NEQ POLLTIMEOUT THEN 03465000
              SI:=SEQARRAY[L],[40:8] ELSE % SELECT TIMEOUT 03465500
            BEGIN %FIND WHICH STA TIMED OUT, IF ONLY ONE ON LINE IT DID IT 03466000
              IF SEQARRAY[L],LINELINK=L THEN GO LEAV1; 03466500
              STREAM(R:=(R+1)&1[32:47:1]);T:=[T]; 03467000
              BEGIN %LOOK FOR GRPMK IN CHARACTER AFTER EOT. 03468000
                SI:=R; 03469000
                B(SI:=SI+5;IF SC="#" THEN SI:=SI+2; 03470000
                  IF SC#"#" THEN R:=SI ELSE JUMP OUT); 03471000
                SI:=R;DSI=LIT "0"; %PUT ADDR CHARS IN T (2&3 POSIT) 03472000
                IF SC="#" THEN %TALLY REFLECTS CASE CHANGE POSIT 03473000
                  BEGIN %CASE CHANGE BEFORE THE FIRST ADDR CHAR 03474000
                    TALLY:=2;SI:=SI+1; 03475000
                  END; 03475100
                DSI=CHR; 03475200
                IF SC="#" THEN 03475300
                  BEGIN %CASE CHANGE BEFORE THE SECOND ADDR CHAR 03475400
                    SI:=SI+1;TALLY:=TALLY+1; 03475500
                  END; 03475600
                DSI=CHR; 03475700
                R:=TALLY; 03476000
              END; 03477000
            END; 03478000
            T:=P&T[34:6:12]; %PLACE IN FORMAT OF SEQARRAY[12:14] 03479000
            DO IF P(SEQARRAY[S],DUP),[12:14]#T THEN 03480000
              ELSE GO LEAV 03481000
            UNTIL (S:=P,LINELINK)=L; 03481100
            % CANT FIGURE IT OUT = POLL AGAIN 03481200
            FORGETAREA(M[R],[2:2],R); 03481300
            SEQARRAY[L]:=(+P(DUP))&SEQARRAY[SEQARRAY[L],[40:8]] 03481500
              [40:32:8]; 03481700
            LSTATUS[L]:=IDL; 03481900
            ENTERREADYQ(L); 03482000
            GO ENDIT;

```

DCTIMEOUT

```

LEAVI      P(DEL); SEQARRAY[L],[40:8]:=S;          03483000
END;      03484000
LEAVI:    IF (C:=LINETABLE[L]),[CF] ≠ 0 THEN      03484100
          IF NOT LOCKED[L] THEN                  03484150
          BEGIN LINETABLE[L]:=C&0[CTC]; %THROWAWAY INPUT QUEUE 03484200
            DO FORGETAREA(T:=M[C]),[2:2],C)      03484300
            UNTIL (C:=T,[CF]) = 0;              03484400
          END;                                    03484500
          IF DISCONNECTING[S] THEN GO TO TRY;    03484550
          M[C:=GETAREA(0)]:=(*P(DUP))&1[4:47:1]&S[10:40:8]; 03484600
          GIVEAWAY(C); %QUEUE A DELETE TO CE IN CASE ANYTHING WAS SENT 03484700
          SEQARRAY[S]:=ABS(*P(DUP))&(T:=P(DUP),[4:2]+1)[4:46:2]; 03484800
          %THE NEXT PART ATTEMPTS TO FIND IF THE STA REALLY TIMED OUT 03484850
          IF T LSS 4 THEN                        03484900
          BEGIN %TRY 3 TIMES TO SEE IF HE HAS GONE AWAY 03485000
            CLICK:=CLOCK+P(RTR)+30;             03485100
            SLEEP(0,0); %WAIT 1/2 SECOND BEFORE DOING ANYTHING 03485200
            SELTIM:=SEQARRAY[S],SELECTED AND SELTIM; 03485250
            M[R]:=(*P(DUP))&(1+SELTIM)[4:42:6]&0[10:10:38]; 03485300
            STREAM(X:=SELTIM,S:=C:=SEQARRAY[S],T:=T:=C,[24:1], 03485400
              C:=C:=C,[25:1],L:=T≠C,R:=R+1)); 03485500
            BEGIN %COMPOSE A SINGLE POLL TO SEND OUT 03485600
              SI:=LOC S;SI:=SI+2;DS:=LIT "S"; 03485700
              T(DS:=LIT "#");DS:=CHR;          03485800
              C(DS:=LIT "#");DS:=CHR;          03485900
              L(DS:=LIT "#");DS:=3 LIT "P%"; 03486000
              X(DI:=DI-3; DS:=LIT "Q"); % SELECT INSTEAD 03486050
            END; %IF THIS POLL GETS A LEGIT RESPONSE THEN FAKE T.O. 03486100
            LSTATUS[L]:=IDL;                    03486200
            ENTERLINEQ(R,S,1);                  03486300
            GO TO ENDIT;                        03486400
          END; %IF THIS TEST FAILS 3 TIMES HE IS DISCONNECTED 03486500
          SEQARRAY[S],[3:3]:=4;                03486600
        TRY: 03486800
          END ELSE                               03487000
          IF STABLE[S],[16:1] THEN % NOT BEING DISCONNECTED 03487100
          IF BLASTREAD(L,7) THEN                 03487200
          IF (T:=SEQARRAY[L]),[4:1] THEN SEQARRAY[L]:=T&0[4:47:1] ELSE 03487300
          BEGIN SEQARRAY[L]:=NABS(T)&1[4:47:1]; 03487400
            M[C:=GETAREA(0)]:=R&SELECT[5:43:5]&L[10:40:8]; 03487500
            STREAM(C:=C+1,R:=R+1);              03487600
            BEGIN DS:=LIT "#"; DS:=7 LIT "#*###+"; 03487700
              DI:=C; DS:=2 LIT "%";            03487800
            END;                                03487900
            M[R]:=(*P(DUP))&0[4:4:44]&L[10:40:8]; 03488000
            ENTERLINEQ(C,L,1);                  03488100
            GO TO ENDIT;                        03488200
          END                                   03488250
          ELSE GO FORGET; % BLASTREAD FAILED    03488300
          IF (T:=STABLE[S]),DIALEDUP THEN       03489000
          BEGIN                                  03490000
            IF CANDYINX≠0 THEN                  03491000
            BEGIN                                03492000
              M[R]:=(*P(DUP))&0[4:4:44]&        03493000
                (S&6[33:41:7])[CTF];          03494000
              QUEVENT(R,CANDEMIX[S]);          03495000
            END ELSE FORGETAREA(M[R],[2:2],R); 03496000
            STABLE[S]:=T&0[15:46:2];          03497000
            LSTATUS[L]:=DISCON;                03498000
            FORK(P(,QUITTER),L&S[CTF],#2,96,1); 03499000
          END ELSE
END ELSE

```

```

BEGIN
    LINETABLE[LJ]:=(P(DUP))&(IF (T=P(DUP),LINEDIS)≠CONTENTION
        THEN IDL ELSE WAITINGENQ)[21:43:5];
    IF T=MULTI AND SEQARRAY[LJ],[32:8]≠L THEN ENTERREADYQ(L);
FORGET; FORGETAREA(M[R],[2:2],R);
    END;
ENDIT;
    KILL([RCW] INX NOT 2);
END DCTIMEOUT;
    $ POP OMIT
PROCEDURE NOTIFYCANDE;
    BEGIN ARRAY S[*];
        REAL T=S,RCW=+0;
        IF (DISKADR≠DISKADR+FIRSTSEG[0],[17:1]) GEQ TANKCHUNKSIZE THEN
        BEGIN
            M[T:=GETAREA(0)]:=(P(DUP))&5[18:41:7];
            M[T+1]:=BASEDISKADR:=GETUSERDISK(-TANKCHUNKSIZE);
            QUEVENT(T,CANDYINX);
            DISKADR:=0;
        END;
        IF (M[FIOADR] AND IOMASK)=0 THEN SLEEP([M[FIOADR]],IOMASK);
        S1=[M[FIOADR]];
        DISKIO(S,FIRSTSEG,[CF]=1,30,BASEDISKADR+DISKADR);
    $ SET OMIT = NOT(STATISTICS)
        COUNTUP(16,1);
    $ POP OMIT
        M[T:=GETAREA(0)]:=(P(DUP))&7[18:41:7];
        M[T+1]:=FIRSTOFFSET;
        M[T+2]:=FIRSTSEG[0],[40:8]-FIRSTOFFSET+1;
        M[T+3]:=DISKADR;
        QUEVENT(T,CANDYINX);
        IF (LASTSEG INX 0) = (FIRSTSEG INX 0) THEN
        BEGIN
            FIRSTOFFSET ← LASTOFFSET;
        END ELSE
        BEGIN
            FIRSTOFFSET ← 1;
            FIRSTSEG,[CF] ← FIRSTSEG[0],[FF];
        END;
        KILL([RCW] INX NOT 2);
    END NOTIFYCANDE;
PROCEDURE PAPERTAPEIO(R,STA,FLAGS,BUFSZ,MI); % HANDLES END OF MESSAGE
    VALUE R,STA,FLAGS,BUFSZ,MI;
    REAL R,STA,FLAGS,BUFSZ,MI;
    BEGIN LABEL READIT;
        DEFINE DELETE = [4:1]#;
        REAL TEMP,J,X;
        SUBROUTINE PAPERTAPEDISKIO;
        BEGIN
            $ SET OMIT = NOT(STATISTICS)
                COUNTUP(16,1);
            $ POP OMIT
                DISKIO(M[R=2],R=1,(X+7) DIV 8 + 1,PROGTANK+
                    CLUMPSIZE×STA+(((X:=J,INPUTN+TEMP,[10:8]))+J,INPUTL)
                    AND NOT CLUMPSIZE));
                INPUTTANK[STA]:=ABS(J:=J&(X+1)[40:40:8]);
                IF INPUTTANK[STA],INPUTREADY THEN
                BEGIN BRINGBACK(STATABLE[STA],MIXNR);
                    J,INPUTREADY:=0;
                END;

```

```

03499100
03499300
03499400
03499450
03499475
03499500
03500000
03501000
03502000
03502005
03504000
03505000
03506000
03507000
03508000
03509000
03510000
03511000
03512000
03513000
03514000
03515000
03516000
03516099
03516100
03516101
03517000
03518000
03519000
03520000
03521000
03537000
03538000
03542000
03543000
03544000
03545000
03546000
03547000
03548000
03549000
03550000
03550100
03550200
03550300
03550400
03550500
03550600
03550700
03550709
03550710
03550711
03550800
03550900
03551000
03551100
03551200
03551300
03551400
03551500

```

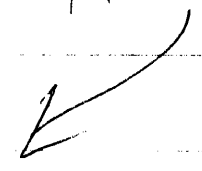
NOTIFYCANDE

PAPERTAPEIO

```

M[R=3],[10:8]:=0; % RESET DELETE POINTER 03551550
IF (X:=TEMP,[CF]=2) GTR 0 THEN 03551600
BEGIN M[X-1]:=M[R-3]; 03551650
IF (M[X] AND IOMASK)=0 THEN SLEEP([M[X]],IOMASK); 03551700
END; 03551750
END; 03551800
% 03551900
J:=INPUTANK[STA]; 03552000
TEMP:=M[R]; 03552100
IF MI,DELETE THEN % DELETE = RESET PTRS TO PREV MESS 03552200
BEGIN IF (X:=M[R-3])=0 THEN ELSE GO READIT; % WE HAVE A PREV MESS 03552300
IF TEMP,[CF]#0 THEN % TRY THE OTHER AREA 03552400
IF (X:=M[R-3])=M[TEMP,[CF]-3]#0 THEN % IF THERE IS ONE, 03552500
READIT: IF X,[10:8]#TEMP,[10:8] THEN % ETX IN A PREVOIUS SEG 03552600
% SET OMIT = NOT(STATISTICS) 03552689
BEGIN COUNTUP(16,1); 03552690
% POP OMIT 03552691
DISKWAIT(=R,30,PROGTANK+CLUMPSIZE*STA+(J,INPUTN+J,INPUTL+ 03552700
X,[10:8]) AND NOT CLUMPSIZE); 03552800
% SET OMIT = NOT(STATISTICS) 03552809
END; 03552810
% POP OMIT 03552811
M[R]:=(*P(DUP))&X[10:10:23]; % REST TO PREVIOUS + 03552900
P(XIT); 03553000
END; 03553100
INPUTANK[STA]:=ABS(J); 03553200
STREAM(A:=(P(TEMP,[FF]=BUFSZ,DUP),[33:12]+R+1)&P(XCH)[30:45:3] ); 03553300
BEGIN SI:=A; % FIND THE LEFT ARROW 03553400
L: IF SC#"*" THEN BEGIN SI:=SI+1; GO TO L END; 03553500
AI:=SI; 03553600
END; 03553700
XI:=P(DUP),[30:3]&(P(XCH)-R-1)[30:33:15]+1; 03553800
M[R=3]:=M[R]:=TEMP&X[CTF]; % SET CHAR COUNT TO + AND 03553900
IF TEMP,[CF]#0 THEN M[TEMP,[CF]-3]:=M[R]; % SAVE FOR DELETE 03554000
IF FLAGS=4 THEN % QUESTION MARK = END OF PAPER TAPE 03554100
BEGIN PAPERTAPEDISKIO; % WRITE IT, SWAP AND UP TANK 03554200
IF X GTR 0 THEN FORGETSPACE(X); % FORGET THE PREVIOUS BUFFER 03554300
SLEEP([M[R-2]],IOMASK); % WAIT FOR CURRENT I/O THEN 03554400
FORGETSPACE(R-2); % FORGET THE BUFFER 03554500
PAPERTAPE[STA]:=FALSE; 03554600
% SET OMIT = TWXONLY 03554699
IF STABLE[STA],STATIONTYPE=TWX THEN 03554700
% POP OMIT 03554701
TWXOUT(0,0,1,STA) 03554800
END ELSE % NOT END = TEST FOR SWAP IN 03555200
IF TEMP,[10:8] GEQ CLUMPSIZE DIV 2 THEN 03555300
BEGIN PAPERTAPEDISKIO; % WRITE IT OUT, SWAP, UP TANK 03555400
J,[FF]:=X+2; % REUSE OLD ARRAY 03555500
M[X+2]:=0&R[CTC]&1[2:47:1]; 03555600
END ELSE J,[FF]:=R; % RESET TO CURRENT BUFFER 03555700
INPUTANK[STA]:=NABS(J); 03555800
END PAPERTAPEIO; 03555900
% 03572000
% HELLO SENDS OUT THE HEADING TO A NEW USER AND UPDATES SYSTEM/DISK 03572100
% TO MARK THE STATION ACTIVE, IT WAITS 7 SECONDS TO ALLOW SUCCESSFUL 03572200
% LINE ESTABLISHMENT, IF THIS IS TOO LONG OR TOO SHORT FOR A GIVEN 03572300
% SITE, THE DELAY CAN BE CHANGED BY CHANGING THE LITERAL AT 03590100, 03572400
% 03572500
PROCEDURE HELLO(K); 03573000
VALUE K; REAL K; 03574000

```

HELLO  


```

BEGIN REAL I,J,L,V,B;                                03575000
REAL RCW=+0;                                         03576000
ARRAY A=B[*];                                        03577000
IF NOT(LINETABLE[J]=M[K+2]).DIRECTLINE) THEN        03589000
IF (I=M[K])+                                         03590000
    448                                             % DELAY IN 64=THS, DEFAULT=7 SECS 03590100
    ) GTR CLOCK+P(RTR) THEN                          03590200
BEGIN                                                03591000
    CLICK+I;                                         03592000
    SLEEP(0,0);                                       03593000
END;                                                 03594000
IF DISCONNECTING[I]=M[K+1] THEN                    03595000
SLEEP([STABLE[I]],@0000020000000000);             03595500
%BE SURE WE ARE THRU DISCONNECTING PREVIOUS USER  03595510
CONNECT(I); %MARK DIALED UP AND ATTACH TO CANDE   03595520
STREAM(TU+(L=LINETABLE[J]),[9:4],BU+L,[14:4],      03595530
    LINEI=I,QQQ:=B:=GETAREA(1)+1);                 03595540
BEGIN                                                03595550
    DS:=24 LIT"***#B5500 TIME SHARING = ";         03595560
    SI:=LOC TU; DS:=2 DEC; DS:=LIT"/"; DS:=2 DEC;   03595570
    DS+10 LIT " STATION "; DS+2 DEC;                03595580
    DS:=4 LIT "=-*=";                                03595590
END;                                                 03595600
TWXOUT(B,63,0&3[1:46:2],1);                        03595620
% NEXT LET CANDE IN ON WHATS HAPPENING             03595630
IF CANDYINX # 0 THEN                                03595640
BEGIN                                                03595650
    M[V:=GETAREA(0)]:=(*P(DUP))&(I&1[33:41:7])[CTF]; 03595660
    M[V+2]:=STABLE[I],[10:3]; %STATION TYPE         03595670
    QUEVENT(V,CANDEMIX[I]);                          03595680
END ELSE                                             03595690
BEGIN STREAM(B);                                     03595750
    DS:=27 LIT"CANDE IS DOWN...PLEASE WAIT";        03595800
    TWXOUT(B,27,1,1);                                03595900
END;                                                 03596000
A := IOQUE&B[CTC]&SYSDISKRL[8:38:10]; % < 9 WORDS LONG 03596100
SYSDISKIO(3,-I,A);                                  03596200
A[0],DIALEDUP + 1;                                  03596300
SYSDISKIO(0,-I,A); %UPDATE SYSTEM DISK             03596400
FORGETAREA(1,A,[CF]=1);                              03596500
FORGETAREA(0,K);                                     03598000
KILL([RCW] INX NOT 2);                               03598200
END HELLO;                                           03599000
PROCEDURE OLDWIERDHAROLD;                             03608000
% DOES MANY STRANGE AND WONDEROUS THINGS.          03609000
% SOME OF THEM USEFUL.                              03610000
BEGIN                                                03611000
    REAL RCW = +0;                                    03612000
    REAL STA; % FIRST SO COM5 CAN FIND IT           03612100
    REAL H,I,J,K,R,S,T,T1,STAB,SOURCE,NUM,FLAGS;    03613000
% SET OMIT = TWXONLY                                03613099
    BOOLEAN UPQFLAG;                                  03613100
% POP OMIT                                           03613101
    REAL MODE=K,BUFSZ=T;                              03614000
    DEFINE CREND = [46:1]#;                          03614100
    DISKDONE = (H+31)#;                              03614200
    DELETE = [4:1]#;                                  03614300
    COUNTEND = [45:1]#;                              03614400
    DEFINE PAPETAPE = (STAB LSS 0)#;                 03614500
    LABEL RESTART,START,LOOP,DELL,CE,AGAIN,PT,DELINKIT; 03615000

```

```

$ SET OMIT = TWXONLY                                03615099
  LABEL NORMALQ,CR,SCANIT;                            03615100
$ POP OMIT                                           03615101
  LABEL FORGIT,DISCARD;                               03615200
  SUBROUTINE UPQ;                                     03616000
$ SET OMIT = TWXONLY                                03616099
  BEGIN IF UPQFLAG THEN                               % SCANNING 30 WORD ARRAYS 03616100
    BEGIN FORGETSPACE(I=2);                           03616200
      IF (I:=M[I],[CF])=0 THEN                       % LAST ARRAY           03616300
        BEGIN UPQFLAG:=0;                             % RESET FLAGS AND I    03616400
          MODE:=1;                                     %MAKE SURE WE STOP     03616430
          GO TO NORMALQ;                               03616500
        END;                                           03616600
      BUFSZ:=M[I],[FF];                               % NUM OF CHARS IN ARRAY 03616700
    END ELSE                                           % DATACOM INPUT BUFFERS 03616800
$ POP OMIT                                           03616801
  BEGIN INPUTANK[STA]:=(+P(DUP))&(M[I],[4:2]#0)[35:47:1]; 03616900
    WORKERINQ,[CF]:=M[I],[CF];                       03617000
    FORGETAREA(M[I],[2:2],I);                         03617100
$ SET OMIT = NOT(TWXONLY)                            03617199
  SOURCE:=(I:=WORKERINQ,[CF])+1;                    03617200
  END;                                                03617300
$ POP OMIT                                           03617301
$ SET OMIT = TWXONLY                                03617399
NORMALQ: I:=WORKERINQ,[CF];                          03617400
  END;                                                03617500
  SOURCE:=I+1;                                        03617600
  T1:=0;                                             % SET SOURCE CHARS SCANNED 03617700
END;                                                03617800
%                                                    03617900
SUBROUTINE SCANINPUT;                               % SCANS INPUT FOR CANDE 03618000
BEGIN                                               % T1 IS SOURCE CHARS SCANNED 03618100
  NUM:=NUM+INPUTSCAN(MODE,SOURCE,J,BUFSZ-T1,FLAGS); 03618200
  T1:=SOURCE,[30:3]&(SOURCE,[CF]-I-1)[30:33:15]; 03618300
END;                                                03618400
$ POP OMIT                                           03618401
%                                                    03618500
SUBROUTINE GETANOTHER;                              03618600
BEGIN                                               % T1 IS OLD OBJ JOB BUFFER 03618700
  INPUTANK[STA],[FF]:=R:=(IF T1 GTR 0 THEN T1 ELSE SPACE(32))+2; 03618800
  M[R-2]:=IOMASK;                                     03618900
END;                                                03619000
%                                                    03619100
SUBROUTINE OBJOBDISKER;                             % DISKIO FOR INPUT TO JOBS 03619200
BEGIN                                               03619300
  J:=INPUTANK[STA]:=ABS(+P(DUP));                    03619400
  DISKIO(M[R-2], R-1, (M[R],[FF]+7) DIV 8+1, PROGTANK+ 03619500
    CLUMPSIZE*STA+((K:=J,INPUTN+M[R],[10:8])+J,INPUTL 03619600
    AND NOT CLUMPSIZE));                             03619700
$ SET OMIT = NOT(STATISTICS)                        03619709
  COUNTUP(16,1);                                     03619710
$ POP OMIT                                           03619711
  IF M[R] LSS 0 THEN J:=J&(K+1)[40:40:8];           % UP INPUTANKS        03619800
  IF (T1:=M[R],[CF]-2) GTR 0 THEN                   % WAIT FOR PREVIOUS I/0 03619900
  IF (M[T1] AND IOMASK) = 0 THEN SLEEP([M[T1]],IOMASK); 03620000
  IF J,INPUTN GEQ ,9*CLUMPSIZE THEN                 % TOO FULL, SLOW HIM BY 03620100
  IF NOT J,SLOWDOWN THEN                             % MESSAGE IF TWX OR NO 03620200
$ SET OMIT = TWXONLY                                03620299
  BEGIN IF STATAB,STATIONTYPE=TWX THEN             % ACK IF NOT TWX      03620300
$ POP OMIT                                           03620301

```

BEGIN STREAM(D:=(K:=GETAREA(0))+1);	03620400
DS:=11 LIT"PLEASE WAIT";	03620500
TWXOUT(K+1,11,1,STA);	03620700
FORGETAREA(0,K);	03620800
\$ SET OMIT = TWXONLY	03620899
END ELSE TANKOK[STA]:=1;	03620900
\$ POP OMIT	03620901
J,SLOWDOWN:=1;	03621000
END ELSE	03621100
IF J.INPUTN=CLUMPSIZE=1 THEN TERMINATE(STATAB,MIXNR&67[CTF]);	03621200
INPUTANK[STA]:=NABS(J);	03621300
% UPDATE AND UNLOCK INPUTANK	03621400
END OF OBJECT JOB INPUT TANK HANDLER;	03621400
%	03621500
RESTART:;	03621600
SOURCE:=(I:=WORKERINQ,[CF])+1;	03621700
% HEAD OF INPUT QUEUE	03621700
START;	03621800
IF I#0 THEN	03621900
% INPUT FROM DATACOM TO PROCESS	03621900
BEGIN	03622000
IF (STA:=M[I],LINENR)=0 THEN	03622100
% SS OR CX FROM SPO	03622100
BEGIN STATAB:=0;	03622200
BUFSZ:=152;	03622300
GO TO CE;	03622400
END;	03622500
IF (STATAB:=STABLE[STA]),OWHROWOUT THEN	03622600
BEGIN	03622700
% RESET ON DELETE OR MESSEND	03622700
STABLE[STA]:=STATAB&(M[I],[4:2]=0)[13:47:1];	03622800
GO TO DELL;	03622900
% IGNORE THIS BUFFER	03622900
END;	03623000
\$ SET OMIT = TWXONLY	03623099
BUFSZ:=28*(IF P(LINETABLE[IF STA GTR LMAX THEN STATAB,LEENKER	03623100
ELSE STA],BUFSIZE,DUP)=2 THEN P+2 ELSE P+1);	03623200
IF NOT (FLAGS:=STATAB,CANDEFLAG OR M[I],[6:1]) OR	03623300
(K:=STATAB,STATIONTYPE#TWX) THEN ELSE GO TO CE;	03623400
\$ POP OMIT	03623401
\$ SET OMIT = NOT(TWXONLY)	03623499
BUFSZ:=28*(IF P(LINETABLE[STA],BUFSIZE,DUP)=2 THEN P+2 ELSE	03623500
P+1);	03623600
IF STATAB,CANDEFLAG OR M[I],[6:1] THEN GO TO CE;	03623700
\$ POP OMIT	03623701
%	03623800
%	03623900
%	03624000
OBJECT JOB OR NOT TELETYPE	03624000
T1:=0;	03624100
IF INPUTANK[STA],FIRSTBUF THEN	03624200
% SCAN OUT HEADER AND QM	03624200
\$ SET OMIT = NOT(TWXONLY)	03624299
IF (P(M[SOURCE],TOP,XCH),[1:5] = @14) AND P THEN	03624300
% QUEST MARK	03624300
IF PAPER TAPE THEN FLAGS:=4 ELSE GO TO CE;	03624400
NUM:=BUFSZ;	03624500
\$ POP OMIT	03624501
\$ SET OMIT = TWXONLY	03624599
BEGIN STREAM(SOURCE,K);	03624600
BEGIN SI:=SOURCE;	03624700
K(28(IF SC="" THEN BEGIN SI:=SI+1; JUMP OUT END);	03624800
SI:=SI+1));	03624900
SOURCE:=SI;	03625000
IF SC#"*" THEN JUMP OUT TO L;	03625100
SI:=SI+1);	03625200
IF SC=MARK THEN TALLY:=1;	03625300
L: K:=TALLY;	03625400
END;	03625500



```

                IF P THEN FLAGS:=4;                % QM FLAG                03625600
                NUM:=ABS((SOURCE:=P),[30:33]&(SOURCE=I-1)[30:33:15]-BUFSZ);03625800
            END ELSE NUM:=BUFSZ;                    03625900
$ POP OMIT                                        03625901
            IF (R:=INPUTANK[STA],[FF])=0 THEN      % GET A NEW BUFFER                03626000
            BEGIN GETANOTHER;                      03626100
                M[R]:=0&STATAB[2:1:1];            % PAPER TAPE FLAG                03626200
            END;                                    03626300
LOOP:                                             03626400
            IF (S:=M[I]),DELETE THEN              03626500
            BEGIN UPQ;                              03626600
                IF PAPERTAPE THEN GO TO PT;        03626700
                DO BEGIN                            % FORGET ALL PIECES OF            03626800
                    SLEEP([M[R-2]],IOMASK);      % CURRENT MESSAGE                03626900
                    FORGETSPACE(R-2);            03627000
                END UNTIL (R:=M[R],[CF])=0;        03627100
                INPUTANK[STA],[FF]:=0;           03627200
                GO TO START;                       03627300
            END;                                    03627400
            IF M[R],[FF]+NUM GTR 232 THEN          % OUT OF ROOM                    03627500
$ SET OMIT = NOT(TWXONLY)                        03627549
            BEGIN OBJBDISKER;                      03627550
$ POP OMIT                                        03627551
$ SET OMIT = TWXONLY                             03627599
            BEGIN IF NOT FLAGS THEN OBJBDISKER;    % IF OBJ. WRITE IT OUT          03627600
$ POP OMIT                                        03627601
                K:=R;                              % GET A NEW ONE AND              03627700
                GETANOTHER;                        % SET FIRST WORD                 03627800
                M[R]:=0&K[CTC]&(M[K],[10:8]+1)[10:40:8]&STATAB[2:1:1];03627900
            END;                                    03628000
%                                                 03628100
            STREAM(N:=NUM DIV 2, SOURCE,          03628200
$ SET OMIT = TWXONLY                             03628299
                N1:=NUM AND 1,                    03628300
$ POP OMIT                                        03628301
                D:=(R+1+M[R],[18:12])&M[R][30:30:3]);03628400
            BEGIN SI:=SOURCE;                      03628500
                DS:=N CHR; DS:=N CHR;            03628600
$ SET OMIT = TWXONLY                             03628699
                DS:=N1 CHR;                      03628700
$ POP OMIT                                        03628701
            END;                                    03628800
            M[R]:=(P(DUP))&(P(DUP),[FF]+NUM)[CTF] OR 0&S[4:4:3];03628900
            UPQ;                                    03629000
            IF NOT S,MESSEND THEN                  % MORE MESSAGE TO COME          03629100
            IF M[I],LINENR=STA AND I#0 THEN        % GO GET NEXT BUFFER            03629200
$ SET OMIT = NOT(TWXONLY)                        03629299
            GO TO LOOP ELSE GO START;              03629300
$ POP OMIT                                        03629301
$ SET OMIT = TWXONLY                             03629399
            BEGIN NUM:=BUFSZ;                      03629400
                GO TO LOOP;                      03629500
            END ELSE GO TO START;                  % NO MORE FOR THIS STA NOW      03629600
$ POP OMIT                                        03629601
            INPUTANK[STA],[FF]:=0;                03629700
            IF PAPERTAPE THEN                      % SPECIAL HANDLING TO PACK      03629800
            BEGIN                                  % PAPER TAPE INPUT              03629900
                PAPERTAPEIO(R,STA,FLAGS,BUFSZ,S);03630000
            GO TO START;                           03630100
            END;                                    03630200
PT:

```



```

$ POP OMIT                                03634701
  BEGIN UPQ;                               03634800
    IF NOT MODE THEN                       % COUNT EXHAUSTED 03634900
    IF I#0 THEN                             03635000
$ SET OMIT = TWXONLY                       03635099
    IF M[I],LINENR=STA OR UPQFLAG THEN GO AGAIN; 03635100
$ POP OMIT                                03635101
$ SET OMIT = NOT(TWXONLY)                 03635199
    IF M[I],LINENR = STA THEN GO AGAIN;    03635200
$ POP OMIT                                03635201
  END;                                     03635300
  IF ABS(SPOWORD) = STA AND STA NEQ 0 THEN %BACK UP SPO 03635305
  BEGIN M[H+1]#0;                          03635310
    IF KEYBOARDCOUNTER = 0 THEN            03635315
    FORK(P(,KEYIN),1&(H+1)[CTF],0,192,0) ELSE 03635320
    IF (R:=KEYBOARDCOUNTER,[FF]) NEQ 0 THEN 03635325
    BEGIN IF (J:=M[R-2],[CF]) NEQ 0 THEN    03635330
      IF M[(R:=J)-2] NEQ 0 THEN GO TO DISCARD; 03635335
      M[R-2]#H+1;                          03635340
    END ELSE KEYBOARDCOUNTER:=KEYBOARDCOUNTER&(H+1)[CTF]; 03635345
    KEYBOARDCOUNTER:=KEYBOARDCOUNTER&1&17:47:1; 03635350
    H:=SPACE(60);                          03635355
    GO TO DISCARD;                         03635360
  END;                                     03635365
% WE HAVE SCANNED INPUT STARTING AT H+1    03635400
IF (P(M[H+1],TOP,XCH),[1:5] = @14) AND P THEN % QUEST MARK 03635500
  FLAG#NABS(FLAGS);                        03635600
IF (R:=(NUM+7) DIV 8)+LASTOFFSET GTR 29 THEN % NO ROOM 03635700
BEGIN IF CANDEINPUTREADY THEN % REUSE CURRENT AREA 03635800
  BEGIN IF (M[IOADR] AND IOMASK) = 0 THEN 03636000
    SLEEP([M[IOADR]],IOMASK);              03636500
    FIRSTOFFSET#1;                        03637000
  END ELSE % GET A NEW ONE                03637500
  BEGIN                                    03638000
    LASTSEG[0],[FF] := (S:=SPACE(32))+2;  03638500
    M[S] := IOMASK;                       03639000
    LASTSEG,[CF]#S+2;                     03639500
  END;                                     03640000
  LASTSEG[0]#0&1[17:47:1];               03640500
  LASTOFFSET := 1;                        03641000
END;                                       03641500
STREAM(MA:=H+1,R,N:=[LASTSEG[LASTOFFSET+1]]); 03642000
BEGIN SI:=MA; DSI=R WDS END;              03642500
LASTSEG[LASTOFFSET]#NUM&FLAGS[1:1:1]&STA[10:40:8]; 03643000
LASTOFFSET,[CF]#(LASTSEG[0]#(*P(DUP))&(LASTOFFSET+R)[CTC])+1; 03643500
IF CANDEINPUTREADY THEN % IF CANDE WANTS IT, GIVE IT TO HER 03644000
BEGIN FORK(P(,NOTIFYCANDE),CANDYINX,0,128,1); 03644500
% OLDWEIRDHAROLD IS A MOTHER FORKER 03645000
  CANDEINPUTREADY#FALSE;                 03645500
END;                                       03646000
DISCARD:                                  03646100
$ SET OMIT = TWXONLY                       03646499
IF MODE,COUNTEND THEN % INPUT WAS TOO LONG SOME LEFT 03646500
IF UPQFLAG THEN % QM OR NOT TWX          03647000
BEGIN % SCAN TO ETX OR CR                03647500
SCANIT: J#H+1;                             03648000
  SCANINPUT;                              03648500
  IF NOT MODE,CREND THEN                  03649000
  BEGIN UPQ;                              03649500
    IF MODE THEN ELSE GO SCANIT; % COUNTEND = TRY AGAIN 03650000

```

```

        END;
        END ELSE STABLE[STA],OWHROWOUT:=1; % TWX = DISCARD BUFFS
        IF MODE,CREND THEN GO TO CR ELSE GO TO START;
$ POP OMIT
$ SET OMIT = NOT(TWXONLY)
        IF MODE,COUNTEND THEN STABLE[STA],OWHROWOUT:=1;
        GO TO START;
$ POP OMIT
%
        END OF INPUT HANDLING;
%
%       WHEN INPUT QUEUE IS EMPTY, DETANK OUTPUT
%
        IF (STAI=HEADOUT) # 0 THEN % OUTPUT TO DETANK
        BEGIN
            IF H=0 THEN H:=SPACE(60);
            IF NOT TANKS[STA],[1:1] THEN
                SLEEP([TANKS[STA]],NABS(0));
            STAB:=STABLE[STA];
$ SET OMIT = TWXONLY
            IF STAB,STATIONTYPE#TWX AND STAB,[CF]#0
            AND SEQARRAY[STA],SELECTED THEN GO TO FORGIT;
$ POP OMIT
            T:=TANKS[STA]:=ABS(*P(DUP));
$ SET OMIT = TWXONLY
            IF STAB,NAKKER THEN % NAKKED = RESET TANK
            BEGIN NT1:=TNAOG[SEQARRAY[STA],[26:6]];
                IF (NT2:=T,[14:5]=NT1,[1:5]) LSS 0 THEN
                    NT2:=NT2 + 32;
                    T:=TANKS[STA]:=T&(T,[27:6]+NT2)[27:42:6]
                    &NT1[14:1:13]&0[12:47:1]);
                STABLE[STA],NAKKER:=0;
            END;
$ POP OMIT
            IF T1:=(T,[SOUSE]=0) THEN %NUTHIN ON DISK
            IF T1:=(T,[CF]=0) OR STAB,SWAPPED
            THEN GO DELINKIT ELSE
            IF T1:=(M[T]=0) THEN GO DELINKIT ELSE
            BEGIN
$ SET OMIT = TWXONLY
                IF STAB,STATIONTYPE NEQ TWX THEN
                    BEGIN DISKWAIT(T,[CF],30*GLOMSIZE*STA+
                    TANKS[STA],[14:5]+TANKADDRESS);
$ SET OMIT = NOT(STATISTICS) OR OMIT
                    COUNTUP(16,1);
$ POP OMIT
                    T:=TANKS[STA]:=(*P(DUP))&1[19:34:14]; %TANK
            END;
$ POP OMIT
                    I:=M[J]=T,[CF] INX 1; %TAKE IT
                    M[J]=0; %FROM CORE
            END ELSE % DRAIN THE DISK
            BEGIN J:=T,[19:8];
                DISKWAIT(=H,30*GLOMSIZE*STA+T,[14:5]+
                TANKADDRESS);
$ SET OMIT = NOT(STATISTICS)
                    COUNTUP(16,1);
$ POP OMIT
$ SET OMIT = TWXONLY
                    IF STAB,STATIONTYPE = TWX THEN

```

```

03650500
03651000
03651500
03651501
03651999
03652000
03652500
03652501
03653000
03653500
03654000
03654500
03655000
03655500
03656000
03656500
03657000
03657500
03658000
03658099
03658100
03658200
03658201
03658500
03658549
03658550
03658600
03658650
03658700
03658750
03658800
03658850
03658900
03658901
03659000
03659500
03659600
03660000
03660500
03660590
03660600
03661000
03661500
03661509
03661510
03661511
03662000
03662500
03662501
03663000
03663500
03664000
03664500
03665000
03665500
03665509
03665510
03665511
03665590
03666000

```

```

BEGIN
$ POP OMIT
I:=M[H] = J + 1;
$ SET OMIT = TWXONLY
IF I > 56*(R:=1+(LINETABLE[
IF STA>LMAX THEN STATAB,LEENKER ELSE
$ POP OMIT
    STA],[6:3] > 2)) THEN
    BEGIN I:=56*R;      %TWX IT OUT
          T:=T&(T,[19:5]+R*7)[19:43:5];
    END ELSE          %IN BUFFER SIZE HUNKS
          T:=T&0[19:40:8]&(T,[SOUSE]=1)[27:42:6]&
          (T,[14:5]+1)[14:43:5]; %KEEP THE
TANKS[STA]:=(*P(DUP))&T[14:14:19]; %POT
%OWH DETANKS TWX ONLY
J:=(H+J,[40:5])&J[30:45:3]; %RIGHT
$ SET OMIT = TWXONLY
END ELSE          %LET DCWRITE DO IT
    BEGIN
      I:=M[H],[CF]; %DONT HARRY HARRY
      J:=(H+J,[40:5])&J[30:45:3];
    END;
$ POP OMIT
    END; %OF TAKIN NOW SOME GIVIN
    DELINKIT;
$ SET OMIT = TWXONLY
    IF NOT T1 THEN
$ POP OMIT
        R:=
            DCWRITE(J+1,STA,I);
            STATABLE[STA],OUTPUTANKING:=(T,TANKN # 0) OR
            (M[T] NEQ 0 AND (NOT STATABLE[STA],SWAPPED) AND
            T,[CF] NEQ 0);
            %CHECK FOR MOAR:IN COAR OR DISK:TANKS
            IF T,[10:2] NEQ 0 THEN %OPEN THE DOAR FOR MOAR
                IF GLOMSIZE/5 GEQ T,[SOUSE] THEN
                    BEGIN %WHRN TANK STARTS GETTING EMPTY,REFILL OIT
                        TANKS[STA]:=(*P(DUP))&0[10:46:2];
                        IF T,[10:1] THEN
                            BEGIN M[(R:=GETAREA(0))+1]:=0;
                                    M[R]:=0&STA[CTF]&10[18:41:7];
                                    QUEVENT(R,CANDEMIX[STA]);
                            END;
                        IF T,[11:1] THEN
                            IF CANDEMIX[STA] NEQ STATAB,MIXNR
                                THEN BRINGBACK(STATAB,MIXNR);
                        END;
                    HEADOUT:=TANKS[STA],[2:8];
                    IF STA = TAILOUT THEN TAILOUT:=0;
                    TANKS[STA]:=NABS((*P(DUP))&0[2:40:8]);
                    GO TO RESTART;
                END OF OUTPUTTER GETTING;
                IF H#0 THEN FORGETSPACE(H);
                WORKING:=FALSE;
                GO TO NOTHINGTODO;
            END OLDEMACIATEDWIERDHAROLD SURE IS SKINNY;
            SAVE PROCEDURE GIVEAWAY(A);
            VALUE A;
            REAL A;
            %A IS THE ADDRESS OF THE LAST BUFFER SEGMENT OF A MESSAGE(THE EOM

```

```

03666500
03666501
03666600
03667000
03667400
03667500
03667600
03668000
03668500
03669000
03669500
03670000
03670500
03671000
03671500
03672000
03672490
03672500
03673000
03673500
03674000
03674500
03674501
03675000
03675500
03676000
03676090
03676100
03676110
03676200
03676500
03677000
03677500
03678000
03678500
03679000
03679500
03680000
03680500
03681000
03681500
03682000
03682500
03683000
03683500
03684000
03684500
03685000
03685500
03686000
03686500
03687000
03687500
03688000
03688500
03688600
03689000
03690000
03691000
03692000

```

```

% IS IN THIS BUFFER) TO BE GIVEN TO WORKER, GIVEAWAY LINKS THE
% ENTIRE MESSAGE INTO WORKERS QUEUE TO BE GIVEN TO A USER JOB OR
% CANDE, FOLLOWING BUFFER CONSOLIDATION,
BEGIN
  INTEGER I;
  I:=A;
  WHILE P(M[A],[FF],DUP) # 0 DO A := P;
  P(DEL);
  IF WORKERINQ,[CF] = 0 THEN
  WORKERINQ := A & I[CTF]
  ELSE
  BEGIN
    M[WORKERINQ,[FF]] := (*P(DUP)) & A[CTC];
    WORKERINQ := WORKERINQ & I[CTF];
  END;
  STARTWORKING;
END GIVEAWAY;
PROCEDURE DCIOFINISH(R);
VALUE R;
REAL R;
BEGIN
  REAL T=NT1,S=NT2,S1=NT3,V=NT4,E=NT5,ST=NT6,I=NT7,D,L;
  LABEL BRAKE,DISCONN,DELINK,JUNKET,BIGJOKE,READRDY,AUT,RDRDYABN,GONE;
$ SET OMIT = TWXONLY
  MSGACK,EOTIT,ACK,SELECTNAK,ENQRD,MSGANS,NOTHERE,BREAKIT,FORGETNQUE,
  EOT,POLLBACK,WAITENQ,ENQACK,ENQNAK,COONGRILL,AUS,ZEITAU,
$ POP OMIT
  FINIS,BREAKER,ERROR,DSIT,ACTINT,WRITERDY,FINISH,FOUND,QUEIT,IQUEIT;
$ SET OMIT = NOT(TWXONLY)
  SWITCH READACTION:=READRDY,BIGJOKE,AUT,AUT,BREAKER,AUT,AUT,AUT,
  RDRDYABN,AUT;
$ POP OMIT
$ SET OMIT = TWXONLY
  SWITCH READACTION:=READRDY,BIGJOKE,MSGANS,ENQRD,BREAKIT,NOTHERE,
  ZEITAU,EOT,RDRDYABN,MSGANS;
  SWITCH WRITEACTION=WRITERDY,POLLBACK,
  BREAKER,BREAKER,POLLBACK,ENQACK,ENQNAK,BREAKER;
$ POP OMIT
  DEFINE INTERROGATE = R,[18:1]#,
  READREADY= R,[24:1]#,
  GROUPMARK = (NOT R,[25:1])#,
  LINKEM= 25:40:8#,
  PTR= 26:6#,
  RDRDY= 43:1#,
  MSGSTAT= 5:5#,
  WRITEREADY= R,[27:1]#,
  ABNORMALFLAG= R,[23:1]#,
  IDLE= ((R,[23:10] AND @1475)=0)#,
  IDLEABNORMAL = ((R,[23:10] AND @1475)=@1000)#,
  NOTREADY = ((R,[24:9] AND @435)=@24)#,
  BUSY= ((R,[27:6] AND @75)=@20)#,
  LAST = #;
  SUBROUTINE HITHERE;
  BEGIN
    M[(V+GETAREA(0))+CLOCK+P(RTR)];
    M[V+1]+ST;
    M[V+2]+S1;
    FORK(P(,HELLO),V,=6,128,1);
  END;
  SUBROUTINE FORGETI;

```

DC IO FINISH

```

03693000
03694000
03695000
03696000
03697000
03698000
03699000
03700000
03701000
03702000
03703000
03704000
03705000
03706000
03707000
03708000
03709000
03710000
03710500
03711000
03711500
03712000
03712500
03712995
03713000
03713500
03713505
03714000
03714495
03714500
03715000
03715005
03715495
03715500
03716000
03716500
03717000
03717005
03717500
03718000
03718500
03719000
03719500
03720000
03720500
03721000
03721500
03722000
03722500
03723000
03723500
03724000
03724500
03725000
03725500
03726000
03726500
03727000
03727500
03728000

```

```

BEGIN
  FORGETAREA(M[I],[2:2],I);
END;
SUBROUTINE SEQUENCE;
BEGIN
  IF (E:=SEQARRAY[ST])# 0 THEN
  BEGIN
    STREAM(E:=E,[2:27],V:=(V:=GETAREA(0))+1);
    BEGIN
      SI*LOC E;DI*DI+16;
      DS*8 DEC;DS*LIT " ";
      DI*DI*9;DS*7 FILL;
      E*DI;DI*V;
      SI*E;DS*9 CHR;
    END;
    ENTERLINEQ(V,S1,0);
  END;
  M[V],[4:1]i=1;
END;
SUBROUTINE JUNKER;
BEGIN
  IF I,[CF]#0 THEN
  BEGIN LINETABLE[S1]i=I&0[CTC];
    WHILE (V:=M[I]),[FF]#0 DO
    BEGIN FORGETAREA(V,[2:2],I);
      I:=V,[FF];
    END END;
  M[V:=GETAREA(0)]i=(P(DUP))&1[4:47:1]&ST[10:40:8];
  GIVEAWAY(V);
END;
SUBROUTINE DETANK;
BEGIN
  IF STABLE[ST],OUTPUTANKING THEN
  IF TANKLINE[ST]=0 THEN
  IF TAILOUT#ST THEN
  BEGIN
    TANKLINE[TAILOUT]*ST;
    TAILOUT*ST;
    STARTWORKING;
  END; END;
$ SET OMIT = TWXONLY
SUBROUTINE NAKHANDLER;
BEGIN
  P([TNAOG[SEQARRAY[ST],[PTR]]],DUP,LOD,DUP);
  IF T=TC500 THEN P(P&(P,[38:10]=1)[38:38:10]) ELSE
    P(P&P[42:36:16]&0[28:40:8]);
  P(XCH,STD);
  TANKS[ST],[13:1]i=1;
  DETANK;
  LI=IDL;
END;
$ POP OMIT
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX START
EI=1; *INITIALIZE ERROR COUNT, ETC
UNIT[30],[16:2]i=0;
IF REMOTE THEN ELSE GO TO FINISH;
S1*LLNR;
IF INTERROGATE THEN
IF LLNR=0 THEN *PASSIVE VARIETY
BEGIN

```

```

03728500
03729000
03729500
03730000
03730500
03731000
03731500
03732000
03732500
03733000
03733500
03734000
03734500
03735000
03735500
03736000
03736500
03737000
03737100
03737200
03737300
03737400
03737500
03737600
03737700
03737800
03737900
03738000
03738100
03738200
03738400
03738500
03739000
03739500
03740000
03740500
03741000
03741500
03742000
03742500
03742590
03742600
03742650
03742700
03742750
03742800
03742850
03742900
03742950
03743000
03743050
03743060
03743250
03743500
03744000
03745000
03745500
03746000
03746500
03747000

```

I:=R.[9:9];	03747500
FOR S1+1 STEP 1 UNTIL LMAX DO	03748000
IF I#LINETABLE[S1].[9:9] THEN ELSE GO FOUND;	03748500
\$ SET OMIT = NOT SEPTICTANK	03748700
DISPOSAL(32,IOQUE[S],R);	03748750
\$ POP OMIT	03748800
E:=0;GO ERROR;	03749000
END;	03749500
FOUND:	03750000
DI=P(LINETABLE[S1],DUP),LINEDIS; LI=P.[21:5];   %LSTATUS	03750500
\$ SET OMIT = NOT SEPTICTANK	03750700
DISPOSAL(L,IOQUE[S],R);	03750750
\$ POP OMIT	03750800
TI=STABLE[ST];	03751000
\$ SET OMIT = TWXONLY	03751495
IF D=MULTI THEN SEQARRAY[S1],[40:8] ELSE	03751500
\$ POP OMIT	03751505
S1].STATIONTYPE;	03752000
\$ SET OMIT = NOT(SAVERESULTS OR DEBUGGING)	03752495
STORAWAY:=R; STORAWAY:=STABLE[ST]; STORAWAY:=LINETABLE[S1];	03752500
\$ POP OMIT	03752505
IF LOCKED[S1] THEN	03753000
BEGIN	03753500
M[LINETABLE[S1]]←NABS(R);	03754000
GO TO AUT;	03754500
END;	03755000
IF INTERROGATE THEN	03755500
BEGIN	03756000
IF LLNR = 0 THEN	03756500
BEGIN% PASSIVE INTERROGATE IN ANSWER TO DC INTERRUPT	03757000
IF READREADY THEN	03757500
BEGIN	03758000
INITIATEDCIO((@0400000004000000 & R[9:9:9]) OR	03758500
(GETAREA(LINETABLE[S1],BUFSIZE)+1), S);	03759000
LLNR←S1;	03760000
\$ SET OMIT = TWXONLY	03760495
IF D=TWX THEN	03760500
BEGIN	03761000
\$ POP OMIT	03761005
IF ABNORMALFLAG THEN	03761500
IF L#WRB OR LINETABLE[S1].READYQD THEN LI=RRA	03762000
ELSE BEGIN	03762500
M[I:=GETAREA(0)]:=(*P(DUP))&	03763000
(ST&2[33:41:7])[CTF];	03763500
L←BROKEN;	03764000
GO DELINK;	03764500
END ELSE	03765000
L←IF L=IDL THEN FIRSTIME ELSE NORMAL;	03765500
IF NOT STABLE[ST].DIALEDUP THEN HITHERE;	03766000
\$ SET OMIT = TWXONLY	03766495
END ELSE%CONTENTION OR MULTIPOINT	03766500
BEGIN	03767000
IF ABNORMALFLAG THEN	03767500
IF NOT (E←LINETABLE[S1]).[20:1] THEN	03768000
BEGIN	03768500
LI:=IF L=WRB THEN 57 ELSE L&6[42:45:3];	03769000
LINETABLE[S1]:=E&L[20:42:6];	03769500
IF L LEQ 49 THEN GO JUNKET ELSE GO FINISH;	03770000
END ELSE GO FINISH ELSE	03770500
BEGIN	03771000



	IF BUSY THEN	03771100
	IF L=SELECT THEN	03771200
	BEGIN LINETABLE[S1],[20:1]:=1;	03771300
	P(SELECTANS);	03771400
	END ELSE	03771500
	IF L=WRB AND T=TC500 THEN P(MSGANSWER)	03771600
	ELSE	03771700
	IF L,[43:3]=0 THEN P(L&5[43:45:3]) ELSE	03771800
	IF L=IDLPOLLING THEN P(POLLTIMEOUT) ELSE	03771900
	P(TIMEDOUT)	03772000
	ELSE	03772100
	IF L=WRB THEN P(MSGANSWER) ELSE	03772200
	P(L&2[43:46:2]);	03772300
	L:=P;	03772400
	END;	03772500
	END;	03773000
\$ POP OMIT	GO TO FINIS;	03773005
	END;	03773500
	IF WRITEREADY THEN	03774000
	BEGIN	03774500
	IF ABNORMALFLAG THEN	03775000
	BEGIN	03775500
\$ SET OMIT = TWXONLY	IF D=MULTI THEN	03776000
	BEGIN SEQARRAY[ST],[4:2]:=0;	03776495
	GO TO IQUEIT;	03776500
	END;	03777000
	IF D=TWX THEN	03777500
\$ POP OMIT	IF STABLE[ST],DIALEDUP THEN	03778000
	BEGIN	03778500
DSIT:	M[1+GETAREA(0)]+(*P(DUP))&(ST&3[33:41:7])	03778505
	[CTF];	03779000
	GO DELINK;	03779500
	END;	03780000
	HITHERE;	03780500
	L:=IDL;	03781000
\$ SET OMIT = TWXONLY	END ELSE	03781500
	IF D=CONTENTION THEN	03782000
	BEGIN	03782500
	L:=GETAREA(0);	03782995
	GO TO EOTIT;	03783000
	END ELSE	03783500
\$ POP OMIT	GO TO AUT;	03784000
	END;	03784500
	L=WAITING;	03785000
	IF STABLE[ST],[CF]=0 THEN GO AUT ELSE GO QUEIT;	03785005
	END;	03785500
	IF L = DISCON THEN GO TO AUT;	03786000
	IF IDLE THEN	03786500
\$ SET OMIT = TWXONLY	BEGIN	03787000
	IF (V:=SEQARRAY[ST]),[3:1] THEN	03787500
	BEGIN	03787600
	SEQARRAY[ST]:=V&0[3:47:1];	03788000
	GO TO AUT;	03788495
	END;	03788500
	END;	03789000
	END;	03790000
	L:=V&0[3:47:1];	03790500
	GO TO AUT;	03791000
	END;	03791500
	L:=P;	03792000

```

L←WAITINGENQ; 03792500
GO QUEIT; 03793500
END ELSE 03794000
IF D=MULTI AND L≠WRBUSY THEN GO AUT ELSE 03794100
$ POP OMIT 03794105
GO IQUEIT; 03794500
%SEQUENCE OF EVENTS AFTER DISCONNECT; 03795000
%1 = NOTIFY C&E OF DISCONNECT 03795500
%2 = MARK DISCONNECTED, STOP FURTHER IO FOR THIS LINE 03796000
%3 = DELETE CORE QUEUES, ENTER MARKER IN C&E TANK 03796500
%4 = FORK TO QUITTER 03797000
%5 = CALL TERMINATE ON USER JOB (IF NECESSARY) 03797500
%6 = C&E COMMUNICATES ACK. OF DS, MARKER, LOG=OFF, ETC, 03798000
%7 = DELETE DISK TANKS, CLEAR TABLES, UPDATE SYSTEMDISK 03798500
IF IDLEABNORMAL OR NOTREADY THEN 03799000
BEGIN 03799500
$ SET OMIT = TWXONLY 03799799
IF D=MULTI THEN 03799800
IF L=SELECT OR L=WRB THEN 03799900
BEGIN LI=IF L=SELECT THEN SELECTANS ELSE MSGANSWER; 03800000
GO TO ACTINT; 03800100
END ELSE GO TO GONE; 03800200
$ POP OMIT 03800210
DISCONN; 03800500
L←DISCON; 03801000
STATABLE[ST],[15:2]:=0; 03801500
M[I←GETAREA(0)]←(*P(DUP))&(ST&6[33:41:7])[CTF]; 03802000
IF CANDYINX=0 OR (ST=ABS(SPOWORD) AND L≠DISCON) THEN 03802500
FORGETI ELSE QUEVENT(I,CANDYINX); 03803000
FORK(P(,QUITTER),S1&ST[CTF],-2,96,1); 03803500
IF LINETABLE[S1],READYQED THEN 03804000
BEGIN 03804500
LINETABLE[S1],READYQED:=0; 03805000
IF (I:=(V:=STATABLE[0]),[CF])=S1 THEN 03805500
STATABLE[0]:=V&STATABLE[S1][LINKEM] ELSE 03806000
BEGIN 03806500
WHILE P(STATABLE[I],DUP),LEENKER≠S1 DO 03807000
I:=P,LEENKER; 03807500
STATABLE[I]:=P(XCH)&STATABLE[S1][LINKEM]; 03808000
IF V,[FF]=S1 THEN 03808500
STATABLE[0]=V&I[CTF]; 03809000
END; 03809500
STATABLE[S1],LEENKER:=0; 03810000
END; 03810500
GO TO AUT; 03811000
END; 03811500
IF BUSY THEN 03812000
BEGIN 03812090
$ SET OMIT = TWXONLY 03812100
IF D=MULTI THEN 03812200
IF L LSS WRBUSY THEN GO AUT; 03812210
$ POP OMIT 03813000
IF NOT (I:=LINETABLE[S1]),[20:1] THEN JUNKER; 03818500
IF L≠WRB THEN % IDLE OR READING 03818750
BEGIN FORK(P(,DCBUSY),I&S1[CTF],1,128,1); 03819000
L←WRBUSY; 03819250
GO TO ACTINT; 03819500
END; 03819750
EI=4; % FALLS THRU TO ERROR 03820000
END; 03820500
% DCBUSY TRIES TO EXPLAIN WHAT HAPPENED TO THE

```

<pre> ERROR:      % KLUTZ ON THE OTHER END OR THE LINE.             FORK(PC,DCERR),R&amp;E[4:44:4],1,90,0);             IF E=0 THEN             BEGIN NEXTDCIO;               GO TO FINISH;             END ELSE GO AUT;           END;           %ACTIVE INTERROGATE           IF NOTREADY THEN GO TO DISCONN;           IF L = RRA THEN           BEGIN             LSTATUS[S1]:=IDL;             IF PAPERTAPE(S1) THEN NEXTDCIO ELSE             BEGIN               M[(V+GETAREA(0))+1]+@2425435736743700;               ENTERLINEQ(V,S1,1);               SEQUENCE;             END;           END;           JUNKET:             I:=LINETABLE[S1];             JUNKER;             IF I,[CF]#0 THEN FORGETI;             GO TO FINISH;           END;           \$ SET OMIT = TWXONLY             IF L=SELECTANS OR L=MSGANSWER THEN             BEGIN I:=GETAREA(0);               GO TO SELECTNAK;             END;           \$ POP OMIT             IF L = WRBUSY THEN %BUSY INTERRUPT             IF IDLE THEN                                %LOST LEFT ARROW             GO IQUEIT ELSE             THROWAWAY[S1]+TRUE;             GO AUT;           END;           % READ OR WRITE           I ← IOQUE[S],[CF]=1;           IF R,[CF] = I+1 THEN           BEGIN %READ OR WRITE BOUNCED BACK             IF R,[30:1] THEN           \$ SET OMIT = TWXONLY             IF D=MULTI THEN             IF STABLE[ST],DIALEDUP THEN GO TO DISCONN ELSE             BEGIN STABLE[ST],[CF]:=0;               LI=SEQARRAY[ST]+(*P(DUP))&amp;1[3:47:1];               SEQARRAY[S1]+(*P(DUP))&amp;L[40:32:8];               GO TO ENQNAK;             END ELSE           \$ POP OMIT             GO TO DISCONN;             IF R,[24:1] THEN             BEGIN %READ               FORGETI;E:=3;GO TO ERROR;             END;           \$ SET OMIT = TWXONLY             IF D=MULTI THEN             BEGIN EI=15;               GO TO ERROR;             END; </pre>	<pre> 03821000 03821500 03822000 03822100 03822200 03822300 03822500 03823000 03823500 03824000 03824500 03825000 03825500 03826000 03826500 03827000 03827500 03828000 03828500 03829000 03830000 03831000 03833000 03833500 03833590 03833600 03833700 03833800 03833900 03833910 03834000 03834500 03835000 03835500 03836000 03836500 03837000 03837500 03838000 03838500 03838600 03838695 03838700 03838800 03838900 03839000 03839100 03839200 03839300 03839305 03839400 03839500 03840000 03840500 03841000 03841490 03841500 03841600 03841700 03841800 </pre>
---	--

\$ POP OMIT	03841810
L:=IDL;	03841900
GO TO AUT;	03842000
END;	03842500
IF L,[RDRDY] THEN GO TO READACTION[L,[44:4]] ELSE	03843000
BEGIN STATABLE[ST],[CF]:=M[I];	03843500
\$ SET OMIT = NOT(TWXONLY)	03843995
GO TO WRITERDY;	03844000
\$ POP OMIT	03844005
\$ SET OMIT = TWXONLY	03844495
GO TO WRITEACTION[L];	03844500
\$ POP OMIT	03844505
END;	03845000
BIGJOKEI:	03845500
L:=NORMAL;	03846000
IF LINETABLE[S1],[20:1] THEN GO TO READRDY;	03846500
\$ SET OMIT = TWXONLY	03846995
IF D=TWX THEN	03847000
BEGIN	03847500
\$ POP OMIT	03847505
IF (P(M[I+1],TOP) AND P(XCH),[1:5]=@37)	03848000
THEN SEQARRAY[S1]+0;	03848500
\$ SET OMIT = TWXONLY	03848995
END ELSE	03849000
BEGIN	03849500
IF D=MULTI THEN	03850000
BEGIN	03850500
STREAM(I:=I+1;E:=E);	03851000
BEGIN	03851500
SI+I;SI+SI+1;DS+LIT "0";	03852000
IF SC="*" THEN	03852500
BEGIN	03853000
SI+SI+1;TALLY+2;	03853500
END;	03854000
DS+CHR;IF SC="*" THEN	03854500
BEGIN	03855000
SI+SI+1;TALLY+TALLY+1;	03855500
END;	03856000
I+TALLY;DS+CHR;	03856500
END;	03857000
EI=P&E[34:6:12]; ST:=S1;	03857500
DO IF P(SEQARRAY[ST],DUP),[12:14]#E	03858000
THEN ELSE GO TO AUS	03858500
UNTIL (ST+P,LINELINK)=S1;	03859000
EI=2;GO TO ERROR;	03859500
TI=STATABLE[ST],STATIONTYPE;	03860000
SEQARRAY[ST]:=P(XCH)&O[4:46:2];	03861000
SEQARRAY[S1],[40:8]+ST;	03861500
END ELSE	03862000
BEGIN %CONTENTION	03862500
STREAM(I:=I+1);	03863000
BEGIN SII=1;	03863500
IF SC # "*" THEN TALLY:=1;	03864000
II=TALLY;	03864500
END;	03865000
IF P THEN ELSE GO TO WAITENQ;	03865500
END;	03866000
IF NOT STATABLE[ST],DIALEDUP THEN HITHERE;	03866500
EI=@2251252142665164;    % "BREAKWRU"	03866900
STREAM(JI=T=BIDS;II:=I+1,E:=E);	03867000

AUS:

```

BEGIN SI:=1;
  10(IF SC # "" THEN SI:=SI+1 ELSE JUMP OUT);
  SI:=SI+1; IF SC#"" THEN SI:=SI+1 ELSE GO TO EXX;
  IF SC=MARK THEN
  BEGIN
    SI:=SI+1;
    IF 5 SC=DC THEN TALLY:=3 ELSE
    BEGIN
      SI:=SI+5;
      IF 3 SC=DC THEN TALLY:=2;
    END;
  END;
  J:=TALLY;
EXX:
  END;
  IF (E:=P)#0 THEN IF E=1 THEN % DISCARD ETX ONLY FOR BIDS
  BEGIN V:=1; LSTATUS[SI]:=IDL; GO TO ACK; END ELSE
  LINETABLE[SI]:=(P(DUP))&E[CTC]&1[20:47:1];
  END;
  $ POP OMIT
  READRDY:=
  M[I],[10:8]+ST;
  P([STABLE[ST]],IOR);
  IF GROUPMARK THEN
  BEGIN
    IF (E:=(L:=(LINETABLE[SI]:=(P(DUP))
    &IDL[21:43:5])),[20:1]) THEN
    BEGIN
      FORGETI;LINETABLE[SI]:=L&0[20:47:1];
    $ SET OMIT = TWXONLY
    IF D#TWX THEN
    BEGIN
      V:=GETAREA(0);
      IF L.[CF]=0 THEN ELSE GO ACK;
      IF (E+P(SEQARRAY[ST],DUP),NAKCNT+1)
      LSS P(XCH),NAKMAX THEN
      BEGIN
        M[V+1]+@0537000000000000; %NAK
        M[V],[MSGSTAT]+NAKING;
      END ELSE
      BEGIN
        E+0;
        M[V+1]:=NOT 0; % DISCONNECT
        M[V],[MSGSTAT]+DISCON;
      END;
      SEQARRAY[ST],NAKCNT+E;
      ENTERLINEQ(V,ST,1);
      GO FINISH;
    END ELSE
    $ POP OMIT
    GO TO AUT;
  END; %NEXT IS NO THROWAWAY
  V:=GETAREA(0);
  $ SET OMIT = TWXONLY
  IF D#TWX THEN
  BEGIN %CONTENTION OR MULTIPPOINT
  ACK:
    IF T # TC500 THEN
    TNAOG[SEQARRAY[ST],[PTR]],[28:20]:=0;
    SEQARRAY[ST],NAKCNT+0;

```

```

03867500
03868000
03868500
03869000
03869500
03870000
03871000
03871500
03872000
03872500
03873000
03873500
03874000
03874100
03874500
03874800
03874900
03875000
03875500
03875505
03876000
03876500
03877000
03877500
03878000
03878500
03879000
03879500
03880000
03880495
03880500
03881000
03881500
03882000
03882500
03883000
03883500
03884000
03884500
03885000
03885500
03886000
03886500
03887000
03887500
03888000
03888500
03889000
03889500
03889505
03890000
03890500
03891000
03891495
03891500
03892000
03892500
03893000
03893500
03894000

```

M[V+1]*#3437000000000000; *ACK	03894500
M[V],[MSGSTAT]*ACKING;	03895000
ENTERLINEQ(V,S1,1);	03895500
IF E THEN GO FINISH;	03896000
END ELSE	03896500
\$ POP OMIT	03896505
IF PAPER TAPE[S1] THEN	03897000
BEGIN	03897500
NEXTDCIO;	03898000
FORGETAREA(0,V);	03898500
END ELSE	03899000
BEGIN	03899500
STREAM(V+V+1);	03900000
DS=4LIT"≤<#<"	03900500
ENTERLINEQ(V,S1,1);	03901000
SEQARRAY[S1]:=(E1=*P(DUP))&	03901500
(E,[21:27]+E,[21:19])[21:21:27];	03902000
SEQUENCE;	03902500
END;	03903000
M[I]:=(*P(DUP))&1[5:47:1]J&(V:=LINETABLE[S1])[CTF];	03903500
IF V,[CF]#0 THEN M[V],[CF]*I;	03904000
LINETABLE[S1]*V&0[CTC];	03904500
GIVEAWAY(I);	03905000
GO FINISH;	03905500
END; *NEXT IS IFAL ENDING=MORE TO COME	03906000
IF THROWAWAY[S1] THEN	03906500
BEGIN	03907000
FORGETI;GO TO AUT;	03907500
END;	03908000
IF (V+LINETABLE[S1]),[CF]#0 THEN	03908500
BEGIN	03909000
M[V],[CF]*I;	03909500
IF (E+M[V],[7:3]+1)*(8 DIV (V,BUFSIZE+1)) THEN	03910000
BEGIN	03910500
M[I]+(*P(DUP))&V[CTF];	03911000
LINETABLE[S1]*V&0[CTC];	03911500
GIVEAWAY(I);	03912000
GO TO AUT;	03912500
END;	03913000
END	03913500
ELSE E*1;	03914000
M[I]+(*P(DUP))&E[7:45:3]J&V[CTF];	03914500
LINETABLE[S1]*V&I[CTC];	03915000
AUT:; NEXTDCIO;	03915500
FINIS:; LSTATUS[S1]*L;	03916000
GO TO FINISH;	03916500
\$ SET OMIT = TWXONLY	03916995
ENQRD:;	03917000
STREAM(V:=I+1);	03917500
BEGIN	03918000
S1:=V; IF SC="#<" THEN S1:=S1+4; *IGNORE HEADER	03918500
IF SC="#<" THEN TALLY:=3 ELSE	03919000
IF SC # "<" THEN TALLY:=1;	03919500
V:=TALLY;	03920000
END;	03920500
IF (V:=P) OR (E:=LINETABLE[S1]),[20:1] THEN	03921000
IF V=3 THEN GO TO EOT ELSE	03921500
BEGIN	03922000
LINETABLE[S1]:=E&0[20:47:1];	03922500
M[I],[5:5]:=POLLING;	03923000

M[I+1]:=#0537000000000000; %NAK	03923500
END ELSE IF NOT TANKOK[ST] THEN	03924000
BEGIN	03924500
M[I],[5:5]:=ACKINGENQ;	03925000
M[I+1]:=#3437000000000000; %ACK	03925500
END ELSE	03926000
BEGIN	03926500
EOTIT:	03927000
M[I],[5:5]:=POLLING;	03927500
STREAM(I:=I+1);DS:=2 LIT "S";	03928000
END;	03928500
COONGRILL:	03929000
LSTATUS[S1]+L+IDL;	03929500
ENTERLINEQ(I,S1,1);	03930000
GO FINISH;	03930500
MSGANS:;	03931000
IF (E1=LINETABLE[S1]),[20:1] THEN	03931500
BEGIN	03932000
LINETABLE[S1]:=E&O[20:47:1];	03932500
GO TO SELECTNAK;	03933000
END;	03933500
IF D=CONTENTION THEN SEQARRAY[ST],[4:1]:=0;	03933900
STREAM(A:=I+1);	03934000
BEGIN SI:=A; IF SC="#*" THEN SI:=SI+4; %SKIP HEADING	03934100
IF SC = LEFTARROW THEN TALLY:=2;	03934150
IF SC="&" THEN TALLY:=1; A:=TALLY;	03934200
END;	03934300
IF (E1=P)=2 AND D=MULTI AND L THEN	03934310
BEGIN STABLE[ST]:=(V:=*P(DUP))&V[23:22:1]&1[24:47:1]&1[21:47:1]	03934320
&O[CTC];	03934330
WHILE (E:=V,[CF]) NEQ 0 DO FORGETAREA((V:=M[E]),[2:2],E);	03934340
NAKHANDLER;	03934350
LSTATUS[S1]:=IDL;	03934360
M[I]:=O&SELECT[5:43:5];	03934370
STREAM(A:=E:=SEQARRAY[S1],C1:=E,[24:1],C2:=E,[25:1],	03934380
C3:=(E,[24:1] NEQ E,[25:1]),R:=I+1);	03934390
BEGIN SI:=LOC A; SI:=SI+2; DS:=LIT "S";	03934400
C1(DS:=LIT "#"); DS:=CHR;	03934410
C2(DS:=LIT "#"); DS:=CHR;	03934420
C3(DS:=LIT "#"); DS:=2 LIT "Q%";	03934430
DS:=LIT LEFTARROW;	03934440
END;	03934450
ENTERLINEQ(I,S1,1);	03934460
GO TO FINISH;	03934470
END;	03934480
IF E THEN	03934490
BEGIN %ACKED THE WRITE	03934500
IF L THEN	03935000
BEGIN %IT WAS A MESSAGE	03935500
MSGACK:	03936000
E1=(TNAOG[V:=SEQARRAY[ST],[PTR]]:=	03936500
(*P(DUP))&TANKS[ST][1:14:13]);	03937000
STABLE[ST]:=(*P(DUP))&P(DUP)[22:23:1];	03937500
IF T = TC500 THEN	03938000
BEGIN IF D*MULTI THEN SEQARRAY[ST],[3:1]:=1;	03938500
DETANK;	03939000
END ELSE	03939500
IF ST = ABS(SPOWORD) THEN	03939600
BEGIN TNAOG[V]:=E&1[42:42:6];	03939700
DETANK;	03939800

END ELSE	03939900
IF (TNAUG[V]=E&E[36:42:6]),[42:6]#E,[14:6] THEN DETANK;	03940000
SEQARRAY[ST]=ABS(*P(DUP));	03940500
IF D = MULTI THEN GO TO ENQNAK ELSE GO TO EOTIT;	03941000
END; % ACK TO SELECT IS NEXT	03941500
SEQARRAY[ST]=(*P(DUP))&0[9:45:3]&0[4:46:2];	03942000
GO ENQNAK;	03942500
END; %NAKED IS NEXT	03943000
SELECTNAK;	03943500
STABLE[ST]=(*P(DUP))&V[23:22:1]&1[24:47:1]&1[21:47:1]&0[CTC];	03944000
WHILE (E=V,[CF])#0 DO FORGETAREA((V=M[E]),[2:2],E);	03944500
IF L THEN	03945000
BEGIN % NAK TO A MESSAGE	03945500
NAKHANDLER;	03946000
GO TO BREAKER;	03950000
END ELSE	03950500
BEGIN % NAK TO SELECTED	03951000
SEQARRAY[ST]=ABS(*P(DUP));	03951500
IF T # TC500 THEN	03952000
IF DISCONNECTING[ST] THEN	03952500
BEGIN	03953000
V:=GETAREA(0);	03953500
M[I]=(*P(DUP))&7[5:43:5]&V[CTC];	03954000
STREAM(I:=I+1);DS:=2 LIT"s+";	03954500
M[V],[5:5]=7;	03955000
M[V+1]=NOT 0;	03955500
GO COONGRILL;	03956000
END;	03956500
M[(E:=GETAREA(0))+1]=ST;	03957000
IF (V:=SEQARRAY[0]),[18:30]=0 THEN	03957500
FORK(P(,NAKQUE),0,1,96,1);	03958000
M[E+2]=CLOCK+P(RTR);	03958500
IF V,[FF]=0 THEN V:=V&E[CTF]	03959000
ELSE M[V,[3:15]],[CF]=E;	03959500
SEQARRAY[0]=V&E[3:33:15];	03960000
IF D=MULTI THEN GO ENQNAK ELSE GO EOTIT;	03960500
END;	03961000
NOTHERE::	03962000
IF D = MULTI THEN	03962100
BEGIN IF SEQARRAY[S1],LINELINK=S1 THEN % ONE STATION ON LINE	03962200
IF NOT STABLE[S1],DIALEDUP THEN	03962300
BEGIN SEQARRAY[S1],[3:3]=4; GO TO ENQNAK; END;	03962400
END ELSE IF T=TC500 THEN	03962500
BEGIN	03963000
WAITENQ;	03963500
L:=WAITINGENQ;	03964000
FORGETNQUE;	03964400
FORGETI;	03964500
GO QUEIT;	03965000
END;	03965500
ZEITAU::	03965900
FORK(P(,DCTIMEOUT),I&S1[CTF],1,96,1);	03966000
GO TO AUT;	03966500
EOT::	03967000
IF (E=LINETABLE[S1]),[CF] # 0 THEN	03967500
BEGIN	03968000
LINETABLE[S1]=E&0[CTC];	03968500
BREAKIT::	03969000
FORGETI;	03969500
IF D=TWX THEN GO AUT;	03970000



```

        IF E THEN GO BRAKE ELSE GO DSIT;
    END;
    DETANK;
    IF D=CONTENTION THEN
    IF T = TC500 THEN
    GO EOTIT ELSE GO WAITENQ
    ELSE GO TO ENQNAK;
    $ POP OMIT
RDRDYABN:;
    STREAM(T+I+1;Q+IF P(LINETABLE[S1],BUFSIZE,DUP)
    =2 THEN P+2 ELSE P+1);
    BEGIN
        SI+T;
        Q(28(IF SC="2" THEN
        BEGIN
            TALLY+1;
            JUMP OUT 2 TO LLLL;
        END ELSE
        IF SC="*" THEN JUMP OUT 2 TO LLLL ELSE
        SI+SI+1));
    LLLL: T+TALLY;
    END;
    IF P THEN %RECEIVED EOT OR EXC. PT.
    BEGIN
        FORGETI;
    ACTINT: LLNR:=S1;
        INITIATEDCIO(INTERROGATEMASK&R[9:9:9],S);
        GO TO FINIS;
    END ELSE
    GO TO READRDY;%THERE ARE 14 S IN THE BUFFER
WRITERDY:;
    P([STABLE[ST]],IOR);
    IF M[I],[4:1] THEN DETANK;
BREAKER:;
    FORGETI;
    GO TO AUT;
    $ SET OMIT = TWXONLY
POLLBACK:;
    IF SEQARRAY[ST],[4:2]=0 AND D=MULTI THEN
    BEGIN LI=IDL POLLING;
        IF STABLE[ST],[CF]#0 THEN GO FORGETNQUE ELSE GO BREAKER;
    END ELSE
    GO TO BREAKER;
ENQACK:;
    L+FIRSTIME;
    GO TO BREAKER;
ENQNAK:;
    FORGETI;
    $ POP OMIT
IQUETI:;
    L+IDL;
QUETI:;
    LSTATUS[S1]+L;
    ENTERREADYQ(ST);
FINISH:;
    IF UNIT[30],[15:3] = 0 THEN
    BEGIN
        UNIT[30],[18:15] + NOT 0;
        RETURNIOSPACE(S);
        IF FIRSTWAIT # NEXTWAIT THEN NEWIO;

```

```

03970500
03971000
03971500
03972000
03972500
03973000
03973500
03973505
03974000
03974500
03975000
03975500
03976000
03976500
03977000
03977500
03978000
03978500
03979000
03979500
03980000
03980500
03981000
03981500
03982000
03982500
03983000
03983500
03984000
03984500
03985000
03985500
03986000
03986500
03987000
03987500
03987995
03988000
03988500
03988600
03988700
03988800
03989000
03989500
03990000
03990500
03991000
03991500
03991505
03992000
03992500
03993000
03993500
03994000
03994500
03995000
03995500
03996000
03996500
03997500

```

END;	03998000
P(INI);	03998500
GO TO EXTERNAL;	03999000
END DCIOFINISH980;	03999500
SAVE PROCEDURE INITIATEIO(ICODESC,MIX,U);%	04000000
VALUE IODESC,MIX,U;%	04001000
REAL MIX,U;%	04002000
REAL IODESC;%	04003000
BEGIN LABEL EXIT;%	04004000
REAL PT=+1;	%R5904004100
P(TIO);	%R5904004200
\$ SET OMIT = NOT(SAVERESULTS OR DEBUGGING)	04004499
IF U=30 THEN STORAWAYI=ICODESC;	04004500
\$ POP OMIT	04004501
CHANNEL[P(DUP)]+U;	04005000
P([ICODESC],IIO);%	04006000
CHANIO[PT]=CLOCK+P(RTR);	%R5904007000
IF U < 16 THEN%	04008000
IF IODESC,[22:1] THEN%	04009000
BEGIN TRANSACTION[U] + IF IODESC,[18:1] THEN 0%	04010000
ELSE TRANSACTION[U]=1;%	04011000
GO TO EXIT;%	04012000
END;	04013000
TRANSACTION[U] + *P(DUP)+1;%	04014000
\$ SET OMIT = NOT(STATISTICS)	04014009
IF (U OR 1)=19 THEN	04014010
BEGIN COUNTUP(0,ICODESC,[27:6]);	04014020
COUNTUPBY(32,CLOCK+P(RTR)=DISKWAITIME[NT3]);	04014022
IF MIX NEQ 0 THEN COUNTUPBY(9,1);	04014025
COUNTUPBY(2,U=18);	04014030
U=M[ICODESC];	04014035
IF (MIXI=U,[6:6]) > 2 THEN COUNTUPBY(5,LEFTHALF1) ELSE	04014038
COUNTUPBY((MIX=2)+4,IF MIX THEN LEFTHALF1 ELSE 1);	04014040
STREAM(UI=[U]); BEGIN SII=U; DS:=8 OCT END;	04014045
MIXI=ICODESC,[27:6];	04014047
IF U GEQ MCPBASE AND U LEQ MCPTOP THEN COUNTUP(10,MIX);	04014050
IF U GEQ ESPDISKBOTTOM AND U LEQ ESPDISKTOP THEN	04014055
COUNTUP(11,MIX) ELSE IF U GEQ DIRECTORYTOP THEN	04014060
IF U LSS BYPASSBOTTOM THEN	04014065
COUNTUP(((U=DIRECTORYTOP),[44:4]=3)+25,MIX) ELSE	04014067
IF U LEQ DISKBOTTOM THEN COUNTUP(12,MIX);	04014070
COUNTUPBY(13,ICODESC,[CF] LSS FENCE);	04014080
COUNTUPBY(36-PT,[47:1],IF PT>3 THEN LEFTHALF1 ELSE 1);	04014085
END ELSE COUNTUPBY(59-PT,[47:1],IF PT>3 THEN LEFTHALF1 ELSE 1);	04014090
\$ POP OMIT	04014095
EXIT;END;	04014100
SAVE PROCEDURE WAITORSWAP(U,A); VALUE U,A; REAL U,A;	04014200
BEGIN REAL N;	04014300
LABEL L;	04014400
LI	04014500
BEGIN CLICK=CLOCK+P(RTR)+32;	04014600
SLEEP([M[A]],IOMASK);	04014700
IF (M[A] AND IOMASK)≠0 THEN P(XIT);	04014710
IF P1MIX≠0 THEN	04014800
BEGIN	04014900
IF (NT2+UNIT[U],[13:5])=16 OR NT2=31 THEN	04015000
IF [MEM[P1MIX,0]],[CF]≥FENCE THEN	04015100
BEGIN REPLY[P1MIX]+VOK; SWAP(WAITSWAP,1); END;	04015200
IF TERMSET(P1MIX) THEN	%R7004015290
IF (N+N+1)>9 THEN	04015300

```

IF U#25 THEN
BEGIN
UNIT[U]:=(*P(DUP))&LOCATQUE[
N:=P(DUP),[FF]][FTF]&0[13:13:5];
RETURNIOSPACE(N);
M[A]:=(*P(DUP))&1[30:47:1] OR IOMASK;
P(XIT);
END;
END;
END; GO TO L;
END;
SAVE PROCEDURE QUEUEUP(U); VALUE U ; REAL U;%
BEGIN IF U=30 THEN
WAITQUE[FIRSTWAIT:=(FIRSTWAIT+31) AND #1]:=U ELSE
BEGIN WAITQUE[NEXTWAIT] + U;%
NEXTWAIT + NEXTWAIT+1 AND 31;%
END;%
END;
$ SET OMIT = NOT(DFX)
SAVE PROCEDURE LINKEU;
BEGIN REAL EU=NT4;
IF EUW,[FF] GTR 1023 THEN EUW:=EU&EU[CTF] ELSE
EUQ[EUW,[CF]],[3:15]:=EUW:=EUW&EU[CTC];
END;
$ POP OMIT
SAVE PROCEDURE STARTIO(U); VALUE U; REAL U;%
BEGIN REAL T=NT1,R=NT2, S=NT3;%
$ SET OMIT = NOT(DFX)
LABEL BACK;
$ POP OMIT
IF (T + UNIT[U]),[13:5] = 0 THEN%
IF (S + T,[18:15]) < #1777 THEN%
$ SET OMIT = NOT(DFX)
BACK;
$ POP OMIT
BEGIN IF P(TIO) # 0 THEN%
BEGIN INITIATEIO(IOQUE[S],LOCATQUE[S],[3:5]
,U);%
P(3);%
END%
ELSE BEGIN QUEUEUP(U);%
P(4);%
END;%
P(T&P(XCH)[15:45:3],[UNIT[U]],+);%
$ SET OMIT = DFX
END;%
$ POP OMIT
$ SET OMIT = NOT(DFX)
END ELSE
IF (U AND #76)=18 THEN
BEGIN
IF EUW<#7777700000 THEN
BEGIN NT4+S+(T+T&EUQ[R+EUW,[FF]][18:18:30]),[FF];
EUW+ EUW & EUQ[R][18:3:15];
EUQ[R]=NABS(U);
IF U=19 THEN DO IOQUE[NT4],[3:5]+12
UNTIL (NT4+LOCATQUE[NT4],[FF])>1023;
GO BACK;
END;
DISKOUNT:=DISKOUNT+1;

```

```

04015310
%R7004015400
%R7004015410
%R7004015420
04015430
%R7004015460
%R7004015470
%R7004015480
04015500
04015600
04015700
04016000
04016100
04016200
04017000
04018000
04019000
04019100
04019499
04019500
04019600
04019700
04019800
04019900
04019901
04020000
04021000
04021099
%DFX04021100
04021101
04022000
04023000
04023099
%DFX04023100
04023101
04024000
04025000
04026000
04027000
04028000
04029000
04030000
04031000
04032000
04032999
04033000
04033001
04033049
%DFX04033050
%DFX04033100
04033150
%DFX04033200
%DFX04033300
%DFX04033400
%DFX04033500
%DFX04033510
%DFX04033520
%DFX04033600
%DFX04033700
04033800

```

END;	04033900
\$ POP OMIT	04033901
END;%	04034000
SAVE PROCEDURE PRINTERFINISH(U); VALUE U; REAL U;%	04035000
BEGIN	04036000
\$ SET OMIT = NOT(NEWLOGGING)	04036099
STOPLOG(P1MIX,0);	04036100
\$ POP OMIT	04036101
IF NOT UNIT[U],[16:1] THEN UNIT[U],[17:1] + 0;	04036200
STARTIO(U);%	04037000
GO TO EXTERNAL;%	04038000
END;%	04039000
SAVE PROCEDURE IOREQUEST(FINAL,IODESC,LOCATION);%	04040000
VALUE FINAL,IODESC,LOCATION;%	04041000
ARRAY FINAL,LOCATION[*];%	04042000
REAL IODESC;%	04043000
BEGIN REAL U=NT1,T=NT2,S=NT3,R=+1;%	04044000
\$ SET OMIT = NOT(DFX)	04044099
REAL EU=NT4;	%DFX04044100
LABEL FAKE,DOIT,IN;	%DFX04044200
\$ POP OMIT	04044201
IF IOQUESLOTS LEQ	04045000
(U:=IF LOCATION,[9:1] OR P1MIX=0 THEN 0 ELSE 7) THEN	04045100
SLEEP([IOQUESLOTS],@77-U);	04045200
IOQUEAVAIL + IOQUE[S:=IOQUEAVAIL];	04046000
\$ SET OMIT = NOT(STATISTICS)	04047009
DISKWAITIME[S]:=CLOCK+P(RTR);	04047010
\$ POP OMIT	04047011
\$ SET OMIT = NOT(DFX)	04047099
IF ((U=LOCATION,[12:6]) AND 62) = 18 THEN	%DFX04047100
BEGIN IF (T+EUQ[EU+M[IODESC],[6:6]])<0 THEN GO FAKE;	%DFX04047200
IF T,[2:1] THEN	04047300
IF DISKOUNT > 0 THEN	04047350
BEGIN U + 18;	04047400
IF P(RRR),[29:1] THEN	04047450
IN:	04047500
IF UNIT[U],[FF] > 1023 THEN	04047500
BEGIN DISKOUNT+DISKOUNT-1;	%DFX04047600
EUQ[EU]+NABS(T+U);	%DFX04047700
FAKE:	%DFX04047800
IODESC+IODESC&TINU[U+ABS(T)][3:3:5];	%DFX04047900
LOCATION,[12:6]+U;	%DFX04048000
GO DOIT	%DFX04048000
END;	04048050
IF U ≠ (U+19) THEN IF P(RRR),[28:1] THEN GO IN;	04048060
DISKOUNT+0;	%DFX04048100
END;	04048125
IF T,[FF]>@1777 THEN	%DFX04048150
IF (T:=T&S[CTF]&S[CTC]),[2:1] THEN LINKEU ELSE	ELSE04048200
LOCATQUE[T,[CF],[FF]+T+T&S[CTC];	%DFX04048450
EUQ[EU]+T;	%DFX04048500
END ELSE	%DFX04048550
BEGIN	%DFX04048600
DOIT:	%DFX04048650
IF (T+UNIT[U]),[13:5] = 0 THEN	04048651
\$ POP OMIT	04048701
\$ SET OMIT = NOT(DKBNODFX AND NOT DFX)	04048705
IF(U:=LOCATION,[12:6])=18 THEN	04048710
IF M[IODESC],[5:1] THEN	04048715
BEGIN	04048715
LOCATION,[12:6]:=U+19;	04048720
IODESC,[3:5]:=12;	04048725
END;	04048730

```

$ POP OMIT                                04048731
$ SET OMIT = DFX                          04048799
IF (T + UNIT[U + LOCATION,[12:6]],,[13:5] = 0 THEN 04048800
$ POP OMIT                                04048801
BEGIN IF P(T10) # 0 THEN%                 04049000
  BEGIN INITIATEIO( IODESC,P1MIX,U);%     04050000
  P(3);%                                   04051000
  END ELSE BEGIN QUEUEUP(U);%             04052000
  P(4);%                                   04053000
  END;%                                     04054000
  T + T&P(XCH)[15:45:3]&S[18:33:15];%    04055000
END ELSE%                                  04056000
  IF T,[FF]>1023 THEN                     04057000
  IF T,[13:5]=@31 THEN                   04057100
  BEGIN                                    04057200
    IOQUEAVAIL+S;                         04057300
    P(XIT)                                 04057400
  END ELSE%                               04057500
    T,[18:15] + S ELSE%                  04058000
    LOCATQUE[P(T,[33:15],DUP)]+LOCATQUE[R]&% 04059000
    S[18:33:15];%                        04060000
$ SET OMIT = NOT(DFX)                    04060099
  UNIT[U]+T&S[CTC];%                     %DFX04060100
END;%                                     %DFX04060200
$ POP OMIT                                04060201
IOQUESLOTS:=IOQUESLOTS-1;              04060500
LOCATQUE[S] + LOCATION&P1MIX[3:43:5] OR @7777700000;% 04061000
$ SET OMIT = DFX                          04061999
UNIT[U] + T&S[33:33:15];%              04062000
$ POP OMIT                                04062001
IOQUE[S] + IODESC;%                     04063000
IOCOUNT[P1MIX]+*P(DUP)+(T,[13:5]#16); 04063100
FINALQUE[S] + FINAL;%                   04064000
END;%                                     04065000
SAVE PROCEDURE FINISHOFFIO(U); VALUE U; REAL U;% 04067000
  BEGIN REAL T=NT1, FIN=NT3, V=NT4, IOD=NT6; 04068000
  LABEL ON,OFF,C0,C1,C2,C3,C4,C5,C6,C7;% 04069000
  SWITCH CSW + C0,C1,C2,C3,C4,C5,C6,C7;% 04070000
  IF FIN > 0 THEN%                       04071000
  IF FIN,[25:1] THEN%                    04072000
    BEGIN T + FIN,[3:5];%                 04073000
    FIN + FIN&IOD[3:3:5]&0[25:25:1];%    04074000
    GO TO CSW[T];%                       04075000
  END ELSE GO ON ELSE GO ON;%           04076000
C0: GO TO C0;%                            04077000
C1: FIN,[8:10] + V;%                      04078000
  GO TO C2;%                              04079000
C3: FIN,[8:10] + V;%                      04080000
C4: FIN + NOT V INX 1 INX FIN;%          04081000
  GO TO C5;%                              04082000
C6: STREAM(A+0:IOD);%                    04083000
  BEGIN DI + LOC A; SI + IOD; SI + SI+4; DS+4 OCT END;% 04084000
  T + P DIV 8-1;%                         04085000
OFF: FIN,[8:10] + T;%                     04086000
  GO TO C2;%                              04087000
C7: STREAM(A+0:IOD);%                    04088000
  BEGIN DI + LOC A; SI + IOD; DS + 4 OCT END;% 04089000
  T + P DIV 8-1;%                         04090000
  FIN + (NOT T INX 1 INX FIN)&T[8:38:10];% 04091000
  GO TO C5;%                              04092000

```

ON: IF U < 16 THEN%	04093000
IF IOD,[22:1] THEN%	04094000
C5: M[IOD INX 1] + M[NOT V INX IOD INX 1] + V%	04095000
ELSE%	04096000
C2: M[IOD INX NOT 0] + V;%	04097000
END;%	04098000
REAL MCP;	04098700
PROCEDURE PROGRAMRELEASE;%	04099000
BEGIN NAME T; REAL FSX=JUNK;	04100000
ARRAY R=-4[*];%	04101000
REAL IOD=NT1;%	04102000
ARRAY LOCN[*];%	04103000
REAL S;	04103050
CHECKSTACKSPACE;%	%WF 04103100
LOCN=M[S+(IF(IOD+NFLAG(MIP(T+[M[PRT[P1MIX,9]]],DUP,PRL))))	04104000
,[22:1] THEN 2 ELSE NOT 1) INX IOD];	04105000
IF IOD,[3:5]= 6 THEN	04105100
BEGIN; STREAM(SI=M[PRT[P1MIX,8]] INX P(DUP,0,XCH,DIA 10,	04105200
DIB 30,TRB 2),D+@600005);	04105300
BEGIN SI+S; DS+2 CHR END;	04105400
\$ SET OMIT = NOT(STATISTICS)	04105409
COUNTUP(27,IOD,[27:6]);	04105410
\$ POP OMIT	04105411
IF JUNK,[36:12]#45 AND RELTOG	04105500
OR M[IOD],[3:6] = 0 AND M[IOD] LSS (DIRDSK * DSKTOG) THEN	04105510
IF (USERCODE[P1MIX] EQV MCP) # NOT 0 THEN	04105550
BEGIN TERMINATE(P1MIX); TERMINALMESSAGE(30) END;	04105600
IF(FS[P1MIX,(FSX+P(*(NOT 2 INX LOCN),4,COC),[13:11]	04105650
DIV 5),[40:4])	04105700
AND TWO(IOD,[24:1]&FSX[43:44:4]))#0 THEN	04105750
BEGIN T[0]:=T[0]&1[19:47:1]&0[26:40:7];	04105800
M[(*(NOT 2)INX LOCN))INX 5 ]:= NABS(*P(DUP));	04105850
GO TO RETURN;	04105890
END;	04105900
IF NOT IOD,[24:1] THEN M[S],[11:1]+1;	04105950
END DISK BUSINESS;	04105990
IOREQUEST(R,IOD,LOCN);%	04106000
T[0],[19:1] + 0;	04107000
IF (NT1+P(*(NOT 2 INX LOCN),13,COC),[10:9]-1)#0 THEN%	04108000
STREAM(NT1,C+T[0],T);	04109000
BEGIN SI + T; SI + SI+8; DS + NT1 WDS;%	04110000
SI + LOC C; DS + WDS;%	04111000
END;%	04112000
GO TO RETURN;%	04113000
END;%	04114000
SAVE PROCEDURE NEWIO;%	04115000
BEGIN REAL S=NT3,U=NT4;%	04116000
S + UNIT[U-WAITQUE[FIRSTWAIT]], [18:15];%	04117000
INITIATEIO(IOQUE[S],LOCATQUE[S],[3:5],U);%	04118000
FIRSTWAIT + FIRSTWAIT+1 AND 31;%	04119000
UNIT[U],[13:5] + 3;%	04120000
END;%	04121000
REAL MDELTA = @11;%	04121050
REAL MLOG = @12;%	04121100
REAL MROW = @13;%	04121150
REAL LOGSIZE = @43;%	04121170
REAL LOGHOLDER = @56;%	04121200
REAL NUMAINTMESS = @57;%	04121250
REAL LOGENTRY = @63;%	04121300
REAL NXDISK = @76;%	04121350

ARRAY MAINTLOGARRAY = @77[*];%	04121400
PROCEDURE DISKORAUERROR(R); VALUE R; REAL R; FORWARD;	04121410
PROCEDURE ACTUALIOERR(R); VALUE R; REAL R; FORWARD;	04121425
PROCEDURE LINKUP(TYPE,KEY); VALUE TYPE,KEY; REAL TYPE,KEY; FORWARD;%	04121450
PROCEDURE CHECKJOBORFILEMESS(MIX,FIB,U);%	04121500
VALUE MIX,FIB,U; REAL MIX,FIB,U; FORWARD;%	04121550
PROCEDURE LOGOUTMAINT(B); VALUE B; REAL B; FORWARD;%	04121600
PROCEDURE MAINTLOGGER(B); VALUE B; REAL B; FORWARD;%	04121650
DEFINE	04121700
LOGVERSION=(	% VERSION NUMBER ON NEXT CARD
2	04121710
&	% CURRENT ENTRIES ON NEXT CARD
	04121720
21	04121730
[30;42;6]);%	04121740
TAPEBUFFERSIZE = 200#;	04121750
ARRAY MAINTBUFFER[*];	04121850
SAVE PROCEDURE IOFINISH(R,C); VALUE R,C; REAL R,C;	04121950
BEGIN BOOLEAN STOP;	04122000
COMMENT	04123000
WHEN E#0, STOP TAKES THE FOLLOWING VALUES;	04123010
0 DISK ERROR (OTHER THAN NOT READY ON A DFX SYSTEM)	04123020
1 OTHER ERROR	04123030
2 LOCKED ADDRESS (SHAREDISK)	04123040
3 ERROR WHEN UNIT[U],[5;8]#0	04123050
WHEN E=0, STOP TAKES THESE VALUES;	04123060
-2 IO FOR WHICH COMPLETE SHOULD NOT BE SET (DISK WRITE	04123070
BEFORE READ WITH UNIT OR EU SWITCH).	04123080
1 PRINTER IO,	04123090
0 NORMAL IO.	04123100
END COMMENT;	04123110
REAL TIM=STOP+1, U=TIM+1;	04123120
LABEL TEST,NOWAIT,PROC,NEW,QUP,INCR;	04123500
LABEL ERRORS,DISKERR,DS,X,SW,LP,TAPE,DK,DX,DX1,DC,OK,L1;	04124000
REAL T=NT1,S=NT2,S1=NT3,V=NT4,E=NT5,I=NT7;%	04125000
NAME LOCN=E; REAL IOD=NT6,FIN=S1;	04126000
SWITCH TYPE ← OK,LP,TAPE,OK,DK,OK,OK,OK,OK,OK;	04127000
	04128000
\$ SET OMIT = NOT(DFX)	04128010
SUBROUTINE FIXUPEUQ;	04128099
BEGIN IF EUQ[V] LSS 0 THEN EUQ[V]:=ABS(NOT 0) ELSE	04128100
BEGIN IF EUQ[V],[FF] LSS 1023 THEN LINKEU	04128200
P([EUQ[V]],IOR);	04128300
END;	04128400
END;	04128500
	04128600
\$ POP OMIT	04128610
\$ SET OMIT = NOT(NEWLOGGING)	04128611
STOPLOG(P1MIX,0);	04128799
\$ POP OMIT	04128800
P(CHANIO[C]);	04128801
S1=(T:=UNIT[P(CHANNEL[C],DUP))],[18;15];	% INITIALIZES TIM 04128900
IF U#30 THEN DCIOFINISH(R);	% INITIALIZES U 04129000
ERRORS;	04129100
IF (E ← R,[26;7])+(V ← T,[5;8]) # 0 THEN%	04129900
BEGIN IF(S1 ← FINALQUE[S]) < 0 THEN%	04130000
IF (E ← S1,[25;8] AND E) = 0 THEN%	04131000
IF V = 0 THEN	04132000
	04133000
\$ SET OMIT = NOT DFX	04133090
	04133100
IF (U AND 62) = 18 THEN	04133200
BEGIN I:=IOQUE[S];	

	GO TO DX;	04133300
	END ELSE	04133400
\$ POP OMIT		04133410
	GO TO OK;	04133500
	IF (U AND @774) # 16 THEN	04134000
	BEGIN	04134050
	RDCTABLE[U]:=(P(DUP))& (C-1)[1:46:2]& R[3:3:5];	04134060
	GO TO X;	04134070
	END;	04134080
	IF E = 0 THEN%	04135000
	BEGIN % RECOVERED MASS STORAGE %	04137000
	MAINTBUFFER[NXDISK:=NXDISK+4 AND 15]	04137100
	:= -0 & U[2:46:2] & LOCATQUE[S][4:3:5] &	04137110
	(LOGENTRY:=LOGENTRY+1)[CTF] &	04137120
	RDCTABLE[U][18:1:2];	04137130
	IF FINALQUE[S] GTR 0 THEN	04137140
	BEGIN	04137150
	MAINTBUFFER[NXDISK]:=(P(DUP)) &	04137160
	((M[M[S1:=LOCATQUE[S] INX NOT 2] INX 4]	04137170
	,[13:11] DIV ETRLNG)+1)[9:39:9];	04137180
	M[S1],[7:1] := 1;	04137190
	END;	04137200
	P(MAINTBUFFER[NXDISK+2]:=IOQUE[S]);	04137202
\$ SET OMIT = NOT(AUXMEM)		04137203
	IF NOT (U,[46:1]) THEN	04137204
	BEGIN	04137205
	STREAM(S:=P,[CF] : );	04137206
	BEGIN	04137207
	SI:=LOC S; DI:=LOC S; DS:=8DEC;	04137208
	END;	04137209
	END ELSE	04137210
\$ POP OMIT		04137211
	P(NFLAG(M[P]));	04137212
	P(P&V[1:44:4],[MAINTBUFFER[NXDISK+1]],STD);	04137215
	MAINTBUFFER[NXDISK+3]:=MAINTBUFFER[U];	04137220
	IF (LOGHOLDER INX 0) = 0 THEN	04137230
	BEGIN	04137240
	LOGHOLDER,[CF]:=[MAINTBUFFER[NXDISK]];	04137250
	FORK(P(,MAINTLOGGER),0,0,128,1);	04137260
	END ELSE M[LOGHOLDER,[FF]],[CF]:=	04137270
	[MAINTBUFFER[NXDISK]];	04137275
	LOGHOLDER,[FF]:=[MAINTBUFFER[NXDISK]];	04137280
	NUMAINTMESS:= NUMAINTMESS+1;	04137290
	T,[5:8] + 0;	04142000
	GO TO SW;	04142500
	END;%	04143000
	IF V = 0 THEN%	04144000
\$ SET OMIT = NOT(SHAREDISK)		04144099
	IF R,[25:4]=9 THEN % LOCKED ADDRESS	04144100
	BEGIN	04144110
	IF FINALQUE[S]>0 THEN % FIND IOD	04144120
	BEGIN I+IOQUE[S],[CF];	04144130
	IOD+LOCATQUE[S],[CF];	04144140
	WHILE (M[IOD],[CF])#I DO IOD+IOD+1;	04144150
	M[IOD],[20:1]+1; % SET LOCK BIT	04144160
	END;	04144170
	LQUE[LQAVAIL]:=M[IOQUE[S]]&S[1:41:7];	04144180
	STOP:=2;	04144200
	IF (LQAVAIL:=LQAVAIL+1)>LQMAX THEN	04144260
	PUNT(8); % LOCK QUE OVFLOW	04144270



```

        IOCOUNT[LOCATQUE[S],[3:5]]+*P(DUP)=1; % TO SWAP 04144275
        GO TO DISKERR; 04144280
    END ELSE 04144300
$ POP OMIT 04144301
        BEGIN % ORIGINAL ERROR ON MASS STORAGE% 04145000
        TINU[U],[18:12]:=P(DUP),[18:12]+1; 04146000
        MAINTBUFFER[U]:=R&TWO(C)[18:43:4]; 04146100
        RDCTABLE[U]:=(*P(DUP))&(C=1)[1:46:2]; 04146200
        V:=129; 04147000
$ SET OMIT = NOT(SHAREDISK) 04147399
        LOCATQUE[S],[8:1]:=M[IOQUE[S]],[42:1]; 04147400
        IF NOT R,[25:1] THEN M[IOQUE[S]],[42:1]:=0; 04147500
$ POP OMIT 04147501
        END% 04148000
    ELSE BEGIN % RECURRENT ERROR ON MASS STORAGE% 04149000
        P(MAINTBUFFER[U]:=P(DUP,LOD) OR 04150100
        R&TWO(C)[18:43:4]); 04150200
        IF (V * V+1) > 137 THEN% 04151000
        BEGIN R:=P; 04151200
            IF LOCATQUE[S],[9:1] THEN % OLAY I/O 04151220
                M[LOCATQUE[S]]:=R OR IOMASK; 04151230
$ SET OMIT = NOT(AUXMEM) 04151235
            IF NOT U,[46:1] THEN % AUXMEM 04151240
                BEGIN 04151250
                    V := 0; GO TO X; 04151260
                END; 04151270
$ POP OMIT 04151275
        DISKERR; 04151300
$ SET OMIT = NOT(DFX) 04151399
            I:=IOQUE[S]; 04151400
            IF R,[30:1] THEN 04151600
                BEGIN V:=M[I],[6:6]; FIXUPEUQ; 04151800
                    V:=0; GO TO X; 04152000
                END; 04152200
$ POP OMIT 04152201
            T,[5:10]:=0; 04152400
            GO TO DX; 04152600
        END; 04152800
        P(DEL); 04152900
    END;% 04153000
    UNIT[U] * T&V[5:40:8];% 04154000
    DS;% 04155000
        CHANNEL[P(TIO)] * U;% 04156000
        CHANIO[P(TIO)]:=TIM; %R59 04156100
        P([IOQUE[S]],[1:10]);% 04157000
        GO TO EXTERNAL;% 04158000
    XI 04159000
        STOP * (V#0)*2+1;% 04160000
        T,[5:13] * 32*E+8;% 04161000
        GO TO TEST; 04161500
    END; 04162000
    SWI: GO TO TYPE[T,[1:4]];% 04163000
    LPI 04164000
        IF STOP := (T := T&0[16:16:1]),[17:1] THEN 04165000
    TEST: IF FIRSTWAIT = NEXTWAIT THEN GO TO INCR ELSE% 04166000
        GO TO NEW ELSE GO TO NOWAIT;% 04167100
    TAPE: 04167200
        IF (R,[21:5] AND @23) = @22 THEN % BINARY READ 04167300
        IF R,[15:3] * ((8-R,[22:1]) AND 7) THEN % PARTIAL WORD TRANSFER 04167400
            IF MOD3IOS THEN % IF MODEL III CONTROLS 04167500
                BEGIN R,[28:1]:=1; % TREAT IT AS A PARITY,

```



```

INITIATEIO(IOQUE[S1],LOCATQUE[S1],[3:5],U)% 04180000
ELSE 04181000
PROC: T := T&0[16:16:2] 04182000
ELSE 04183000
BEGIN% 04187000
NEW: NEWIO;% 04188000
IF STOP THEN GO TO INCR;% 04189000
QUP: IF LOCATQUE[S],[FF] GTR @1777 THEN GO TO PROC; 04190000
QUEUEUP(U);% 04191000
T + T&4[13:43:5];% 04192000
END;% 04193000
INCR: 04194000
IF (TIM+CLOCK+P(RTR)=TIM) LSS 0 THEN TIM+0; %027-04194050
IOD:=IOQUE[S]; 04194100
IF (U OR 1)=19 THEN 04194200
BEGIN 04194300
IF (JUNK:=M[IOD],[5:7])>9 THEN 04194400
JUNK:=NEUP,[CF]+(JUNK AND @17); 04194500
IF JUNK<NEUP,[FF] THEN 04194550
PEUIO[JUNK]:=P(DUP,LOD)+CLOCK+P(RTR)-EUIO[C]; 04194600
END; 04194650
I:=(S1:=NFLAG(LOCATQUE[S]),[3:5]); % MIX INDEX 04194700
$ SET OMIT = NOT(NEWLOGGING) 04194799
IF FINALQUE[S] < 0 THEN % MCP I/O 04194800
MCPIOTIME[I]+(*P(DUP))+TIM; 04194900
$ POP OMIT 04194901
IOTIME[I]+(*P(DUP))+TIM; 04195000
IF S1.[10:1] THEN % CANDE I/O OR NO MEM 04195100
IF U=25 THEN FORGETAREA(0,IOD) ELSE 04195150
BEGIN 04195200
IF M[C]:=S1,[CF]-1],[18:7]=13 THEN FORGETAREA(0,C) 04195300
ELSE QUEVENT(C,I); 04195400
IF (V:=M[C+2]) < 0 THEN FORGETSPACE(V) ELSE 04195600
BEGIN 04195700
IF V.[2:1] THEN ELSE P([M[V]],PRL); 04195800
M[V+1]:=M[C+3]; 04195900
END 04196000
END CANDE IO HANDLER; 04196100
IF E#0 THEN 04196200
IF STOP THEN 04196400
P(T) 04196600
ELSE GO TO L1 04196800
ELSE BEGIN 04197000
IOCOUNT[I]+*P(DUP)=1; 04197500
RETURNIOSPACE(S); 04199000
L1: P(T&P(,S1,LOD)[FTF]); 04201000
END; 04202000
P([UNIT[U]],STD); 04203000
FIN + FINALQUE[S] AND NOT MEMORY;% 04205000
IF (U OR 1) NEQ 17 THEN 04205012
IF IOD.[24:1] THEN% 04206000
BEGIN V + ABS(IOD,[33:15]-R,[33:15]);% 04207000
IF IOD.[8:10] < V THEN% 04208000
IF IOD.[23:1] THEN% 04209000
V + IOD.[8:10];% 04210000
IF U < 16 THEN% 04211000
IF IOD.[21:2] = 0 THEN% 04212000
BEGIN; STREAM(A+0IB+M[S1,[33:15]+V-1]);% 04213000
BEGIN SI + LOC B;% 04214000
IF SC = "*" THEN TALLY + 1;% 04215000

```

```

                                A ← TALLY;%
                                END;%
                                V ← -P+V;%
                                END;%
                                FINISHOFFIO(U);%
                                END;%
                                IF E ≠ 0 THEN%
$ SET OMIT = NOT(SHAREDISK)
                                IF STOP≠2 THEN
$ POP OMIT
                                BEGIN IF STOP LEQ 1 THEN
                                    BEGIN
                                        FORK(
                                            P(,DISKORAUERROR)+((U AND @774) NEQ 16),
                                            R&S[3:43:5],-2,140,0);
                                        LOCATQUE[S],[11:1]:=1;
                                        END
                                    ELSE IF FIN < 0 THEN P(LOCATQUE[S],R,XCH,+);%
                                END%
$ SET OMIT = NOT(SHAREDISK)
                                ELSE
$ POP OMIT
                                ELSE BEGIN%
                                    IF FIN < 0 THEN P(R OR IOMASK,LOCATQUE[S],+)%
                                    ELSE
$ SET OMIT = NOT (DFX OR DKBNODFX)
                                    IF STOP GEQ 0 THEN
$ POP OMIT
                                        BEGIN
                                            LOCN ← [MLOCATQUE[S]];%
                                            IOD ← IOD,[33:15];%
                                            WHILE LOCN[0],[33:15] ≠ IOD DO%
                                                LOCN ← 1 INX LOCN;%
                                                LOCN[0] ← M OR FIN;%
                                            END
                                        END;%
                                    IF P1MIX = 0 THEN GO TO NOTHINGTOD0;%
                                    IF I = P1MIX THEN GO TO RETURN;%
                                    GO TO INITIATE;%
                                END IOCOMPLETE;%
                                SAVE REAL PROCEDURE WAITIO(IOD,MASK,U);%
                                VALUE MASK,U,IOD;%
                                REAL MASK,U,IOD;%
                                BEGIN%
                                    REAL T;
                                    DEFINE OCTADE= DS+3 RESET;3(IF SB THEN DS+SET ELSE
                                        DS+RESET;SKIP SB);%
                                    IOD ← NFLAG(P(,IOD,LOD))&TINU[U][3:3:5];%
                                    MASK ← NOT MASK;%
                                    IOREQUEST(NABS(IOD)&MASK[25:40:8],IOD,
                                        [IOD]&U[12:42:6]);%
                                    IOD ← IOD&0[25:25:8]&0[19:19:1];%
                                    WAITORSWAP(U,[IOD],[CF]);
                                    IF ((WAITIO+IOD.[26:7]) AND MASK AND MASK.[18:15])≠0 THEN
                                        BEGIN
                                            STREAM(IOD+IOD,[26:7],MASK+(NOT MASK),[41:7],
                                                Z:[TINU[U]], T:=T:=SPACE(12));
                                            BEGIN DS+20 LIT" UNEXP I=0 ERROR ON ";SI+Z;
                                                SI+SI+5;DS+3 CHR;DS+8 LIT";RESULT=";
                                                SI+LOC IOD;SI+SI+6;SKIP 3 SB;3(OCTADE);

```

```

04216000
04217000
04218000
04219000
04219100
04220000
04221000
04222000
04222499
04222500
04222501
04223000
04223500
04224000
04224010
04224100
04224500
04224750
04225000
04226000
04226499
04226500
04226501
04227000
04228000
04228100
04229099
04229100
04229101
04229200
04230000
04231000
04232000
04233000
04234000
04235000
04236000
04237000
04238000
04239000
04240000
04241000
04242000
04243000
04243100
04243200
04243300
04244000
04245000
04246000
04247000
04248000
04249000
04250000
04251000
04251200
04251300
04251400
04251500
04251600

```



DS:=8LIT" ERR,MX="; DS:=2DEC; T:=DI; DI:=DI-2; DS:=FILL;	04264600
DI:=T; DS:=LIT" ";	04264800
MEMPAR(DS:=7LIT"[M,PAR]");	04265000
LOKOUT(DS:=8LIT"[WRT,LK]");	04265200
PAR (DS:=5LIT"[PAR]");	04265400
NOTRDY(DS:=8LIT"[ENT,RDY]");	04265600
DESPAR(DS:=7LIT"[D,PAR]");	04265800
BSY (DS:=5LIT"[BSY]");	04266000
DS:=4LIT" DA="; T:=DI; DS:=8DEC; DI:=T; DS:=7FILL;	04266200
SI:=T; DI:=T; 8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR);	04266400
DS:=4LIT" SZ=";	04266600
SI:=LOC SZ; DS:=4DEC; T:=DI; DS:=3FILL; DI:=T;	04266800
DS:=3LIT" R=";	04267000
16(DS:=3RESET; 3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));	04267200
DS:=4LIT" IO="; SI:=SI-5;	04267400
IF SB THEN DS:=2LIT"4,"; SKIP SB;	04267600
IF SB THEN DS:=2LIT"3,"; SKIP SB;	04267800
IF SB THEN DS:=2LIT"2,"; SKIP SB;	04268000
IF SB THEN DS:=2LIT"1,";	04268200
DI:=DI-1; DS:=LIT"@";	04268400
END STREAM STATEMENT;	04268600
END SUBROUTINE AUXMESSAGE;	04268800
SUBROUTINE FIXUNIT; % FREE UNIT ARRAY ON AUXMEM ERROR	04269000
BEGIN	04269200
IF LOCATQUE[S],[FFF] LSS 1023 THEN % MORE ENTRIES IN THE QUEUE	04269400
BEGIN	04269600
UNIT[U]:=(*P(DUP))&LOCATQUE[S][FTF]&0[13:46:2]&0[5:40:8];	04269800
STARTIO(U);	04270000
END	04270200
ELSE UNIT[U]:=(*P(DUP))&(NOT 0)[18:18:30]&0[13:46:2]&0[5:40:8];	04270400
END SUBROUTINE FIXUNIT;	04270600
% POP OMIT	04270800
SUBROUTINE DISKMESSAGE;	04271000
BEGIN	04271200
STREAM(MSG, MK, A:=TINU[U], MIX, BI:=DSKADRS,	04271400
SI:=IOQUE[S],[27:6], R, KEY1:=KEY1:=SPACE(10));	04271600
BEGIN	04271800
SI:=LOC MK; SI:=SI+7; DS:=CHR;	04272000
SI:=SI+5; DS:=3CHR; DS:=LIT" ";	04272200
CI:=CI+MSG;	04272400
GO L0; GO L1; GO L2; GO L3; GO L4; GO L5; GO L6; GO L7;	04272600
L0: DS:= 9LIT"NOT READY"; GO TO MX;	04272800
L1: DS:= 4LIT"BUSY"; GO TO MX;	04273000
L2: DS:= 8LIT"I/O MEM ";	04273200
L3: DS:= 6LIT"PARITY"; GO TO MX;	04273400
L4: DS:=12LIT"I/O INV ADDR"; GO TO MX;	04273600
L5: DS:= 3LIT"EU "; GO TO L0;	04273800
L6: DS:=13LIT"INV DISK ADDR"; GO TO MX;	04274000
L7: DS:=10LIT"WRITE LOCK";	04274200
MX: DS:= 6LIT", MIX="; DS:=2DEC;	04274400
MSG:=DI; DI:=DI-2; DS:=FILL; DI:=MSG;	04274600
DS:=5LIT", DA="; DS:=8CHR;	04274800
DS:=7LIT", SEGS="; DS:=2DEC;	04275000
DS:=4LIT", R=";	04275200
16(DS:=3RESET; 3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));	04275400
SI:=SI-5; DS:=5LIT", IO=";	04275600
IF SB THEN DS:=2LIT"4,"; SKIP SB;	04275800
IF SB THEN DS:=2LIT"3,"; SKIP SB;	04276000
	04276200
	04276400

IF SB THEN DS:=2LIT"2,"; SKIP SB;	04276600
IF SB THEN DS:=2LIT"1,";	04276800
DI:=DI-1; DS:=LIT"@";	04277000
END STREAM STATEMENT;	04277200
END SUBROUTINE DISKMESSAGE;	04277400
SUBROUTINE DETAILRECORDENTRY;	04277600
BEGIN	04277800
KEY2 := GETSPACE(6,9,0) + 2;	04278000
M[KEY2] := 0 & RDCTABLE[U][18:1:2];	04278200
IF MIX NEQ 0 THEN	04278400
BEGIN	04278600
M[KEY2] := (*P(DUP)) & MIX[20:43:5] &	04278800
(IF FINALQUE[S] LSS 0 THEN 0 ELSE	04279000
(M[M[LOCATQUE[S] INX NOT 2] INX 4],[13:11] DIV ETRLNG)+1)[9:39:9];	04279200
END;	04279400
M[KEY2+1] := TRANSACTION[U];	04279600
IF NOT DISC THEN	04279800
BEGIN	04280000
STREAM(S:=IOD,[FF], D:=KEY2+2);	04280200
BEGIN	04280400
S:=LOC S; DS:=8DEC;	04280600
END;	04280800
END	04281000
ELSE M[KEY2+2] := DSKADRS;	04281200
M[KEY2+3] := IOQUE[S];	04281400
M[KEY2+4] := R & RDCTABLE[U][3:3:5];	04281600
M[KEY2+5] := IF FINALQUE[S] LSS 0 THEN 0 ELSE LOCATQUE[S] INX NOT 2;	04281800
END DETAILRECORDENTRY;	04282000
SUBROUTINE FINISHDETAIL;	04282200
BEGIN	04282400
IF MIX NEQ 0 THEN CHECKJOBORFILEMESS(MIX,M[KEY2+5],U);	04282600
LINKUP(4+DISC,KEY2);	04282800
END;	04283000
	04283200
	04283400
	04283600
	04284000
DISC:=(U:=LOCATQUE[S]=R,[3:5]),[12:6]),[46:1];	04284200
MIX:=LOCATQUE[S],[3:5];	04284400
IF (OLAYIO := ((FINALQUE[S] LSS 0) AND (LOCATQUE[S],[9:1]))) THEN	04284600
BEGIN	04284800
STREAM(S:=0&FINALQUE[S][CTC]&FINALQUE[S][21:8:12], D:=[DSKADRS]);	04285000
BEGIN	04285200
S:=LOC S; DS:=8DEC; % DISK ADDRESS IN FINALQUE FOR OLAY I/O	04285400
END;	04285600
END ELSE DSKADRS := M[IOQUE[S]]; % DISK ADDRESS IN FINALQUE FOR OLAY I/O	04285800
MK:=*"; MSG:=(-1);	04286000
R:=R&IOQUE[S][3:3:5]; % RESTORE HARDWARE UNIT TYPE	04286200
IOD := IOQUE[S];	04286400
IF DISC THEN	04286600
BEGIN	04286800
IF R,[30:1] THEN % DISK NOT READY	04287000
BEGIN	04287200
% SET OMIT = NOT(DFX)	04287400
U:=IOD,[4:1] OR 18;	04287600
T:=S;	04287800
IF NOT (U:=IF U THEN 18 ELSE 19) THEN	04288000
DO IOQUE[T],[3:5] := 6 UNTIL	04288200
(T := LOCATQUE[T],[FF]) = @77777;	04288400
IF OLAYIO THEN % IOFINISH HAS ALREADY PLACED RESULT DESCRIPTOR	04288600

BEGIN	04288800
UNIT[U],[FF] := LOCATQUE[S],[FF]; % NEXT ELEMENT IN QUEUE	04289000
RETURNIOSPACE(S);	04289200
S := UNIT[U],[FF];	04289600
IF MIX NEQ 0 THEN IOCOUNT[MIX] := *P(DUP) -1;	04289800
END;	04290000
IF S NEQ @77777 THEN % MORE ENTRIES IN THE QUEUE	04290200
BEGIN	04290400
E:=UNIT[U];	04290600
IF (T2:=EUQ[T1=M[IOQUE[S]],[6:6]]) LSS 0 THEN	04290800
BEGIN % EU IN PROCESS	04291000
LOCATQUE[UNIT[U1],[CF]],[FF] := S;	04291200
UNIT[U1],[CF] := E;	04291400
END	04291600
ELSE % EU NOT IN PROCESS	04291800
IF T2,[FF] GTR 1023 THEN % EUQ EMPTY	04292000
BEGIN	04292200
IF (EUQ[T1]=T2&E[18:18:30]),[2:1] THEN	04292400
BEGIN	04292600
IF EUW,[FF]=@77777 THEN EUW,[FF]:=T ELSE	04292800
EUQ[EUW,[CF]],[3:15]:=T;	04293000
EUW,[CF]:=T;	04293200
STARTIO(U1);	04293400
END;	04293600
END	04293800
ELSE	04294000
BEGIN % EUQ NOT EMPTY	04294200
LOCATQUE[T2,[CF]],[FF] := S;	04294400
EUQ[T],[CF] := E;	04294600
END;	04294800
END; % IF MORE ENTRIES IN THE QUEUE	04295000
\$ POP OMIT	04295200
UNIT[U]:=(*P(DUP))&@77777[5:20:28];	04295400
MSG:=0; MK:="*"; % NOT READY	04295600
DISKMESSAGE;	04295800
DETAILRECORDENTRY;	04296000
READY := NOT TWO(U) AND READY;	04296200
RRRMECH := NOT TWO(U) AND RRRMECH;	04296400
UNIT[U],[5:10] := 2;	04296600
GO TO KILLL;	04298800
END; % IF NOT READY	04299000
LOCATQUE[S],[FF] := NOT 0;	04299200
T1 := MIX;	04299400
IF R,[26:7] NEQ 1 AND NOT OLAYIO THEN % NOT BUSY OR SPECIAL I/O	04299600
BEGIN	04299800
PARITY := (IOD,[24:1] AND (R,[26:7]=16)); % PARITY CONDITION	04300000
IF FINALQUE[S] GTR 0 THEN % OBJECT JOB ERROR	04300200
BEGIN	04300400
IF PARITY THEN GO TO START; % RECOVERABLE ERROR	04300500
PRTROW[T1],[FF] := 20; % I/O ERROR	04300600
PRTROW[T1],[PSF]:= 1; % TERMINATE	04300800
END % OBJECT ERROR	04301000
ELSE	04301200
BEGIN % MCP I/O	04301400
IF (T2:=IOQUE[S],[CF]=FENCE) GTR 0 THEN % ABOVE THE FENCE	04301600
IF (T1:=POSSESS[T2 DIV CHUNKZIZE]) NEQ 0 THEN % NON-ZERO MIX	04301800
BEGIN	04302000
IF JAR[MIX,9],[1:1] THEN % "SYSTEM" JOB	04302200
IF PARITY THEN GO TO START;	04302600
% DONT DS LIBMAIN/DISK ON PARITY ERROR	04302800



```

                GO TO DSIT;
                END; % NON-ZERO MIX
            END; % MCP I/O
        END; % NOT BUSY OR SPECIAL I/O

START:
    TRANSACTION[U] := TRANSACTION[U]-1;
    MASK := IF (FIN := FINALQUE[S]) LSS 0 THEN FIN.[25:8] ELSE @377;
    IF (E := R.[25:8] AND MASK) = 0 THEN % ERRORS ARE ACCETABLE
        BEGIN % FIX UP IOQUE

QUIT:
    IF MSG NEQ (-1) AND DISC THEN DISKMESSAGE;
    DETAILRECORDENTRY;
    $ SET OMIT = NOT(AUXMEM);
    IF NOT DISC THEN % AUXMEM
        BEGIN
            FIXUNIT;
            IF R.[30:1] THEN % NOT READY CONDITION
                BEGIN
                    IF (NOTRDYCOUNT=0) OR (E NEQ 0) THEN AUXMESSAGE;
                    NOTRDYCOUNT := NOTRDYCOUNT + 1;
                END
            ELSE
                BEGIN
                    AUXMESSAGE;
                    NOTRDYCOUNT := 0;
                END;
            IF E=0 THEN R := RSLT;           % ERROR WAS RECOVERED
        END;
    $ POP OMIT
    RETURNIOSPACE(S);

    FIN:=FINALQUE[S] AND NOT MEMORY;
    IF (T1:=FIN) LSS 0 THEN % MCP I/O
        BEGIN
            IF NOT OLAYIO THEN % I/O FINISH PLACES RESULT DESC. FOR OLAY
                M[LOCATQUE[S]]:=R&E[25:40:8]&IOD[3:3:5] OR IOMASK;
            END % IF MCP I/O
        ELSE
            BEGIN
                IF E NEQ 0 THEN % ERRORS
                    BEGIN
                        P(,T1,PRL);
                        T1 := T1&E[25:40:8];
                    END
                ELSE P(,T1,IOR);
                LOCN := [M[LOCATQUE[S]]];
                IOD := IOD.[33:15];
                WHILE LOCN[0].[33:15] NEQ IOD DO LOCN := 1 INX LOCN;
                LOCN[0] := P(,T1,LOD);
            END;
            IOCOUNT[MIX] := (*P(DUP)) -1;
            GO TO KILL;
        END;
    IF E THEN % BUSY
        BEGIN
            MSGI:=1; % BUSY

RETRY:
    $ SET OMIT = NOT(AUXMEM)

```

```

04303000
04303200
04303400
04303600
04303800
04304000
04304200
04304400
04304600
04304800
04305000
04305200
04305400
04305600
04305800
04306000
04306200
04306400
04306600
04306800
04307000
04307200
04307400
04307600
04307800
04308000
04308200
04308400
04308600
04308800
04309000
04309200
04309400
04309600
04309800
04310000
04310200
04310400
04310600
04310800
04311000
04311200
04311400
04311600
04311800
04312000
04312200
04312400
04312600
04312800
04313000
04313200
04313400
04313600
04313800
04314000
04314200
04314400
04314600
04314790

```

IF NOT DISC THEN AUXMESSAGE ELSE	04314800
\$ POP OMIT	04314810
DISKMESSAGE;	04314820
DETAILRECORDENTRY;	04315000
\$ SET OMIT = NOT(AUXMEM)	04315190
IF NOT DISC THEN FIXUNIT; % ALLOW IO TO AUXMEM	04315200
\$ POP OMIT	04315210
T1:=(IF DISC THEN IOQUE[S]&6[3:43:5] ELSE IOQUE[S]);	04315400
RETURNIOSPACE(S);	04315600
	04315800
P1MIX:=MIX;	04316000
IOCOUNT[MIX] := (*P(DUP)) -1;	04316200
IF NOT OLAYIO THEN % RETRIES ARE OK	04316400
IOREQUEST(FINALQUE[S], T1,	04316600
(IF DISC THEN LOCATQUE[S]&@22[12:42:6] ELSE	04316800
LOCATQUE[S]));	04317000
	04317200
P1MIX:=0;	04317400
GO TO KILLER;	04317600
END; % IF BUSY	04317800
IF E,[46:1] THEN % I/O MEMORY PARITY	04318000
BEGIN	04318200
MSG:=2;	04318400
E:=@1537;	04318600
GO TO QUIT;	04318800
END;	04319000
IF E,[41:1] THEN % INVALID ADDRESS	04319200
BEGIN	04319400
MSG:=4;	04319600
E:=@1537;	04319800
GO TO QUIT;	04320000
END;	04320200
\$ SET OMIT = NOT(SHAREDISK)	04320400
IF R,[25:1] THEN % READ/WRITE LOCK	04320600
BEGIN	04320800
AREA:=GETSPACE(10,0,0)+2;	04321000
IF R,[29:1] THEN	04321200
BEGIN	04321400
STREAM(AREA);	04321600
DS:=10 LIT " FPM FULL*";	04321800
END	04322000
ELSE	04322200
BEGIN	04322400
STREAM(AREA);	04322600
DS:=15 LIT " FPM NOT READY*";	04322800
END;	04323000
SPOUTER(AREA,PSEUDOMIX[MIX],35);	04323200
MSG := (=1); % DONT SPOUT DISK MESSAGE	04323400
E:=@16;	04323600
GO TO QUIT;	04323800
END;	04324000
IF LOCATQUE[S],[8:1] THEN % MUST UNLOCK ADDRESS	04324200
BEGIN	04324400
STREAM(A:=DSKADRS,D:=[JUNK]);	04324600
BEGIN S1:=LOC A;DS:=8 OCT END;	04324800
UNLOCK(JUNK);	04325000
END;	04325200
\$ POP OMIT	04325400
IF NOT E,[43:1] THEN % NOT PARITY,CHECK DISK ADDRESS	04325600
BEGIN	04325800
STREAM(DA:=MASK:=DSKADRS : EUI:=MASK,[6:6], A:=0,	

EUA=[MULTITABLE[16+2×MASK,[5:1]]];	04326000
BEGIN	04326200
SII=LOC DA;	04326400
IF SC GTR "1" THEN GO TO BAD;	04326600
IF SC LSS "0" THEN GO TO BAD;	04326800
\$ SET OMIT = SHAREDISK	04327000
7(	04327200
\$ POP OMIT	04327400
\$ SET OMIT = NOT(SHAREDISK)	04327600
4(	04327800
\$ POP OMIT	04328000
IF SC LSS "0" THEN JUMP OUT TO BAD; SII=SI+1;	04328200
IF SC GTR "9" THEN JUMP OUT TO BAD);	04328400
\$ SET OMIT = SHAREDISK	04328600
SII=SI-5;	04328800
\$ POP OMIT	04329000
\$ SET OMIT = NOT(SHAREDISK)	04329200
SII=SI-2;	04329400
\$ POP OMIT	04329600
DII=LOC DA; DS:=2 OCT;	04329800
SII=EUA; SI:=SI+14; SKIP EU SB;	04330000
DII=LOC A; DII=DI+7; SKIP 2 DB;	04330200
IF SB THEN SKIP DB;	04330400
SII=LOC DA; SI:=SI+6;	04330600
IF SC NEQ "0" THEN GO TO BAD; SII=SI+1;	04330800
4(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB);	04331000
SII=LOC A; SII=SI+7; IF SC GTR "4" THEN GO BAD;	04331200
IF SC LSS "0" THEN GO BAD;	04331400
SII=EUA; SII=SI+EU; SKIP SB; SKIP A SB;	04331600
IF SB THEN GO TO OK;	04331800
BAD:	04332000
OK:	04332200
TALLY:=1;	04332400
DA:=TALLY;	04332600
END;	04332800
IF (MASK:=P) OR E.[42:1] THEN % BAD ADDRESS OR EU NOT READY	04333000
BEGIN	04333200
MSG:=5+MASK; % 5=EU NOT READY, 6=INVALID DISK ADDRESS	04333400
IF NOT MASK THEN MKI="#";	04333600
IF (MIX NEQ 0) OR OLAYIO THEN	04333800
BEGIN	04334000
EI=@1537; GO TO QUIT;	04334200
END;	04334400
DISKMESSAGE;	04334600
DETAILRECORDENTRY;	04334800
GO TO KILLER; % LET IT HANG	04335000
END	04335200
ELSE	04335400
BEGIN % MUST BE E.[44:1], MEM,PAR.	04335600
MSG:=2; EI=@1537; GO TO QUIT;	04335800
END;	04336000
END; % IF NOT PARITY	04336200
IF IOQUE[S].[24:1] THEN % DISK PARITY ON READ	04336400
BEGIN	04336600
MSG:=3; % PARITY	04336800
EI=@20;	04337000
GO TO QUIT;	04337200
END;	04337400
MSG:=7; % WRITE LOCK	04337600
EI=@1537;	04337800
GO TO QUIT;	
END; % IF DISK	



```

        STREAM(A:=[PUNTER[6]]); DS:=15LIT"AUX MEM ERROR,*";
        PUNT([PUNTER[6]]);
        END;
        AUXERRORTOG := (*P(DUP)) OR TWO(MIX);
        PRTRW[MIX],[FF] := 20; % I/O ERROR
        PRTRW[MIX],[PSF]:= 1; % TERMINATE
        GO TO QUIT;
$ POP OMIT % AUXMEM
KILLL:
        LOCATQUE[S],[11:1]:=0;
KILLER:
        IF KEY1 NEQ 0 THEN SPOUTER(KEY1,PSEUDOMIX[MIX],35);
        IF KEY2 NEQ 0 THEN FINISHDETAIL;
        KILL([R] INX NOT 1);
        END PROCEDURE DISKORAUERROR;
PROCEDURE ACTUALIOERR(R); VALUE R; REAL R;
BEGIN
REAL
        E,
        T,
        S,
        F,
        U,
        T1,
        T2,
        T3,
        KEY,
        FIN      = NT3,
        IOD      = NT6,
        MASK,
        DISC,
        TYPE;

NAME      LOCN = T3;

LABEL L1, L2, D17, D19, D22, START, NOTREADYMESS, NTRDY,
        EOF, REALEOF, TAPERTRY, SIX, SEVEN, FIX, LEAVE,
        REWINDING, NOCODE, CLEAR, KILLL, KILLER;
LABEL READER, PRINTER, TAPE, DRUM, DISK, SPO, PUNCH,
        PAPERPUNCH, PAPER, DATACOM;

SWITCH W := READER,PRINTER,TAPE,DRUM,DISK,SPO,PUNCH,NOCODE,
        PAPERPUNCH,PAPER,DATACOM;

SUBROUTINE MAKEMESS;
BEGIN
        STREAM(S1:=F,[43:5], S2:=F,[38:5], A:=TINU[U],
        MX:=LOCATQUE[S],[3:5], KEY:=KEY:=SPACE(10));
        BEGIN
                S1:=LOC A; S1:=S1+5;
                DS:=LIT"*"; DS:=3 CHR; DS:=LIT" ";
                C1:=C1+S1; GO TO LL;
                GO L1; GO L2; GO L3; GO L4; GO L5; GO L6; GO LL; GO LL;
                DS:=19 LIT"BLANK TAPE ON WRITE"; GO TO MXX;
L1:   DS:= 4 LIT"BUSY"; GO TO MXX;
L2:   DS:= 8 LIT"I/O MEM ";
L3:   DS:= 6 LIT"PARITY"; GO TO MXX;
L4:   DS:=12 LIT"I/O INV ADDR"; GO TO MXX;
L5:   DS:= 9 LIT"I/O ERROR"; GO TO MXX;
L6:   DS:=10 LIT"WRITE LOCK"; GO TO MXX;

```

```

04350000
04350200
04350400
04350600
04350800
04351000
04351200
04351400
04351600
04351800
04352000
04352200
04352400
04352800
04353000
04353200
04353400
04353600
04353800
04354000
04354200
04354400
04354600
04354800
04355000
04355200
04355400
04355600
04355800
04356000
04356200
04356400
04356600
04356800
04357000
04357200
04357400
04357600
04357800
04358000
04358200
04358400
04358600
04358800
04359000
04359200
04359400
04359600
04359800
04360000
04360200
04360400
04360600
04360800
04361000
04361200
04361400
04361600
04361800
04362000

```

LL:	GO TO PS;	04362200		
MX:	GO TO MIXIT;	04362400		
PS:	DI:=DI-5; DS:=LIT"#"; DI:=DI+4;	04362600		
	CI:=CI+S2; GO TO LLO; GO TO LL1; GO TO LL2;	04362800		
NR:	DS:= 9 LIT"NOT READY";	GO TO MIXIT;	04363000	
LLO:	DS:= 5 LIT"PRINT";	GO TO CHK;	04363200	
LL1:	DS:= 4 LIT"READ";	GO TO CHK;	04363400	
LL2:	DS:= 5 LIT"PUNCH";		04363600	
CHK:	DS:= 5 LIT"CHECK";		04363800	
MIXIT:	DS:= 6 LIT", MIX="; DS:=2 DEC; DS:=LIT"+";		04364000	
	DI:=DI-3; DS:=FILL;		04364200	
	END;		04364400	
	END OF MAKEMESS;		04364600	
			04364800	
SUBROUTINE	DETAILRECORDENTRY;		04365000	
	BEGIN		04365200	
	KEY := GETSPACE(ABS(T2),9,0)+2;		04365400	
	M[KEY] := (ABS(T2) DIV 5 -1) & RDCTABLE[U][18:1:2];		04365600	
	IF (NT1:=LOCATQUE[S],[3:5]) NEQ 0 THEN		04365800	
	BEGIN		04366000	
	M[KEY] := (*P(DUP)) & NT1[20:43:5] &		04366200	
	(IF FINALQUE[S] LSS 0 THEN 0 ELSE		04366400	
	(M[M[LOCATQUE[S] INX NOT 2] INX 4],[13:11] DIV ETRLNG)+1)[9:39:9];		04366600	
	CHECKJOBORFILEMESS(NT1,		04366800	
	(IF FINALQUE[S] LSS 0 THEN 0 ELSE LOCATQUE[S] INX NOT 2),		04367000	
	U);		04367200	
	END;		04367400	
	M[KEY+1] := TRANSACTION[U];		04367600	
	M[KEY+2] := IF TYPE=2 THEN RDCTABLE[U] & U[3:43:5] ELSE 0;		04367800	
	M[KEY+3] := IOQUE[S];		04368000	
	M[KEY+4] := R & RDCTABLE[U][3:3:5];		04368200	
	IF TYPE=2 THEN		04368400	
	BEGIN		04368600	
	M[KEY+5] := MULTITABLE[U];		04368800	
	M[KEY+6] := LABELTABLE[U];		04369000	
	M[KEY+7] := PRNTABLE[U];		04369200	
	M[KEY+8] := 0;		04369400	
	M[KEY+9] := 16;		04369600	
	END;		04369800	
	IF T2 GTR 0 THEN LINKUP(TYPE+1,KEY);		04370000	
	END DETAILRECORDENTRY;		04370200	
			04370400	
	DEFINE MAKEMLOG(MAKEMLOG1) =		04370600	
	BEGIN		04370800	
	T2:=MAKEMLOG1; DETAILRECORDENTRY;		04371000	
	END#;		04371200	
			04371400	
			04371800	
	U:=LOCATQUE[S]:=R,[3:5],[12:6];		04372000	
START:			04372200	
	T:=UNIT[U]&0[13:13:2];		04372400	
	TRANSACTION[U] := TRANSACTION[U]-1;		04372600	
	TYPE := T,[1:4];		04372800	
	MASK:=IF (T2:=FINALQUE[S]) LSS 0 THEN T2,[25:8] ELSE @377;		04373000	
	IF (E:=T,[5:8] AND MASK) = 0 THEN		% ACCEPTBLE	04373200
	BEGIN			04373400
	F:=1; % RETAIN ERROR FIELD			04373600
	GO TO FIX;			04373800
	END;			04374000
	IF E THEN		% BUSY	04374200

BEGIN	04374400
T3:=1 & (U=30)[43:47:1]; % BUSY/INCOMPLETE MASK	04374600
IF U LSS 16 AND TRANSACTION[U] LEQ 0 THEN	04374800
BEGIN	04375000
P(0); % DONT SPOUT MESSAGE	04375200
GO TO REWINDING;	04375400
END;	04375600
IF U NEQ 25 THEN % NOT SPO	04375800
BEGIN	04376000
F:=1; % BUSY	04376200
MAKEMESS;	04376400
SPOUTER(KEY,PSEUDOMIX[LOCATQUE[S],[3:5]],35);	04376600
END;	04376800
MAKEMLOG(IF TYPE=2 THEN 10 ELSE 5);	04377000
L1: DO BEGIN	04377200
SLEEP([CLOCK],NOT CLOCK);	04377400
UNIT[U]:=(*P(DUP))&P(T,XCH)[CTC];	04377600
STARTIO(U);	04377800
SLEEP([UNIT[U]],@100000000000);	04378000
TRANSACTION[U] := TRANSACTION[U]-1;	04378200
END UNTIL (UNIT[U],[5:8] AND T3) = 0;	04378400
TRANSACTION[U] := TRANSACTION[U]+1;	04378600
IF (UNIT[U],[5:8] AND MASK) = 0 THEN GO TO CLEAR;	04378800
GO TO START;	04379000
END;	04379200
IF E,[45:1] THEN % NOT READY	04379400
BEGIN	04379600
IF E,[43:1] THEN	04379800
BEGIN	04380000
IF TYPE=0 THEN GO TO READER; % READ CHECK	04380200
IF TYPE=1 THEN GO TO PRINTER; % PRINT CHECK	04380400
IF TYPE=6 THEN GO TO PUNCH; % PUNCH CHECK	04380600
END;	04380800
IF U NEQ 25 THEN % NOT SPO,	04381000
BEGIN	04381200
NOTREADYMESS;	04381400
F:=96; % NOT READY	04381600
MAKEMLOG(IF TYPE=2 THEN 10 ELSE 5);	04381800
MAKEMESS;	04382000
P(1); % SPOUT MESSAGE	04382200
REWINDING;	04382400
READY := NOT TWO(U) AND READY;	04382600
NTRDY;	04382800
RRRMECH:=NOT TWO(U) AND RRRMECH;	04383000
IF P THEN SPOUTER(KEY,PSEUDOMIX[LOCATQUE[S],[3:5]],35);	04383200
END;	04383400
UNIT[U],[5:10] := 2;	04383600
IF (T1:=LOCATQUE[T2:=S],[3:5]) NEQ 0 THEN	04383800
BEGIN	04384000
DO BEGIN	04384200
T2:=LOCATQUE[T2],[FF];	04384400
IOCOUNT[T1]:=*P(DUP)-1;	04384600
END UNTIL T2 GTR 1023;	04384800
END;	04385000
GO TO KILLL;	04385200
END;	04385400
D17: IF E,[46:1] THEN % I/O MEMORY PARITY	04385600
BEGIN	04385800
	04386000
	04386200

	F1=2; % I/O MEM PARITY	04386400
L2:	MAKEMESS;	04386600
	SPOUTER(KEY,PSEUDOMIX[LOCATQUE[S],[3:5]],35);	04386800
	MAKEMLOG(IF TYPE=2 THEN 10 ELSE 5);	04387000
	P(@1537); % ACCEPT EOF/EOT/EOP	04387200
	GO TO SIX;	04387400
	END;	04387600
	IF E,[41:1] AND TYPE NEQ 2 THEN % I/O INVALID ADDRESS	04387800
	BEGIN % [41:1] FOR TAPE = BACKWARD DRIVE	04388000
D22:	F1=4; % I/O INVALID ADDRESS	04388200
	GO TO L2;	04388400
	END;	04388600
		04388800
	GO TO W[TYPE];	04389000
		04389200
D19:	E := 1023; GO TO D17;	04389400
		04389600
SPO:		04389800
	IF E,[43:1] THEN GO TO L1; % ERROR BUTTON	04390000
	GO TO D19;	04390200
		04390400
PRINTER:		04390600
	IF E,[42:1] THEN % END OF PAGE	04390800
	BEGIN	04391000
	IF IOQUE[S],[27:6]=0 THEN GO FIX; % NOT SPACING	04391200
	COMMENT IGNORE EOP IF NO SPACE OR SKIP;	04391400
	IOQUE[S],[18:15] := @40001; % INHIBIT DATA XFER, SKIP TO CHANNEL	04391600
	GO TO CLEAR;	04391800
	END;	04392000
	IF E,[43:1] THEN	04392200
	BEGIN	04392400
	F:=0; % PRINT CHECK	04392600
	MAKEMESS;	04392800
	SPOUTER(KEY,PSEUDOMIX[LOCATQUE[S],[3:5]],35);	04393000
	IF E,[45:1] THEN GO TO NOTREADYMESS; % PRINTER NOT READY	04393200
	MAKEMLOG(IF TYPE=2 THEN 10 ELSE 5);	04393400
	P(0); % CLEAR ERROR FIELD	04393600
	TINU[U],[18:12] := P(DUP),[18:12]+1;	04393800
	GO TO SIX;	04394000
	END;	04394200
	GO TO D19; % PARITY	04394400
		04394600
READER:		04394800
	IF E,[43:1] THEN % READ CHECK	04395000
	BEGIN	04395200
	TINU[U],[18:12] := P(DUP),[18:12]+1;	04395400
	F:=32; % READ CHECK	04395600
	MAKEMLOG(5);	04395800
	MAKEMESS;	04396000
	P(1); % SPOUT MESSAGE	04396200
	GO TO NTRDY;	04396400
	END;	04396600
	IF E,[42:1] THEN % EOF CARD READER=TREAT AS NOT READY	04396800
	BEGIN	04397000
	UNIT[U],[5:8] := 4; % ERROR FIELD=NOT READY	04397200
	R,[25:8] := 4; % RESLT,DESC.=NOT READY	04397400
	TRANSACTION[U] := TRANSACTION[U]+1;	04397600
	GO TO START;	04397800
	END;	04398000
	COMMENT MUST BE D19 = USUALLY INVALID CHARACTOR;	04398200



```

STREAM(A:=0 : B:=IOQUE[S]);
BEGIN
  DI := A; SI := B; DI := DI+8;
  IF SC = @14 THEN A := DI;
  2(40(DI:=DI+8; SI:=SI+1);
  IF SC = @14 THEN JUMP OUT 2 TO L);
  DI := DI-8; SI := SI-1;);
  DI := A;
L:  A := DI;
  END;
  IF (T1 := P) = 0 THEN GO TO D19; % NOT INVALID CHARACTER
  IF T1 NEQ 1 THEN % NOT IN COLUMN 1
  BEGIN
    STREAM(A:=TINU[U],T1,KEY:=KEY:=GETSPACE(10,0,0)+2);
    BEGIN
      DS := LIT "#"; SI := LOC A; SI := SI+5;
      DS := 3 CHR;
      DS := 16 LIT " INV CHR IN COL ";
      DS := 2 DEC; DS := LIT "+";
    END;
    P(1); % SPOUT MESSAGE
    GO TO NTRDY;
  END;
  T,[CF] := IOQUE[S];
  GO TO REALEOF;

PUNCH:
  IF E,[43:1] THEN
  BEGIN
    F:=64; % PUNCH CHECK
    MAKEMESS;
    SPOUTER(KEY,PSEUDOMIX[LOCATQUE[S],[3:5]],35);
    % NEW PUNCH DOES NOT GO NOT=READY ON PUNCH CHECK
    IF E,[45:1] THEN GO TO NOTREADYMESS; % NOT READY
    MAKEMLOG(5);
    TINU[U],[18:12]:=P(DUP),[18:12]+1;
    F:=0; % ZERO ERROR FIELD
    GO TO CLEAR;
  END;
  GO TO D19; % PARITY

PAPERPUNCH:
  IF R,[27:1] THEN % EOR
  BEGIN
    P(@40);
    GO TO SIX;
  END;
  GO TO D19; % PARITY

PAPER:
  IF R,[27:2] NEQ 0 THEN GO TO EOF; % BOT/EOT
  IF E,[44:1] THEN % PARITY
  BEGIN
    P(@20);
    GO TO SIX;
  END;
  GO TO NOCODE;

DATACOM:
  IF(T3:=1&E[43:43:1])=@21 THEN GO TO L1;

```

```

04398400
04398600
04398800
04399000
04399200
04399400
04399600
04399800
04400000
04400200
04400400
04400600
04400800
04401000
04401200
04401400
04401600
04401800
04402000
04402200
04402400
04402600
04402800
04403200
04403400
04403600
04403800
04404000
04404200
04404400
04404600
04404800
04405000
04405200
04405400
04405600
04405800
04406000
04406200
04406400
04406600
04406800
04407000
04407200
04407400
04407600
04407800
04408000
04408200
04408400
04408600
04408800
04409000
04409200
04409400
04409600
04409800
04410000
04410200
04410400

```

PAPER TAPE  
 DATACOM

NOCODE:	04410600
F := 5; % I/O ERROR	04410800
GO TO L2;	04411000
DRUM: % DRUM NOW HANDLED IN DISKORAUERROR	04411200
DISK: % DISK NOW HANDLED IN DISKORAUERROR	04411400
DO UNTIL FALSE;	04411600
04411800	
04412000	
04412200	
TAPE:	04412400
TRANSACTION[U] := TRANSACTION[U]+1;	04412600
IF E,[44:1] THEN	04412800
IF R,[2:1] THEN % MOD III DESCRIPTOR	04413000
BEGIN % COULD BE MEM,PAR, BLANK TAPE,BOT,EOT	04413200
IF R,[11:1] THEN GO TO D19; % MEMORY PARITY	04413400
OPTION:=OPTION OR M; % MEANS MOD3IOS:=TRUE	04413600
IF R,[24:1] THEN % READING	04413800
BEGIN	04414000
IF R,[13:1] THEN R,[27:1]:=1; % BOT, SET EOF	04414200
IF R,[14:1] THEN % EOT	04414400
IF (E AND @367)=0 THEN % PARITY	04414600
IF R,[27:1]=0 THEN % NOT EOF	04414800
GO TO FIX; % FINISH I/O	04415000
END	04415200
ELSE	04415400
BEGIN % WRITING	04415600
IF R,[12:1] THEN % BLANK TAPE ON WRITE	04415800
BEGIN	04416000
F:=9; % BLANK TAPE ON WRITE	04416200
MAKEMESS;	04416400
SPOUTER(KEY,PSEUDOMIX[LOCATQUE[S],[3:5]],35);	04416600
MAKEMLOG(10);	04416800
P(16);	04417000
GO TO SIX;	04417200
END;	04417400
IF R,[14:1] THEN R,[27:1]:=1 ELSE GO FIX; % EOT,SET EOF BIT	04417600
END;	04417800
END % MOD III DESCRIPTOR	04418000
ELSE GO TO D19; % PARITY	04418200
IF R,[24:1] THEN	04418400
BEGIN	04418600
IF E,[41:1] THEN GO TO D22; % INVALID ADDRESS	04418800
IF R,[27:1] THEN % EOT	04419000
EOF:      IF MASK,[42:1] THEN % EOF OK	04419200
BEGIN	04419400
REALEOF:  F:=18(IF R,[24:1] THEN @31 ELSE 0)[CTF];	04419600
T,[5:8] := @40;	04419800
GO TO FIX;	04420000
END	04420200
ELSE	04420400
BEGIN % EOF NOT ACCEPTABLE	04420600
P(@40);	04420800
GO TO SIX;	04421000
END;	04421200
TAPERETRY:	04421400
MAKEMLOG(=TAPEBUFFERSIZE);	04421600
IF (TI=TAPEPARITYRETRY(R,U,KEY)).[5:8]=32 AND	04421800
LOCATQUE[S],[3:5] NEQ 0 THEN GO TO REALEOF;	04422000
P(T,[5:8]);	04422200
GO TO SIX;	04422400
END;	

IF E,[41:1] THEN % WRITE RING	04422600
IF E,[43:1] THEN % PARITY,WRITE RING	04422800
BEGIN	04423000
F:=6; % WRITE LOCK	04423200
GO TO L2;	04423400
END	04423600
ELSE GO TO D22; % INVALID ADDRESS	04423800
IF E,[43:1] THEN GO TO TAPERTRY; % PARITY,WRITE RING ONLY	04424000
P(@40);	04424200
SIX:	04424400
T := T&P(XCH)[5:40:8];	04424600
F := 1;	04424800
FIX:	04425000
E := T,[5:8]×F;	04425200
FIN := S;	04425400
IOD := IOQUE[S];	04425600
SEVEN:	04425800
RETURN IOSPACE(S);	04426000
	04426200
IOCOUNT[LOCATQUE[S],[3:5]]:=P(DUP,LOD)-1;	04426400
T,[FF]:=S:=LOCATQUE[S],[FF];	04426600
IF F = @3100001 THEN	04426800
IF S NEQ @77777 THEN GO TO SEVEN;	04427000
S:=FIN;	04427200
IF FALSE THEN	04427400
LEAVE:	04427600
IOD := IOQUE[S];	04427800
FIN := FINALQUE[S] AND NOT MEMORY;	04428000
IF IOD,[24:1] THEN	04428200
BEGIN	04428400
NT4 := M[IOD INX ( IF IOD,[22:1] THEN 1 ELSE NOT 0)];	04428600
FINISHOFFIO(U);	04428800
END;	04429000
IF ( T1:= FIN) LSS 0 THEN	04429200
P(R&E[25:40:8]&IOD[3:3:5] OR IOMASK,LOCATQUE[S],+)	04429400
ELSE	04429600
BEGIN	04429800
IF E NEQ 0 THEN	04430000
BEGIN	04430200
P(,T1,PRL);	04430400
T1 := T1&E[25:40:8];	04430600
END	04430800
ELSE P(,T1,IOR);	04431000
LOCN := [M[LOCATQUE[S]]];	04431200
IOD := IOD,[33:15];	04431400
WHILE LOCN[0],[33:15] NEQ IOD DO LOCN := 1 INX LOCN;	04431600
LOCN[0] := P(,T1,LOD);	04431800
END;	04432000
UNIT[U] := T;	04432200
CLEAR:	04432400
UNIT[U] := (*P(DUP))&F[5:20:13];	04432600
STARTIO(U);	04432800
KILLL:	04433000
LOCATQUE[S],[11:1]:=0;	04433200
KILLER:	04433400
KILL(IR) INX NOT 1);	04433600
END;	04433800
% SET OMIT = NOT(DEBUGGING)	04544999
REAL NSTOP,B,C,ERROR,NSYMBS,LP;%	04545000
ARRAY WB[*],RBX[*],TBL[*],STOPS[*];	04546000

REAL TYPETOG;%	04547000
\$ POP OMIT	04547001
REAL PROCEDURE TAPEPARITYRETRY(R,U,KEY);%	04548000
VALUE R,U,KEY;%	04549000
REAL R,U,KEY;%	04550000
BEGIN REAL T1,T2,T3; INTEGER I= T1;%	04551000
REAL RESULT,IOD,OID,SPACEMASK,SPACEIOD,M,N,W,MODE;%	04552000
REAL J,K;%	04553000
REAL ERASEIOD=SPACEMASK;%	04554000
REAL Z,Y,MIX,BSIZE;	04554100
LABEL XIO,GIVEUP;	04554200
LABEL RP,LX;	04554300
REAL SIZE,T4,LIMIT;	04554500
REAL PTR,BUFFER,BUFFERSIZE,%	04554600
PATTERN,PATTERN1,PATTERN2,PATTERNWORD;% DON'T CHANGE ORDER	04554700
BOOLEAN TESTING,SPACING;%	04554800
LABEL XXIT,EXIT;%	04555000
SUBROUTINE RECORDRETRY;%	04555050
BEGIN%	04555100
IF PTR=KEY = TAPEBUFFERSIZE=1 THEN%	04555150
BEGIN%	04555200
T4:=GETSPACE(TAPEBUFFERSIZE,9,5)+2;	04555250
MOVE(10,KEY,T4);%	04555300
MEMORY[KEY+8]:= TAPEBUFFERSIZE=10;%	04555350
MEMORY[KEY+9]:= 1023;%	04555400
LINKUP(3,KEY);%	04555450
KEY:= T4; PTR:= KEY+9;%	04555500
END;%	04555550
MEMORY[PTR:=PTR+1]:= IOD;%	04555600
MEMORY[PTR:=PTR+1]:= RESULT & RDCTABLE[U][19:1:2];%	04555650
END RECORDRETRY;%	04555700
SUBROUTINE DOIONOW;%	04556000
BEGIN FOR Y+1 STEP 1 UNTIL 18 DO	04556100
BEGIN IF R,[24:1]THEN	04557000
BEGIN % WAIT 1/15 SEC BETWEEN READ RETRIES	04557100
WHILE T4>CLOCK+P(RTR) DO SLEEP(1,1);	04557200
T4=CLOCK+P(RTR)+4;	04557300
END;	04557400
IF IOQUESLOTS=0 THEN SLEEP([IOQUESLOTS],63);	04558000
IOQUESLOTS:=IOQUESLOTS-1;	04558500
IOQUEAVAIL:=IOQUE[T1:=IOQUEAVAIL];	04559000
IOQUE[T1]+ IOD;%	04560000
IOCOUNT[(LOCATQUE[T1]+LOCATQUE[T2+(T3+UNIT[U])],[18:15]]&	04561000
[RESULT][CTC]&T2[CTF]],[3:5]]+*P(DUP)+1;	04562000
UNIT[U] + T3&T1[18:33:15]&64[5:35:13];%	04563000
STARTIO(U);%	04564000
FINALQUE[T1] + NABS(IOD)& 0 [25:40:8] OR IOMASK;%	04565000
RESULT + 0;%	04566000
SLEEP([UNIT[U]],@100000000000);%	04567000
IF RESULT,[30:1] THEN % NOT READY	04567010
BEGIN	04567020
MODE := (-16);	04567030
GO TO EXIT;	04567040
END;	04567050
IF RESULT,[29:1] AND RESULT,[2:1] THEN	04567100
BEGIN	04567150
IF RESULT,[12:1] THEN % BLANK TAPE	04567200
IF IOD,[24:1] THEN % READ	04567250
TRANSACTION[U]+TRANSACTION[U]-1&IOD[1:22:1] ELSE	04567300
BEGIN; % WRITE	04567310

STREAM(A:=TINU[U],T:=T2:=SPACE(3));	04567320
BEGIN SI+LOC A; SI+SI+5; DS+3 CHR;	04567400
DS+21 LIT" BLANK TAPE ON WRITE+";	04567500
END;	04567550
SPOUTIT(T2,35);% BLNK TAPE=TAPEPARITYRTRY	%R6404567600
GO TO XXIT;%	04567700
END;	04567750
IF RESULT,[11:1] THEN * MEM PARITY	04567770
BEGIN;	04567780
STREAM(A+TINU[U],T+T2+SPACE(3));	04567790
BEGIN SI+LOC A; SI+SI+5; DS+3 CHR;	04567800
DS+13 LIT" I/O MEM PAR+";	04567810
END;	04567820
SPOUTIT(T2, 35);%I/O MEM PAR=TAPEPARITYRETRY	%R6404567830
XXIT: MODE I= 16;	04567840
IF TESTING THEN GO XIO;	04567845
RECORDRETRY;	04567850
GO TO EXIT;	04567855
END;	04567860
IF RESULT,[13:2]≠0 THEN Y+18;	04567870
END ELSE GO TO XIO;	04567900
END;%	04568000
RESULT,[27:1]+1; MODE+32;	04568100
XIO: IF NOT SPACING THEN RECORDRETRY;	04568200
END DOIONOW;%	04568250
SUBROUTINE SPACEBACK;	04568300
BEGIN	04568310
IF TRANSACTION[U]=1 THEN	04568320
BEGIN	04568330
IOD:=#4200000000&0IOD[3:3:5];	04568340
DOIONOW;	04568350
I:=TWO(U);	04568360
RRRMECH:=RRRMECH OR I; % MASK OUT STATUS WHILE	04568362
CLICK:=CLOCK+P(RTR)+600; % WE WAIT FOR REWIND,	04568364
COMPLEXSLEEP((P(RRR) AND I)≠0);	04568366
RRRMECH:=RRRMECH AND NOT I;	04568368
IF (P(RRR) AND I)=0 THEN % TIME OUT => NOT READY	04568370
BEGIN MODE:=16;	04568372
GO TO EXIT;	04568374
END;	04568376
END ELSE	04568380
BEGIN	04568390
M:=W;	04568400
IOD:=SPACEIOD;	04568410
J:=0;	04568420
SPACING:= TRUE;%	04568425
DO BEGIN	04568430
DOIONOW;	04568440
TRANSACTION[U]:=(P(DUP))+1;	04568450
J:=J+1;	04568460
END UNTIL ((M:=RESULT,[CF]=SPACEIOD,[CF]+M) LSS 0	04568470
OR RESULT,[27:1]) AND J GTR 1;	04568480
IF NOT TESTING THEN SPACING:= FALSE;	04568485
TRANSACTION[U]:=(P(DUP))-2;	04568490
IOD:=SPACEIOD&0[22:47:1];	04568500
DOIONOW;	04568510
IF N=0 THEN BSIZE:=RESULT,[CF]=IOD,[CF] ELSE	04568520
IF BSIZE≠RESULT,[CF]=IOD,[CF] THEN	04568530
BEGIN	04568540
STREAM(A:=TINU[U],D:=T2:=SPACE(10));	04568550

```

                BEGIN SI:=LOC A;SI:=SI+5;DS:=3 CHR;          04568560
                  DS:=13 LIT" ERASE ERROR*";                04568570
                END;                                        04568580
                SPOUT(T2);                                  04568590
                GO GIVEUP;                                  04568600
            END;                                           04568610
        END;                                              04568620
    END; % OF SPACEBACK                                    04568630
    TINU[U],[18:12] + P(DUP),[18:12]+1;%                  04569000
    MIX := LOCATQUE[UNIT[U],[FF]],[3:5];                  04569500
    OIOD * NFLAG(IOQUE[UNIT[U],[18:15]]);%                04570000
    PTR:= KEY+9;                                          04570100
    STREAM(A:=TINU[U],REEL:=PRNTABLELU],[30:18],          %11504570300
          MFID:=MULTITABLE[U],FID:=IF LABELTABLE[U]*@314 %11504570305
          THEN "UNLABLD" ELSE LABELTABLE[U],             %11504570307
          RD:=R,MIX:=MIX,T:=T2:=SPACE(10));              %11504570310
    BEGIN                                                 %11504570320
        SI:=LOC A; SI:=SI+5;DS:=3CHR;                    %11504570330
        DS:=7 LIT " REEL #"; DS:=4 DEC;                  %11504570340
        SI:=SI+1;DS:=LIT"/";DS:=7CHR;SI:=SI+1;DS:=LIT"/";DS:=7CHR;%11504570345
        DS:=4 LIT";RD=";                                   %11504570370
        16(DS:=3RESET;3(IF SB THEN DS:=SET ELSE           %11504570380
          DS:=RESET; SKIP SB));                          %11504570390
        DS:=5 LIT ";MIX="; DS:=2DEC;                    %11504570400
        DS:=LIT LEFTARROW;                               %11504570410
    END STREAM;                                          %11504570420
    SPOUTIT(T2,34);                                      %11504570430
    IF R,[24:1] THEN%                                     04571000
    BEGIN COMMENT READ RETRY;%                            04572000
        SPACEMASK * OIOD,[21:2]*@1111 EQV NOT @0123;%    04573000
        SPACEIOD * OIOD&1[8:38:10]&1[23:47:1];%         04574000
        FOR M * 1 STEP 1 UNTIL 3 DO%                      04575000
            BEGIN SPACEIOD * SPACEIOD&SPACEMASK[21:46:2];% 04576000
                FOR N * 1 STEP 1 UNTIL 5 DO%             04577000
                    BEGIN IOD * SPACEIOD;%              04578000
                        IF N#1 OR M#1 THEN DOIONOW ELSE 04579000
                        IF NOT(R,[29:1]AND R,[2:1] AND R,[12:1]) 04579100
                        THEN DOIONOW;                    04579200
                        IF RESULT,[28:1] THEN%           04580000
                            BEGIN MODE * 0;%           04581000
                                IOD * OIOD;%           04582000
                            END%                         04583000
                        ELSE BEGIN MODE * 8;%           04584000
                            IOD * OIOD&SPACEMASK[21:43:2];% 04585000
                        END;%                             04586000
                    DOIONOW;%                            04587000
                    IF NOT RESULT,[28:1] THEN GO TO EXIT;% 04588000
                    IF MOD3IOS THEN IF OIOD,[23:1] THEN 04588010
                    BEGIN Z+IOD+OIOD&SPACEMASK[21:40:2] 04588020
                        &(OIOD,[33:15]+(OIOD,[8:10]*1) 04588030
                    &OIOD[1:22:1])[33:33:15];          04588040
                    DOIONOW; MODE*0;                    04588050
                    IF RESULT,[28:1] THEN               04588060
                    BEGIN IOD*OIOD; DOIONOW;           04588070
                        IF NOT RESULT,[28:1] THEN      04588080
                        GO TO EXIT;                     04588090
                        IOD+Z&SPACEMASK[21:46:2];     04588100
                        DOIONOW; MODE*8;               04588110
                        IF RESULT,[28:1] THEN          04588120
                        BEGIN IOD*OIOD&SPACEMASK      04588130

```

	[21:43:2];	04588140
RP:	DOIONOW;	04588150
	IF RESULT,[28:1] THEN	04588160
	GO TO LX;	04588170
	GO TO EXIT;	04588180
	END;	04588190
	END;	04588200
	Z=ABS(IOD,[33:15]-RESULT,[33:15]);	04588210
	IF IOD,[21:2]=0 THEN	04588220
	Z=Z-(RESULT,[15:3]=0);	04588230
	IF IOD,[8:10]<Z THEN	04588240
	BEGIN IOD=0IOD; MODE=0; GO TO RP END;	04588250
	IF IOD,[22:1] THEN	04588260
	STREAM(Z,Y=Z DIV 64,	04588270
	S=RESULT,[33:15]+1,	04588280
	SK=(RESULT,[15:3]+1),[45:3],	04588290
	GM=(IF IOD,[21:1] THEN 0	04588300
	ELSE "+"),	04588310
	D=0IOD,[33:15]);	04588320
	BEGIN SI=S; SI=SI+SK;	04588330
	Y(16(DS+32 CHR));	04588340
	Z(DS+8 CHR);	04588350
	SK(DS=LIT "0");	04588360
	DI=DI-SK; SI=LOC GM;	04588370
	SI=SI+7; DS=CHR;	04588380
	END ELSE	04588390
	STREAM(Z,Y=Z DIV 64,	04588400
	S=RESULT,[33:15]-1,	04588410
	SK=(RESULT,[15:3]+7),[45:3],	04588420
	FL=(IF IOD,[21:1] THEN 0	04588430
	ELSE @14),	04588440
	FK=(8-RESULT,[15:3]),[45:3],	04588450
	D=0IOD,[33:15]);	04588460
	BEGIN SI=S; SI=SI+SK; DI=DI+7;	04588470
	Y(16(32(DS+CHR; SI=SI-2;	04588480
	DI=DI-2)));	04588490
	Z(8(DS+CHR; SI=SI-2; DI=DI-2));	04588500
	SI=LOC FL; SI=SI+7;	04588510
	FK(DS+CHR; SI=SI-1; DI=DI-2);	04588520
	END;	04588530
	IOD=@140000005&0IOD[22:22:1]	04588540
	&0IOD[3:3:5];	04588550
	DOIONOW; GO TO EXIT;	04588560
LX:	END;	04588570
	END;%	04589000
N *	IF TRANSACTION[U] < 15 THEN%	04590000
	TRANSACTION[U] ELSE 15;%	04591000
IOD *	SPACEIOD&SPACEMASK[21:40:2];%	04592000
SPACING	= TRUE;	04592100
FOR W *	1 STEP 1 UNTIL N DO%	04593000
	BEGIN DOIONOW;%	04594000
	IF RESULT,[27:1] THEN N=0;%	04595000
	END;%	04596000
IOD *	SPACEIOD&SPACEMASK[21:37:2];%	04597000
FOR N *	3 STEP 1 UNTIL W DO DOIONOW;%	04598000
IOD *	0IOD;%	04599000
MODE *	0;%	04600000
SPACING	= FALSE;	04600100
DOIONOW	;	04601000
IF NOT RESULT,[28:1] THEN	GO TO EXIT;%	04602000

```

        END;%
        MODE + 16;%
    END ELSE BEGIN COMMENT WRITE RETRY;%
        LIMIT+@100000;
        ERASEIOD + (SPACEIOD + 010D&0[8:38:10]&7[22:45:3]&[T2]%
            [33:33:15])&@112[18:41:7];%
        W + R,[33:15]-010D,[33:15]+2;%
        WHILE TRUE DO
            BEGIN
                SPACEBACK;
                IF (N1=N+W+128) GTR LIMIT OR
                    TERMSET(MIX) THEN GO GIVEUP;
                IOD + ERASEIOD&N[9:39:9];%
                SPACING:= TRUE;%
                FOR J + 0 STEP 512 UNTIL N DO%
                    BEGIN TRANSACTION[U] + TRANSACTION[U]-1;%
                    DOIONOW;%
                    IOD + ERASEIOD&1[8:47:1];%
                IF RESULT,[27:1] THEN
                    BEGIN
                        IF NOT R,[27:1] THEN LIMIT+J+3000;
                        R,[27:1]+1;
                    END;
                END;%
                SPACING:= FALSE;%
                IOD:= IOD & N[10:10];%
                RECORDRTRY;%
                IOD + 010D;%
                DOIONOW;%
                IF RESULT,[27:1] THEN R,[27:1] + 1;%
                IF NOT RESULT,[28:1] THEN%
                    BEGIN
                        SIZE+RESULT,[CF]-010D,[CF];
                        SPACEBACK;
                        IOD+SPACEIOD&0[22:47:1];
                        DOIONOW;
                        IF NOT(RESULT,[28:1] OR (010D,[21:1] AND
                            (RESULT,[CF]-SPACEIOD,[CF]#SIZE))) THEN
                            BEGIN
                                MODE+0&R[42:27:1];
                                GO TO EXIT;
                            END;
                        END;
                    END;%
                END;%
            END;%
        GIVEUP;
        T2 := SPACE(3);
        STREAM(A+TINU[U],T2);%
        BEGIN SI + LOC A; SI + SI+5; DS + 3 CHR;%
        DS + 11 LIT " WR PARITY=";%
        END;%
        SPOUTIT(T2,35);% WR PAR
        MODE + 16;%
    END;%
    EXIT; TAPEPARITYRETRY:= UNIT[U] & MODE[5:40:8];
        MEMORY[KEY+8] := PTR-KEY-9;
        MEMORY[KEY+9]:=ABS(MODE);
        MEMORY[KEY] := P(DUP,LOD) & ((PTR-KEY) DIV 5)[39:39:9];
        IF (MODE#16) OR (R,[24:1]) THEN LINKUP(3,KEY) ELSE
            BEGIN
                BUFFER:= 010D INX 0;

```

```

04603000
04604000
04605000
04605500
04606000
04607000
04608000
04609000
04610000
04611000
04627000
04627100
04628000
04628100
04629000
04630000
04631000
04632000
04633000
04633100
04633200
04633300
04633400
04634000
04634100
04634200
04634300
04635000
04636000
04637000
04638000
04638100
04638200
04638300
04638650
04638700
04638800
04638900
04639000
04639100
04639200
04640000
04641000
04642000
04642900
04643000
04644000
04645000
04646000
04647000
&R6404648000
04649000
04650000
04651000
04651050
04651100
04651200
04651300
04651400
04651500

```



BUFFERSIZE:= 010D,[8:10];	04651600
IF NOT 010D,[21:1] THEN % ALPHA WRITE = CHECK Q-MARKS	04651700
BEGIN	04651800
STREAM(T:=0;	04651900
TEMP:=0, SVSI:=0,	04652000
BUFFSTART:=BUFFER,	04652100
BUFFEND:=BUFFER+BUFFERSIZE);	04652200
BEGIN	04652300
SI:=BUFFEND; DI:=LOC TEMP; DS:= CHR;	04652400
DI:=BUFFEND; DS:=LIT""; DI:=DI-1; DS:=RESET; %Q-MARK	04652500
SI:=BUFFSTART;	04652600
IF SC > 9 THEN	04652700
BEGIN	04652800
L1: SI:=SI+1; IF SC>9 THEN GO L1;	04652900
END;	04653000
L2: SI:=SI+1; IF SC<9 THEN GO L2;	04653100
SVSI:=SI;	04653200
SI:=LOC SVSI; SI:=SI+5;	04653300
DI:=LOC BUFFEND; DI:=DI+5;	04653400
IF 3 SC#DC THEN TALLY:=1;	04653500
DI:=BUFFEND; SI:=LOC TEMP; DS:= CHR;	04653600
T:=TALLY;	04653700
END;	04653800
I:=POLISH;	04653900
MEMORY[KEY+2]:= P(DUP,LOD) & I[1:47:1];	04654000
END;	04654100
IF STOPTTEST THEN LINKUP(3,KEY) ELSE	04654200
BEGIN	04654300
MEMORY[KEY] := NABS(P(DUP,LOD));	04654400
LINKUP(3,KEY);	04654500
TESTING:= SPACING:= TRUE; N:=0;	04654600
BUFFERSIZE:= BUFFERSIZE-1;	04654700
010D:= 010D & 1[18:42:6];	04654800
PTR:= KEY+8;	04654900
STREAM(MOD2IOS:=NOT(MOD3IOS+62), D:=[PATTERN]);	04655000
BEGIN	04655100
DS:=13 LIT"01248+x+<<(,G<";	04655200
MOD2IOS(DI:=DI-6; DS:=LIT""; DI:=DI+5);	04655300
DS:= LIT""; DS:= LIT"";	04655400
DS:=3 LIT" ]S(";	04655500
END;	04655600
SLEEP([MEMORY[KEY]],@10000000000000000);	04655700
MEMORY[PTR]:= 0; MOVE(191,PTR,PTR+1);	04655800
FOR K:=0 STEP 1 UNTIL 15 DO	04655900
BEGIN	04656000
STREAM(A:=[PATTERN],	04656100
K:=K+(K=15), M:=4+4x(K<14), N:=1+(K>13),	04656200
SIZEDIV64:=BUFFERSIZE,[36:6], BUFFERSIZE,	04656300
BUFFER);	04656400
BEGIN	04656500
SI:=A; SI:=SI+K;	04656600
M(DS:=N CHR; SI:=SI-N);	04656700
SI:=BUFFER;	04656800
SIZEDIV64(DS:=32 WDS; DS:=32 WDS); DS:=BUFFERSIZE WDS;	04656900
DI:=A; DI:=DI+24; DS:=WDS;	04657000
END;	04657100
I0D:= 010D:= 010D & ((K<7) OR (K>13))[21:47:1];	04657200
DOIONOW;	04657300
MEMORY[PTR]:= RESULT & RDCTABLE[U][19:1:2];	04657400
SPACEBACK;	04657500

STREAM(SIZEDIV64:=BUFFERSIZE,[36:6],BUFFERSIZE, BUFFER);	04657600
BEGIN	04657700
DS:=8 LIT" "; SI:=BUFFER;	04657800
SIZEDIV64(DS:=32 WDS; DS:=32 WDS); DS:=BUFFERSIZE WDS;	04657900
END;	04658000
IOD:= OIOD & 1[24:47:1];	04658100
DOIONOW;	04658200
MEMORY[PTR+1]:= RESULT & RDCTABLE[U][19:1:2];	04658300
STREAM(A:=[PATTERN] INX 3, CHRERR:=0, WRDERR:=0, WRDCNT:=0, LOOP:=0, FORSEVEN:=1, LEAPFROG:=0, WDSLEFT:=1:=(J:=IF (SIZE:=ABS(BUFFER=(RESULT INX 0))) LEQ BUFFERSIZE THEN SIZE ELSE BUFFERSIZE+1) MOD 63, V:=IF J<64 THEN J ELSE 63, N:=IF J<64 THEN 1 ELSE J DIV 63, RECYCLE:= IF J<64 THEN 0 ELSE IF I=0 THEN 0 ELSE 1, TEMP:=0, SVDI:=0, BITLOCN:=PTR+3, WRDLOCN:=PTR+5, BUFFER);	04658400
BEGIN;	04658500
LEAPFROG:= CI; TALLY:=0; % USED ONLY FOR LEAPFROG RETURN	04658600
NCV(SI:=A; IF 8 SC#DC THEN	04658700
BEGIN	04658800
SI:=WRDERR; SI:=SI+8; WRDERR:=SI;	04658900
FORSEVEN(SVDI:=DI; DI:=BITLOCN; LOOP(DI:=DI+2);	04659000
SI:=LOC WRDCNT; SI:=SI+6; DS:=2 CHR;	04659100
DI:=WRDLOCN; LOOP(DI:=DI+8);	04659200
SI:=SVDI; SI:=SI+8; DS:= WDS;	04659300
TALLY:=LOOP; TALLY:=TALLY+1; LOOP:=TALLY;	04659400
SI:=LOC LOOP; SI:=SI+7;	04659500
IF SC="7" THEN	04659600
BEGIN TALLY:=0; FORSEVEN:=TALLY; END;	04659700
DI:=SVDI);	04659800
SI:=A; DI:=DI-8; TALLY:=0;	04659900
8(IF SC#DC THEN TALLY:=TALLY+1);	04660000
TEMP:=TALLY;	04660100
SI:=CHRERR; TEMP(SI:=SI+8); CHRERR:=SI;	04660200
END;	04660300
SI:=WRDCNT; SI:=SI+8; WRDCNT:=SI;	04660400
));	04660500
RECYCLE(TALLY:=1; N:=TALLY;	04660600
TALLY:=WDSLEFT; V:=TALLY;	04660700
TALLY:=0; RECYCLE:=TALLY;	04660800
JUMP OUT TO TADPOLE);	04660900
GO TO FROG;	04661000
TADPOLE:	04661100
FROG:	04661200
CI:=LEAPFROG;	04661300
DI:=BITLOCN; DI:=DI-5;	04661400
SI:=LOC CHRERR; SI:=SI+5; DS:=3 CHR;	04661500
SI:=LOC WRDERR; SI:=SI+6; DS:=2 CHR;	04661600
END;	04661700
IF MEMORY[PTR],[27:1] THEN SPACEBACK;	04661800
PTR:=PTR+12;	04661900
END;	04662000
MEMORY[KEY]:= P(DUP,L0D) & 0[1:1:2] & 39[39:39:9];	04662100
MEMORY[KEY+2]:= P(DUP,L0D) & OPTION[2:2:1];	04662200
LINKUP(20,KEY);	04662300
END;END;	04662400
END TAPEPARITYRETRY;	04662500
REAL PROCEDURE PLACEFINDER(S,A,L);	04662600
	04662700
	04662800
	04662900
	04663000
	04663100
	04663200
	04663300
	04666000
	04700000

VALUE S,A;	04701000
REAL S,A,L;	04702000
BEGIN INTEGER I; ARRAY B[*];	04703000
REAL T,W,E,J,AA;	04704000
LABEL NULL,FOUND,EXIT;	04705000
LABEL SANDA; REAL SS;	04705500
W=-1;	04706000
B := [M[T := SPACE(30)]]&30[8:38:10];	04707000
SS:=S;	04707500
IF S=0 THEN	04708000
NULL: BEGIN STREAM(T); DS:=20 LIT " "; GO EXIT; END;	04709000
DISKWAIT(-T,30,JAR[P1MIX,10]);	04710000
IF (AA*B[0],[FF])=0 THEN	04711000
SANDA: BEGIN STREAM(S:=SS,A,K:=M[PRT[P1MIX,8]],[10:2],T);	04712000
BEGIN DS=5 LIT " S =";	04713000
SI=LOC S; DS=4 DEC;	04714000
DS=5 LIT " A =";	04715000
DS=4 DEC;	04716000
DS:=LIT " "; SI:=SI+7; DS:=CHR;	04716100
DI=T; DI=DI+5; DS=3 FILL;	04717000
DI=T; DI=DI+14; DS=3 FILL;	04718000
END;	04719000
GO TO EXIT;	04720000
END;	04721000
DISKWAIT(-T,30,I+JAR[P1MIX,AA DIV JAR[P1MIX,8]+10]+	04722000
AA MOD JAR[P1MIX,8]+ S DIV 30);	04723000
IF (J+B[S MOD 30])<0 THEN GO TO NULL;	04725000
AA:=I+JAR[P1MIX,J,[CF] DIV JAR[P1MIX,8]+10]+	04726000
J,[CF] MOD JAR[P1MIX,8];	04727000
I=0; J=J,[FF];	04728000
DO BEGIN S=(I+J),[36:11];	04729000
IF W*(W+S DIV 30) THEN	04730000
DISKWAIT(-T,30,AA+W);	04731000
IF (E+B[S=W*30],[38:10])=A THEN GO TO FOUND;	04732000
IF E<A THEN I+S ELSE J+S;	04733000
END UNTIL J=I<1;	04734000
S=I;	04735000
FOUND: L:=B[S MOD 30],[10:28];	04736000
IF L=0 THEN GO TO SANDA;	04736500
STREAM(L+ABS(L),T);	04737000
BEGIN DS:=11 LIT " NEAR LINE ";	04738000
SI=LOC L; DS=8 DEC;	04739000
DS:=LIT " "; DI:=DI-9; DS:=7 FILL;	04740000
END;	04741000
EXIT:PLACEFINDER=T;	04742000
END PLACEFINDER;	04743000
REAL PROCEDURE SECURITYCHECK(MID,FID,USE,HEAD);	04790000
VALUE MID, FID, USE;	04791000
REAL MID,FID,USE,HEAD;	04792000
FORWARD;	04793000
PROCEDURE SHEETDIDDLER(BUFF,TYPE,SID); VALUE BUFF,TYPE,SID;	04798000
REAL BUFF,TYPE,SID; FORWARD;	04799000
PROCEDURE ZIPPER(X,Y); VALUE X,Y; REAL X,Y; FORWARD;	04800000
PROCEDURE DISKLOG(MID,FID,H); VALUE MID,FID,H; ARRAY H[*];	811204800100
REAL MID,FID; FORWARD;	811204800200
DEFINE INOUTK = 21#;	04801100
% SYSTEM/DISK RECORD FORMAT;	04803000
% A[0] = ADAPTER INFO	04804000
% A[1] = USER CODE	04805000
% A[2] = TIME OF LAST ACTIVITY	04806000

REAL INDIAN;	05000000
% FORMATS FOR PARAMETERS TO INDIAN BOY	05010000
% LO = SEARCH;	05010100
%           A[1] = FIRST NAME     : A[1] = SECURITY CODE	05010200
%           A[2] = SECOND NAME    : A[2] = WORD[4] OF HEADER	05010300
%           A[3] = USER CODE      : A[3] = END OF FILE COUNT	05010400
%                                   : A[4] = ADDRESS OF FIRST RECORD	05010500
% L1 = MAKE FILE	05010600
%           A[1] = FIRST NAME     : A[1] = ADDRESS OF FIRST RECORD	05010700
%           A[2] = SECOND NAME    : A[2] = ADDRESS OF HEADER	05010800
%           A[3] = FILE TYPE	05010900
%           A[4] = SIZE OF FILE	05011000
% L3 = FILE REMOVE	05011100
%           A[1] = FIRST NAME	05011200
%           A[2] = SECOND NAME	05011300
% L4 = FILE REPLACE	05011400
%           A[1] = FIRST NAME OF OLD FILE	05011500
%           A[2] = LAST NAME OF OLD FILE	05011600
%           A[3] = FIRST NAME OF NEW FILE	05011700
%           A[4] = LAST NAME OF NEW FILE	05011800
%           A[5],[38:10]=SAVE FACTOR ( ZERO IF NOT SPECIFIED %SM	05011900
%           A[5],[32:6]=FILE TYPE CODE ( FROM CANDE ) %SM	05012000
%           A[5],[31:1]="AUTO UNLOCK" REQUEST	05012100
% L5 = FILE CHANGE	05012200
%                                   : IGNORE CODES :	05012300
%           A[1] = FIRST NAME OF OLD FILE     0 => 1 NOT THERE	05012400
%           A[2] = LAST NAME OF OLD FILE     1 => 1 IN USE	05012500
%           A[3] = FIRST NAME OF NEW FILE    2 => 2 THERE	05012600
%           A[4] = LAST NAME OF NEW FILE     3 => 2 IN USE	05012700
% L6 = JOB STARTER	05012800
%           A[1] = ADDRESS OF SKELETON ENTRY FOR SCHEDULE	05012900
% L7 = FILE SECURER	05013000
%           A[1] = FIRST NAME OF FILE (<0 => NOT BY CANDE)	05013100
%           A[2] = SECOND NAME OF FILE	05013200
%           A[3] = NEW CONTENTS OF HEADER[2]	05013300
%           A[4] = NEW CONTENTS OF HEADER[5]	05013400
%           A[5] = NEW CONTENTS OF HEADER[6]	05019000
%	05019100
%	05020000
PROCEDURE INDIANBOY;	05020100
BEGIN ARRAY A[*],B[*],C[*];	05020200
REAL RCW=+0,1,T;	05020300
LABEL LO,L1,L3,L4,L5,L6,L7,FOG,FORGET,LAST,RETURN,	05020400
RB1,RB2,NX1,NX2,BALLS,FORFOR,MAGUS;	05020500
SWITCH WHAT1=LO,L1,MAGUS,L3,L4,L5,L6,L7,MAGUS,MAGUS,	05020600
MAGUS,MAGUS,MAGUS,MAGUS;	05021000
WHILE INDIAN,[CF]#0 DO	05021100
BEGIN A=IOQUE&INDIAN[CTC];	05021200
LOGLINE + A[0],[FF];	05021300
GO TO WHAT[A[0],[8:10]];	05022000
DO UNTIL FALSE;	05030000
IF (T+DIRECTORYSEARCH(A[1],A[2],5))#0 THEN	05030100
BEGIN IF (A[1]+SECURITYCHECK(A[1],A[2],A[3],T))#0 THEN	05030200
BEGIN A[2]+M[T INX 4]&T[FTF];	05030300
A[3]+M[T INX 7];	05030400
A[4]+M[T INX 10];	05030500
END;	05030600
FORGETSPACE(T);	05030700
END ELSE A[1]+=1;	05030800
GO TO RETURN;	05040000
B := IOQUE&(SPACE(30))[CTC];	
L1:	

	MOVE(30,B,[CF]-1,B);	05040100
	B[8]+ABS(I+A[4]);	05040200
	P(DIRECTORYSEARCH(=A[1],A[2],7),DEL);	05040300
	IF I GTR 0 THEN	05040400
	IF (B[10]:=GFTUSERDISK(I OR MEMORY)) = 0 THEN	05040500
	BEGIN M[A+1]:=0; % NO USER DISK	05040600
	IF A[3] LSS 0 THEN A[0],[18:7] := 25;	05040700
	GO TO FOG; % TELL CANDE	05040800
	END ELSE ELSE I:=0;	05040900
	B[7]+I-1;	05041000
	B[9]+1;	05041100
	B[2]+A[2];	05041200
	B[4]+0&2[9:46:2]&A[3][36:42:6];	05041300
	STREAM(=DATE,X=[B[3]]);	05041400
	BEGIN SI=LOC DATE; DS= 8 OCT;	05041500
	DI=X; DS+2 LIT "+1";	05041600
	SI:=X; SI:=SI+5; DS:=3 CHR; % DATE OF LAST ACCESS %DS	05041700
	END;	05041800
	B[0]+@0003600036000101;	05041900
	B[1]+(XCLOCK+P(RTR))&B[3][6:30:18];	05042000
	A[2]:=EUF(A[1],A[2],B,[CF]-1);	05042100
	B[1]+B[8];	05042200
	A[1]+B[10];	05042300
FOG:	FORGETSPACE(B);	05042400
	GO TO RETURN;	05042500
L3:	IF (T:=DIRECTORYSEARCH(=A[1],=A[2],7)) GEQ 64 THEN	05050000
	REPORTBACK(REMOVED,0,0) ELSE GO RB1;	05050100
	GO TO FORGET;	05050200
L4:	T:=DIRECTORYSEARCH(A[3],=A[4],7);	05060000
	IF T=1 THEN	05060100
RB2:	BEGIN REPORTBACK(IGNORE,(T=1)+2,0); GO FORGET; END;	05060200
BALLS:	BI:=IF T LSS 64 THEN 0 ELSE IOQUE&T[CTC];	05060300
	IF (T:=DIRECTORYSEARCH(A[1],=A[2],4)) LSS 64 THEN	05060400
	BEGIN IF B#0 THEN FORGETSPACE(B);	05060500
RB1:	REPORTBACK(IGNORE,T=1,=(A[1]));	05060600
	GO TO FORGET;	05060700
	END;	05060800
	IF B#0 THEN	05060900
	BEGIN M[T INX 2]+B[2];	05061000
	M[T INX 3],[2:10]:=B[3],[2:10]; %SM	05061100
	M[T INX 5]+B[5];	05061200
	M[T INX 6]+B[6];	05061300
	M[T INX 4],[9:39]:=B[4]&1[11:47:1]; % (SHM)	05061400
	FORGETSPACE(B);	05061500
	END;	05061600
	IF A[5]>0 THEN M[T INX 4],[36:6]:=A[5],[32:6]; %FILE TYPE	05061700
	IF A[5],[38:10] NEQ 0 THEN M[T INX 3],[2:10]:=A[5] %SM	05061800
ELSE	IF M[T INX 3],[2:10] LSS 7 THEN M[T INX 3],[2:10]:=7;%SM	05061900
	IF A[5],[31:1] THEN % "AUTO UNLOCK" FROM CANDE	05062000
	IF NOT(M[T INX 5],[1:1]) THEN % NOT "GUARDED"	05062100
	M[T INX 5]:=M[T INX 6] := 12;	05062200
	M[T+4],[1:3]:=0;	05062300
	DISKLOG(A[1],A[2],IOQUE&T[CTC]);	05062400
	ENTERUSERFILE(=A[3],A[4],T,[CF]-1);	05062500
	FORGETSPACE(T);	05062600
	P(DIRECTORYSEARCH(=A[1],A[2],8),DEL);	05062700
	REPORTBACK(CHANGED,0,0);	05062800
	GO TO FORGET;	05062900
L5:	IF (T:=DIRECTORYSEARCH(=A[3],A[4],5))=0 THEN GO TO BALLS;	05070000
	GO TO RB2;	05070100

L6:	B*IOQUE&A[1][CTC];	05080000
	IF (T:=DIRECTORYSEARCH(B[14],-B[3],3)) LSS 64 THEN	05080100
	BEGIN REPORTBACK(NOTIN,T=1,0);	05080200
FORFOR:	FORGETSPACE(B);	05080300
	GO TO FORGET;	05080400
	END;	05080500
	C*IOQUE&T[CTC];	05080600
	IF C[4],[9;2]=2 THEN	05080700
NX1:	BEGIN REPORTBACK(NOTX,0,0);	05080800
NX2:	FORGETSPACE(T);	05080900
	P(DIRECTORYSEARCH(-B[14],B[3],13),DEL);	05081000
	GO TO FORFOR;	05081100
	END;	05081200
	IF SECURITYCHECK(B[14],B[3],B[24],T)=0 THEN	05081300
	BEGIN REPORTBACK(SECURED,0,0);	05081400
	GO TO NX2;	05081500
	END;	05081600
	A[1],[FF]+T;	05081700
	T,[CF]+B[7]+SPACE(30);	05081800
	DISKWAIT(-T,[CF],30,C[10]);	05081900
	C+C&T[CTC];	05082000
	FOR I:=1 STEP 1 UNTIL 4 DO	05082100
	IF (NOT ABS(C[I]&O[CTC]))#NOT 0 THEN I+7;	05082200
	IF NOT I THEN	05082300
	BEGIN FORGETSPACE(A[1],[FF]); GO TO NX1 END;	05082400
	FOR I+15 STEP 1 UNTIL 22 DO	05082500
	IF B[I]=0 THEN B[I]:=(C[I] OR B[I]);	05082600
	B[23]+(CLOCK+P(RTR)) DIV 60;	05082700
	B[25]+T,[FF];	05082800
	B[26]+LOGLINE;	05082900
	STREAM(A+0;S+P(,SCHEDULEIDS));	05083000
	BEGIN SI+S;	05083100
	47(SKIP SB; SKIP DB; TALLY+TALLY+1;	05083200
	IF SB THEN BEGIN END ELSE JUMP OUT);	05083300
	DS+SET; A+TALLY;	05083400
	END;	05083500
	I+P;	05083600
	B[3]+0&I[8;38;10];	05083700
	FORK(P(,SELECTRUN),A[1],-1,196,1);	05083800
	GO TO FORGET;	05083900
L7:	IF (T:=DIRECTORYSEARCH(ABS(A[1]),-A[2],4)) LSS 64 THEN	05090000
	IF A[1]<0 THEN GO TO FORGET ELSE GO TO RB1;	05090100
	B*IOQUE&T[CTC];	05090200
	I+B[4];	05090300
	IF (B[2]+A[3])<0 THEN I,[36;6]+9;	05090400
	B[4]:=(I AND NOT MEMORY);	05090500
	B[5]+A[4];	05090600
	B[6]+A[5];	05090700
	DISKWAIT(T,[CF],30,T,[FF]);	05090800
	IF A[1]<0 THEN GO TO FORFOR;	05090900
	REPORTBACK(CHANGED,0,0);	05091000
	GO TO FORFOR;	05091100
FORGET:	T+A[0];	05100000
	FORGETAREA(A[0],[2;2],A,[CF]);	05100100
	GO TO LAST;	05100200
RETURN:	T+A[0];	05100300
	QUEVENT(A,[CF],CANDEMIX[A[0],[25;8]]);	05100400
LAST:	INDIAN,[CF]+T;	05100500
	END;	05100600
	INDIAN,[FF]+[INDIAN];	05100700

```

KILL([RCW] INX NOT 2);                                05100800
END INDIAN BOY;                                       05100900
% FORMATS FOR PARAMETERS TO INDIAN GIRL              05210000
% L2 = LOG IN OR OUT                                  05210100
% A[1] = USER CODE                                    05210200
% A[2] = CHARGE CODE                                  05210300
% A[3] = 0 IF ON, 1 IF OFF                            05210400
% A[4] = NUMBER OF SIXTIETHS OF SECONDS TILL AUTO BYE 05210500
% L8 = PAPER TAPE STARTER                             05210600
% A[1] = FIRST NAME OF PAPER TAPE FILE               05210700
% A[2] = USER CODE                                    05210800
% L9 = SCHEDULE I/P FILE REPOSITION                  05210900
% A[1] = RECORD # OF WHERE WE WANT TO BE(-1=> EOF) 05211000
% L10= SCHEDULE I/P REQUEST == NO PARAMETERS         05211100
% L11= SCHEDULE QUERY                                05211200
% A[1] = FIRST NAME ; A[1] = 0 => NOT FOUND          05211300
%                                                    = 1 => NOT SCHEDULE TASK 05211400
%                                                    = 2 => SCHEDULED          05211500
%                                                    = 3 => RUNNING           05211600
%                                                    = 4 => DONE              05211700
%                                                    = 5 => ABORTED          05211750
% A[2] = SECOND NAME; A[2] = IP RECORD # LAST READ 05211800
% L12= SCHEDULE TERMINATE == SAME PARAMETERS & RETURN AS L11 05211900
% L13= STATUS OF RUNNING JOB                          05212000
% A[1]=FIRST NAME OF JOB ; PROCESS TIME              05212100
%                                                    ; -1 IF NOT RUNNING      05212200
% A[2]=SECOND NAME OF JOB; IO TIME                   05212300
% A[3]=USERCODE (A[3],[1:1]=1 IF COMPILING)          05212400
%                                                    ; CONTENTS OF R+27 (COMPILER) 05212500
% INDIANGIRL HELPS SATISFY CANDE'S NEEDS EVERY NOW AND THEN, 05219000
% LEAVING THE MORE ENERGETIC TASKS TO INDIANBOY,    05219100
% PROCEDURE INDIANGIRL;                               05220000
% BEGIN ARRAY A[*],B[*],C[*];                         05220100
% REAL RCW=+0,I,T,J;                                  05220200
% LABEL L2,L8,L9,L10,L11,L12,L13;                     05220300
% MAGUS,FORGET,LAST,RETURN,INFORMONLY,RUNING;       05220350
% SWITCH WHAT:=MAGUS,MAGUS,L2,MAGUS,MAGUS,MAGUS,MAGUS, 05220400
% MAGUS,L8,L9,L10,L11,L12,L13;                        05220500
% WHILE INDIAN,[3:15] NEQ 0 DO                         05221000
% BEGIN A:=IOQUE&INDIAN[33:3:15];                      05221100
% LOGLINE:=A[0],[FF];                                  05221200
% GO TO WHAT[A[0],[8:10]];                              05221300
% DO UNTIL FALSE;                                       05221400
% B := IOQUE&(SPACE(30))[CTC];                          05230000
% I:=0;                                                  05230100
% IF (T:=LOGLINE,[40:8]) LEQ LMAX THEN                 05230200
% BEGIN IF (I:=SCHEDLINE[T]) OR T=0 THEN GO INFORMONLY; 05230300
% LONGCARRIAGE[T]:=0;                                   05230400
% END;                                                  05230500
% SYSDISKIO(3,-LOGLINE,B);                              05230600
% B[1]:=IF A[3] THEN 0 ELSE A[1];                       05230700
% B[2]:=CLOCK;                                          05230800
% B[3]:=A[4];                                           05230900
% SYSDISKIO(0,-LOGLINE,B);                              05231000
% INFORMONLY;                                          05231100
% STREAM(X:=IF A[3] THEN " OFF " ELSE " ON ", T, N:=[A[1]], 05231200
% B);                                                  05231300
% BEGIN DS=LIT " "; SI=N; SI=SI+1; DS=7 CHR;          05231400
% SI:=LOC X; SI:=SI+3; DS:=5 CHR;                     05231500
% DS:=3 DEC;                                           05231600

```

	B*DI; DI*DI=3; DS*2 FILL; DI*B;	05231700
	SI*N; SI*SI+8;	05231800
	IF SC="0" THEN	05231900
	BEGIN DS=LIT "(";	05232000
	SI*SI+1; DS*7 CHR;	05232100
	DS=LIT")";	05232200
	END;	05232300
	DS:=LIT"+";	05232400
	END;	05232500
	SPOUTIT(B,A[3],[47:1]x2+INOUTK);	05232600
	IF I THEN	05232700
	FORK(P(,SCHEDIDLE),(-T),0,160,0);	05232800
FORGET:	T*A[0];	05232900
	FORGETAREA(A[0],[2:2],A,[CF]);	05233000
	GO TO LAST;	05233100
L8:		05240000
	T:="OK#*1";	05240100
	TWXOUT([T],[CF],8,-0,LOGLINE);	05240200
	STABLE[LOGLINE,[40:8]]:=NABS(*P(DUP))&0[4:4:1]; % PT&CE	05240300
	GO TO FORGET;	05240400
L9:	%GO TO RECORD FOR SCHED FILE.	05250000
	STABLE[T:=LOGLINE,[40:8]],DIALEDUP:=0; %TURN OFF FOR A	05250100
	IF A[1] LSS 0 THEN A[1]:=M[SEQARRAY[T] INX 78];	05250200
	SYSDISKIO(3,-T,B:=(IOQUE&(GETAREA(1)+1)[CTC]));	05250300
	B[3]:=M[SEQARRAY[T] INX 37]&A[1][CTF]; SYSDISKIO(0,-T,B);	05250400
	FORGETAREA(1,B INX NOT 0);	05250500
	FORK(P(,SCHEDIDLE),T,0,160,0);	05250600
	GO TO FORGET;	05250700
L10:	%NOTIFY THAT C&E READY FOR SCHEDULE I/P	05260000
	IF CANDEINPUTREADY AND STABLE[T:=LOGLINE,[40:8]],DIALEDUP	05260100
	THEN FORK(P(,SCHEDIO),-T,0,125,1)	05260200
	ELSE SEQARRAY[T]:=NABS(*P(DUP));	05260300
	GO TO FORGET;	05260400
L11:	%CHECK STATUS OF SCHEDULED TASK	05270000
L12:	%TERMINATE SCHEDULE TASK	05280000
	IF (T:=DIRECTORYSEARCH(A[1],A[2],5))=0 THEN	05280100
	BEGIN A[1]:=0; GO TO RETURN; END;	05280200
	I:=M[T+6]; J:=M[T+4],[36:6]; FORGETSPACE(T);	05280300
	IF (I OR @77777) NEQ NOT 0 THEN	05280400
	BEGIN	05280500
	A[1]:=1+((J=TYPEINFO)*3)+((J=63)*4); % 63 = ABORTED	05280510
	GO RETURN;	05280520
	END;	05280530
	T:=DIRECTORYSEARCH(((I INX "FILO000")&SYSNO[24:42:6]),-"SCHEDUL",4);	05280600
	IF T=0 THEN	05280700
	BEGIN A[1]:=4; GO TO RETURN END;	05280800
	IF T NEQ 1 THEN	05280900
	BEGIN IF A[0],[8:10]=12 THEN %THIS IS A TERMINATE	05281000
	IF NOT(M[T+6],[2:1]) THEN %NOT ALREADY "STOPPED"	%09805281100
	BEGIN M[T+6]:=*P(DUP) OR M;	%09805281200
	FORK(P(,SCHEDIDLE),I,0,160,0);	05281300
	END;	05281400
	HEADERUNLOCK(((I INX "FILO000")&SYSNO[24:42:6]),"SCHEDUL",T);	05281500
	A[1]:=2; GO TO RETURN;	05281600
	END;	05281700
	A[1]:=3; A[2]:=0;	05281800
	J:=1;	05281900
	DO	05282000
	BEGIN IF SCHEDLINE[J] THEN ELSE GO TO RETURN;	05282100
	IF NOT SCHEND[J] THEN%MAKE SURE NOT BEING TERMINATED	05282200



```

IF (C[1]=[M[SEQARRAY[J]]]&80[8:38:10]),[CF]GTR 511 THEN 05282300
IF C[70]=1,[CF] THEN 05282400
BEGIN IF NOT(C[32],[1:1]) THEN SLEEP([C[32]],0); 05282500
C[32]:=ABS(*P(DUP)); 05282600
$ SET OMIT = SHAREDISK 05282700
LOCKDIRECTORY; 05282800
$ POP OMIT 05282900
DISKWAIT(=(T:=SPACE(30)),-30,C[36]); 05283000
IF (M[T+7]=C[37]=3) LSS 0 THEN M[T+7]=0; 05283100
DISKWAIT(T,-30,C[36]); FORGETSPACE(T); 05283200
$ SET OMIT = SHAREDISK 05283300
UNLOCKDIRECTORY; 05283400
$ POP OMIT 05283500
C[32]:=NABS(*P(DUP)); A[2]:=C[77]; 05283600
IF A[0],[8:10]=12 THEN %TERMINATE 05283700
BEGIN IF (T:=STATABLE[J],MIXNR) GTR 0 THEN 05283800
IF T NEQ CANDYINX THEN 05283900
BEGIN TERMINATE(T&61[CTF]); 05284000
HALT; NOPROCESSTOG:=NOPROCESSTOG-1; 05284100
END ELSE SHEETDIDDLER(0,-1,J); 05284200
C[77]:=C[78]; %FORCE EOF ON I/P 05284300
STREAM(T:=T:=SPACE(5)); 05284400
DS:=26 LIT***TASK TERMINATED BY USER*** 05284500
SCHEDIO(26,1,T&J[CTF]); 05284600
FORGETSPACE(T); 05284700
END; 05284800
GO TO RETURN; 05284900
END; 05285000
END UNTIL (J:=J+1) GEQ LMAX; 05285100
GO TO RETURN; 05290000
L13: % STATUS OF RUNNING JOB 05290100
FOR I := 1 STEP 1 UNTIL MIXMAX DO 05290200
IF JAR[I,*] NEQ 0 THEN 05290300
BEGIN 05290400
TABCNT[I]:=TABCNT[I]+1; 05290500
IF (T:=PUTORTAKE(I,[JAR[I,0]],1&1[2:47:1],0)) NEQ NOT 0 THEN 05290600
IF (A[1],[6:42] EQV T,[6:42]) = NOT 0 THEN 05290700
IF (A[2],[6:42] EQV PUTORTAKE(I,[JAR[I,1]],1,0),[6:42])= 05290800
NOT 0 THEN 05290900
IF (A[3],[6:42] EQV PUTORTAKE(I,[UV[I,4]],1,0),[6:42])= 05291000
NOT 0 THEN GO TO RUNING; 05291100
TABCNT[I]:=TABCNT[I]-1; 05291200
END; 05291300
A[1]:=-1; GO TO RETURN; 05291400
RUNING; 05291500
A[1]:=PUTORTAKE(I,[JAR[I,3]],1,0)+PUTORTAKE(I,[PROCTIME[I]],1,0); 05291600
IF I=P2MIX THEN A[1]:=(P(DUP))+CLOCK+P(RTR); 05291700
A[2]:=PUTORTAKE(I,[JAR[I,4]],1,0)+PUTORTAKE(I,[IOTIME[I]],1,0); 05291800
IF A[3],[1:1] THEN A[3]:=PUTORTAKE(I,[PRT[I,@27]],1,0); 05291900
TABCNT[I]:=TABCNT[I]-1; 05292000
GO TO RETURN; 05292100
RETURN; T=A[0]; 05300000
QUEVENT(A,[CF],CANDEMIX[A[0],[25:8]]); 05300100
LAST; INDIAN,[3:15]*T; 05300200
END; 05300300
KILL([RCW] INX NOT 2); 05300400
END INDIAN GIRL; 05300500
PROCEDURE COMM15; % INDEPENDENT STARTER FOR CANDE 05407000
BEGIN ARRAY A[*],B[*]; 05408000
REAL RCW=-4; 05409000

```

	REAL I,N;	05410000
	LABEL BOY,GIRL;	05410100
	SWITCH TYPE=BOY,BOY,GIRL,BOY,BOY,BOY,BOY,	05410200
	BOY,GIRL,GIRL,GIRL,GIRL,GIRL,GIRL;	05410300
	B=IOQUE&RCW[FTC];	05411000
	N=[RCW],[CF]-B,[CF]-3;	05412000
	A=B&GETAREA(N>4)[CTC];	05413000
	A[0]+(*P(DUP))&B[1][CTF]&B[2][8:38:10];	05414000
	FOR I=1 STEP 1 UNTIL N DO	05415000
	A[I]+B[I+2];	05416000
	GO TYPE[B[2]];	05416100
BOY:	IF INDIAN,[CF]=0 THEN	05417000
	FORK(P(.INDIANBOY),0,0,128,1);	05418000
	M[INDIAN,[FF]], [CF]+A;	05419000
	INDIAN,[FF]+A;	05420000
	GO TO RETURN;	05421000
GIRL:	IF (I=INDIAN,[3:15]) = 0 THEN	05421100
	BEGIN INDIAN,[3:15]=A;	05421200
	FORK(P(.INDIANGIRL),0,0,128,1);	05421300
	END	05421400
	ELSE	05421500
	BEGIN WHILE (N:=M[I],[CF]) NEQ 0 DO I:=N;	05421600
	M[I],[CF]=A;	05421700
	END;	05422000
	END;	05423000
	PROCEDURE SYSDISKIO(IO,LINE,A);	05424000
	VALUE IO,LINE,A;	05425000
	ARRAY A[*];	05426000
	REAL IO,LINE;	05427000
	BEGIN LABEL EXIT,OUT;	05428000
	OWN REAL X;	05429000
	REAL I,J,T;	05430000
	LINE+=255 AND LINE;	05431000
	IF SYSDISKADR=0 THEN	05432000
	BEGIN A[1]=0; GO TO EXIT END;	05433000
	IF LINE<0 AND IO THEN	05434000
	BEGIN SLEEP([TOGGLE],SYSDISKMASK);	05435000
	LOCKTOG(SYSDISKMASK);	05436000
	XI=SPACE(30);	05437000
	END;	05438000
	IF LINE GTR STATIONMAX THEN	05438100
	BEGIN A[0]=A[1]=0; GO TO OUT END;	05439000
	I+(-(T+ABS(LINE) DIV SYSDISKRPB)*SYSDISKRPB+ABS(LINE))*	05440000
	SYSDISKRL;	05441000
	T+SYSDISKADR+T;	05442000
	IF ABS(J+M[X=1])#T THEN	05443000
	BEGIN IF J<0 THEN DISKWAIT(X, 30, ABS(J));	05444000
	DISKWAIT(-X, 30, T);	05444099
\$ SET OMIT = NOT(STATISTICS)	IF J LSS 0 THEN COUNTUP(23,1); COUNTUP(23,1);	05444100
\$ POP OMIT	J+T;	05445000
	END;	05446000
	IF NOT IO THEN J=NABS(J);	05447000
	MOVE(SYSDISKRL,X+1,P(A,IF IO THEN P ELSE P(XCH)));	05448000
	M[X=1]=J;	05450000
	IF LINE.[1:1] AND IO < 2 THEN	05451000
OUT:	BEGIN IF (J+M[X=1])<0 THEN	05452000
	DISKWAIT(X, 30, -J);	05453000
\$ SET OMIT = NOT(STATISTICS)	IF J LSS 0 THEN COUNTUP(23,1);	05453099
		05453100

\$ POP OMIT

```
UNLOCKTOG(SYSDISKMASK);
FORGETSPACE(X);
X=0;
END;
EXIT:END;
PROCEDURE LOGWARN(RC); VALUE RC; REAL RC;
BEGIN REAL T,V;
DEFINE IO=LOGARRAY[31]#,
DELTA=LOGARRAY[32]#,
N=LOGARRAY[33]#,
S=LOGARRAY[34]#,
R=LOGARRAY[35]#,
H=LOGARRAY[36]#;
IF RC#0 THEN
BEGIN
STREAM(B:=RC=19,M:=5*RC,T:=T:=SPACE(10));
BEGIN DS=5 LIT "#LOG ";
SI:=LOC M; DS:=3 DEC; DI:=DI+3; DS:=2 FILL;
DI:=T; DI:=DI+8; DS:=6 LIT"% FULL";
B(DS+ 9 LIT "(AUTO LN)");
DS=LIT"+";
END;
SPOUT(T);
END;
IF RC=19 THEN LOGOUT ELSE
IF R=0 THEN % GET A ROW UNDER NO
IF RC LSS 20 THEN % USER DISK CONDITIONS.
BEGIN
STREAM(T:=T:=SPACE(10));
DS:=30 LIT"#NO USER DISK FOR NEW LOG ROW*";
SPOUT(T);
T:=GETUSERDISK(S&2[1:46:2]);
IF NOT DELTA,[1:1] THEN SLEEP([DELTA],-0);
DELTA:=-DELTA;
R:=T-N;
SLEEP([IO],IOMASK);
M[V:=SPACE(30)] := NOT 0;
DISKWAIT(V,1,T);
DISKWAIT(-V,30,H);
M[V INX RC INX IO] := T;
M[V INX 7] := (M[V INX 9] := (*P(DUP))+1)*S*3-1;
DISKWAIT(V,30,H);
DELTA:=-DELTA;
STREAM(T:=T:=SPACE(10));
DS:=9 LIT"**LOG OK*";
SPOUT(T);
END;
KILL([RC] INX NOT 1);
END;
COMMENT THE FOLLOWING DEFINES GIVE THE TYPE NUMBER FOR LOGGING;
DEFINE
SPIN = 2#,
BOJK = 4#,
EOJK = 6#,
PBEOJK = 8#,
OPENK = 10#,
CLOSEK = 12#,
HALTK = 15#,
EOJSTATS = 16#;
```

05453101  
05454000  
05455000  
05456000  
05457000  
05460000  
05461000  
05462000  
05462100  
05462200  
05462300  
05462400  
05462500  
05462600  
05463000  
05463500  
05464000  
05465000  
05466000  
05467000  
05468000  
05469000  
05470000  
05471000  
05471050  
05471100  
05471150  
05471200  
05471225  
05471250  
05471300  
05471350  
05471400  
05471450  
05471500  
05471550  
05471600  
05471650  
05471700  
05471750  
05471800  
05471850  
05471900  
05471950  
05472000  
05472050  
05472100  
05472150  
05472500  
05473000  
05499000  
05500000  
05501000  
05502000  
05503000  
05504000  
05505000  
05506000  
05507000  
05508000

```

%
%
FILESTATS =18#,
INOUTK =21#, ( SEE 04801100 )
=22
CHRGK =24#,
DISKLOGGER=26#,
DATEK =29#,
TIMEK =31#,
CNTRLCARD =32#,
HRDWREK = 35#, %TYPE 17,HARDWARE FAIL,MESSAGES
OCM = 36#, % OPERATOR COMMENT ( TYPE 18 )
PKTK =512#, % FOR LBMESS AND FILEMESS
$ SET OMIT = NOT(STATISTICS)
$ POP OMIT
PBCCARD = 40#,
QXK =99#,
PROCEDURE FORMTIME(W,T); VALUE W,T; REAL W,T;
BEGIN INTEGER S,M;
T+(T+60) DIV 60;
S+T MOD 60;
T+T DIV 60;
M+T MOD 60;
T+T DIV 60;
STREAM(T,M,S,W+[W]);
BEGIN SI+LOC T; DS+2 DEC;
2(DS+LIT ";"; DS+2 DEC);
DI+W; DS+7 FILL;
END;
END;
PROCEDURE MAKELOG(MESS,TYPE);
VALUE MESS,TYPE;
REAL MESS,TYPE;
BEGIN ARRAY A=LOGARRAY[*];
DEFINE IO=A[31]#,
DELTA=A[32]#,
N=A[33]#,
S=A[34]#,
R=A[35]#,
H=A[36]#;
REAL T;
IF (=DELTA),[1:1] THEN SLEEP([DELTA],0);
DELTA+=DELTA;
IF (IO AND IOMASK)=0 THEN SLEEP([IO],IOMASK);
MOVE(9,MESS INX 1,[A[DELTA+1]]);
A[DELTA]+XCLOCK&P1MIX[2:43:5]&(LOGLINE,[33:7]#0)[7:47:1]&
LOGLINE[8:40:8]&TYPE[16:40:8];
IF (DELTA+DELTA+10)=30 THEN
IF R#0 THEN
BEGIN T + (N+N+1) MOD S = 0;
IO+0;
$ SET OMIT = NOT(STATISTICS)
COUNTUP(24,1);
$ POP OMIT
DISKIO(IO,A,[CF]-1,31-T,R+N-1);
IF T THEN
BEGIN IF (T:=N DIV S) GEQ 20 THEN R:=0 ELSE
IF (R:=PETUSERDISK(=S,1))#0 THEN
BEGIN R:=R-N;
SLEEP([IO],IOMASK);
DISKWAIT(A INX 30,1,R+N); %MARK ROW %LOG0

```

```

05509000
05510000
05511000
05512000
05513000
05514000
05515000
05515100
%R6405517000
05517100
05517200
05518999
05519000
05519001
05520000
05599000
05607000
05608000
05609000
05610000
05611000
05612000
05613000
05614000
05615000
05616000
05617000
05618000
05619000
05620000
05621000
05622000
05623000
05624000
05625000
05626000
05627000
05628000
05629000
05630000
05631000
05631500
05632000
05634000
05636000
05637000
05638000
05638500
05639000
05639500
05639899
05639900
05639901
05640000
05641000
05641500
05642000
05642500
05643000
%LOG05643500

```

	DISKWAIT(=A,[CF],30,H);	05644000
	A[T+10] := R+N;	05646000
	A[7] := (A[9] := (*P(DUP))+1)*S*3-1;	%LOG 05646500
	DISKWAIT(A,[CF],30,H);	05648000
	END;	05648500
	FORK(PC,LOGWARN),T,-1,128,1);	05649000
	END;	05650000
	DELTA+0;	05651000
	END ELSE DELTA+20;	05652000
	DELTA+DELTA;	05653000
	END;	05654000
PROCEDURE	STARTCANDY(ESED,PRIORITY);	05655000
	REAL PRIORITY,ESED;	05656000
	BEGIN REAL MIX=PIMIX,T;	05657000
	REAL I;	05657100
	LABEL NOFILE,EXIT;	05657200
ARRAY	A[*], B[*];	05658000
	IF CANDYINX#0 THEN ESED+1 ELSE	05659000
	BEGIN	05660000
	A := IOQUE&GETAREA(0)[CTC];	05660050
	IF BASEDISKADR=0 THEN	05660100
	BEGIN I := "TANK "	05660110
\$ SET OMIT =	NOT(SHAREDISK)	05660119
	&(SYSNO+17)[30;42;6]	05660120
\$ POP OMIT		05660121
	; GO TO NOFILE;	05660130
	END;	05660140
	IF (T:=DIRECTORYSEARCH(I:="MESSAGE ", "CANDE " ,3)) GEQ 64	05660150
	THEN BEGIN A[2] :=	05660200
	(M[T+10]);	05660240
	FORGETSPACE(T);	05660260
	END ELSE GO TO NOFILE;	05660280
\$ SET OMIT =	NOT(SHAREDISK)	05660282
	A[2] := (*P(DUP))&SYSNO[2;46;2];	05660284
\$ POP OMIT		05660285
	IF (T:=DIRECTORYSEARCH(I:="USERS ", "CANDE " ,3)) GEQ 64	05660300
	THEN BEGIN	05660400
	A[2] := A[2]&T[5;18;15]; % DSK ADRS OF HEADER (SM)	05660410
	DISKWAIT(=T,[CF],30,M[T INX 10]); % SEGMENT ZERO (SM)	05660420
	A[4] := M[T INX 2]; % ACCESS DATE (SM)	05660430
	FORGETSPACE(T); % (SM)	05660440
	END ELSE BEGIN % (SM)	05660450
NOFILE:	STREAM(I,T:=T:=SPACE(5));	05660500
	BEGIN DS := LIT"#";	05660550
	SI := LOC I; SI := SI+1; DS := 7 CHR;	05660600
	DS := 17 LIT"FILE NOT ON DISK-";	05660700
	END;	05660800
	FORGETAREA(0,A,[CF]);	05660850
	SPOUT(T); ESED := 1; GO TO EXIT;	05660900
	END;	05660950
	PRIORITY+0;	05661000
	EVENT[MIX]+0&[EVENT[MIX]][CTF];	05662000
	A[0] := (*P(DUP))&5[18;41;7];	05664000
	A[1] := BASEDISKADR;	05665000
	IF (T:=DIRECTORYSEARCH("TANK "	05665010
\$ SET OMIT =	NOT(SHAREDISK)	05665014
	&(SYSNO+17)[30;42;6]	05665015
\$ POP OMIT		05665016
	, "DISK " ,4)) GEQ 64 THEN A[3] := T,[FF]	05665020
	ELSE	05665025

STARTCANDY

```

BEGIN B := IOQUE&(T := SPACE(30))[CTC];
MOVE(30, T=1, T); B[0] := @0003600036000101;
STREAM(DATE, T:=T+3);
BEGIN SI:=LOC DATE; DS:=8 OCT;
DI:=T; DS:=2 LIT "01";
END STREAMING;
B[4] := (O&1[9:47:1]&SYSNO[4:46:2]) OR MEMORY;
B[2] := MCP; B[9] := 20;
B[8] := TANKCHUNKSIZE; B[10] := BASEDISKADR;
A[3] := "EUF("TANK "
$ SET OMIT = NOT(SHAREDISK)
$ POP OMIT
      ,"DISK ", T=1);
END BUILD NEW TANK FILE;
DISKWAIT(=T, =30, 0); % PUT ADDRESS OF BYPASS IN
M[T+4] := DISKBOTTOM+2; % SEGMENT 0 WHERE CANDE CAN
DISKWAIT(T, =30, 0); % FIND IT
FORGETSPACE(T);
QUEVENT(A, [CF], MIX);
A+A&SPACE(SYSDISKRL)[CTC];
FOR I:=0 STEP 1 UNTIL STATIONMAX+1 DO
BEGIN SYSDISKIO(1, I, A);
IF A[0].DIALEDUP THEN
IF SCH(A) THEN
BEGIN STABLE[I], DIALEDUP:=0;
SEQARRAY[I], [FF]:=1;
FORK(P(, SCHEDIDLE), I, 0, 160, 0)
END ELSE
IF A[1] NEQ MCP THEN % TELL CANDE, % %DS
BEGIN T+GETAREA(0);
M[T]+O&9[18:41:7]&I[25:40:8];
M[T+1]:=A[1]; M[T+2]:=STABLE[I], STATIONTYPE;
IF A[1]=0 THEN M[T], [18:7]:=1; % %DS
QUEVENT(T, MIX);
END;
END;
FORGETSPACE(A);
IF EVENT[0], [CF]#0 THEN
BEGIN M[EVENT[MIX], [FF]], [CF] + EVENT[0];
EVENT[MIX], [FF]+EVENT[0], [FF];
EVENT[0]+O&[EVENT[0]][CTF];
END;
DAT[CANDYINX:=MIX], NDSABLE := 1;
FOR I:=1 STEP 1 UNTIL STATIONMAX DO
IF STABLE[I], CANDEFLAG THEN
IF STABLE[I], MIXNR=0 THEN STABLE[I], MIXNR:=MIX;
END;
EXIT;
END;
PROCEDURE STOPCANDY;
BEGIN REAL MIX=PIMIX, T;
;STREAM(T + T + SPACE(10));
DS + 40 LIT "**CANDE ERROR == PLEASE TAKE DUMP, ETC, +";
SPOUT(T); HALT; COMPLEXSLEEP(=100=NUMESS);
DO UNTIL KEYIN(0); NOPROCESSTOG + NOPROCESSTOG=1;
FOR T:=1 STEP 1 UNTIL STATIONMAX DO
IF STABLE[T], CANDEFLAG THEN
IF STABLE[T], MIXNR=MIX THEN STABLE[T], MIXNR:=0;
CANDYINX+LOGLINE+0;

```

STOPCANDY

IF EVENT[MIX],[CF]#0 THEN	05706000
EVENT[0]=EVENT[MIX];	05707000
IF (T:=DIRECTORYSEARCH("TANK "	05710000
\$ SET OMIT = NOT(SHAREDISK)	05710099
&(SYSNO+17)[30:42:6]	05710100
\$ POP OMIT	05710101
,"DISK ",14))#0 THEN FORGETSPACE(T);	05711000
IF (T := DIRECTORYSEARCH("MESSAGE ", "CANDE ", 13))#0	05712000
THEN FORGETSPACE(T);	05713000
IF (T := DIRECTORYSEARCH("USERS ", "CANDE ", 13))#0	05714000
THEN FORGETSPACE(T);	05715000
END;	05719000
PROCEDURE SPREADTHEWORD; %UPDATES ADINFO & SENDS H/L MESSAGE TO	05720000
BEGIN %LINES MARKED DIALUP	05721000
LABEL NODISK, SPOUTERR, NOSYSDISK;	05721100
REAL I, J, K, L, N, B, T, GARBAGE;	05722000
ARRAY A[*];	05723000
\$ SET OMIT = TWXONLY	05723049
SUBROUTINE SETNAOG; SEQARRAY[I],[26:6]:=N:=N+1;	05723050
\$ POP OMIT	05723051
B:=SPACE(10);	05723100
N:=1;	05723150
IF SYSDISKADR=0 THEN	05723200
BEGIN STREAM(S:=SYSNO+17,B);	05723300
BEGIN DS:=18 LIT"*=NO SYSTEM DISK*";	05723310
\$ SET OMIT = NOT(SHAREDISK)	05723319
SI:=LOC S; SI:=SI+7; DI:=DI-7; DS:=CHR;	05723320
\$ POP OMIT	05723321
END;	05723350
GO TO SPOUTERR;	05723380
END;	05723400
IF NOT P(RRR),[17:1] THEN	05723420
BEGIN; STREAM(B);	05723450
DS + 16 LIT"*=DTC NOT READY*";	05723500
GO TO SPOUTERR;	05723520
END;	05723540
IF PROGTANK=0 THEN	05723560
IF (PROGTANK:=GETUSERDISK((STATIONMAX+1)*CLUMPSIZE OR	05723580
MEMORY)) = 0 THEN GO TO NODISK;	05723600
IF TANKADDRESS=0 THEN	05723620
IF (TANKADDRESS:=GETUSERDISK((STATIONMAX+1)*GLOMSIZE OR	05723640
MEMORY)) = 0 THEN GO TO NODISK;	05723660
IF BASEDISKADR=0 THEN	05723680
IF (BASEDISKADR:=GETUSERDISK(=TANKCHUNKSIZE OR MEMORY))=0 THEN	05723700
BEGIN	05723720
NODISK; STREAM(B);	05723740
DS:=33 LIT"*=NO USER DISK FOR DATACOM TANKS*";	05723760
SPOUTERR; SPOUT(B);	05723800
GO TO NOSYSDISK;	05723900
END;	05723950
\$ SET OMIT = NOT SEPTICTANK	05723959
COMMENT REMOVE THIS CARD TO GET SEPTIC AT EACH INITIALIZATION	05723960
RUNSEPTIC(0);	05723970
\$ POP OMIT	05723971
IOREQUEST(0, INTERROGATEMASK, 0&30[12:42:6]);	05723990
A := [M[SPACE(SYSDISKRL)]] & SYSDISKRL[8:38:10];	05724000
SYSDISKIO(1,0,A);	05724100
SPOWORD:=A[3];	05724400
STREAM(B);	05725000
BEGIN	05726000

DS:=19 LIT "##P##L##0##P##";	05726100
DS:=28 LIT "RESTARTING...PLEASE WAIT##";	05726200
END;	05726300
FOR I:=1 STEP 1 UNTIL STATIONMAX DO	05727000
BEGIN	05728000
SYSDISKIO(1,I,A);	05729000
IF SCH(A) THEN	05729010
BEGIN STABLE[T:=I]:=0&1[4:47:1]&1[16:47:1];	05729020
LINETABLE[I]:=A[4]&1[2:47:1];	05729025
IF A[0],DIALEDUP THEN	05729030
IF A[2],[6:18] NEQ "FIL" THEN	05729040
BEGIN A[0],DIALEDUP:=0; SYSDISKIO(0,I,A) END;	05729050
END	05729060
ELSE BEGIN	05729070
TANKS[I],[1:1]:=1;	05731100
INPUTANK[I]:=010000;	05731200
STABLE[I]:=0&A[0][10:10:3]&3[15:46:2]&A[0][25:2:8]	05731300
&(CANDEMIX[I]+32)[4:42:6];	05731400
IF I LEQ LMAX THEN       % LOCK IT TO GARBAGE	05731500
LINETABLE[I]:=NABS(A[4])&IDL[21:43:5]&[GARBAGE][CTC];	05731550
\$ SET OMIT = TWXONLY	05731599
IF (J:=STABLE[I],STATIONTYPE)≠TWX THEN	05731600
BEGIN	05731700
SEQARRAY[I]:=0&A[0][12:34:14]&A[0][6:17:3]	05731800
&I[40:40:8];	05731850
IF J THEN       % SCREEN DEVICE	05731900
BEGIN SETNAOG;	05732000
TNAOG[N]:=IF I=ABS(SPOWORD) THEN 0	05732100
ELSE 0&A[0][14:20:14];	05732150
END ELSE	05732200
IF J=TC500 THEN SETNAOG;	05732300
IF I LEQ LMAX THEN	05732400
IF LINEDISC[I]=MULTIPOINT THEN	05732450
SEQARRAY[I],LINELINK:=I ELSE ELSE	05732500
IF (J:=A[0],[2:8])≠0 THEN	05732600
BEGIN L:=J;	05732700
DO J:=SEQARRAY[K:=J],LINELINK UNTIL J=L;	05732800
SEQARRAY[K],LINELINK:=I;	05732900
SEQARRAY[I],LINELINK:=L;	05733000
END END;	05733100
\$ POP OMIT	05733101
A[0],DIALEDUP:=1;	05733200
A[2]=0;	05734100
SYSDISKIO(0,I,A);	05734200
END;	05734900
END;	05735000
SYSDISKIO(1,STATIONMAX+1,A);	05736000
IF UNIT[30],[16:2] ≠ 0 THEN	05736200
COMPLEXSLEEP(UNIT[30],[16:2]=0);	05736400
REMOTE ← 1;	05737000
\$ SET OMIT = NOT(SAVERESULTS OR DEBUGGING)	05737099
STORAWAY:=NOT FALSE;	05737100
\$ POP OMIT	05737101
FOR I:=T+1 STEP 1 UNTIL STATIONMAX DO	05738000
BEGIN	05740000
\$ SET OMIT = TWXONLY	05740099
IF I LEQ LMAX THEN	05740100
\$ POP OMIT	05740101
IF BLASTREAD(I,1) THEN	05742000
TWXOUT(B,60,1&3[1:46:2],I);	05743000



```

END;
FORGETSPACE(A);
FORGETSPACE(B);
NOSYSDISK;
END SPREADTHEWORD;
PROCEDURE COMM17;
BEGIN REAL T;
REAL L=-7,U=-6,C=-5;
STREAM(L+L,[40:8],C,U,T+T+SPACE(10));
BEGIN DS:=5 LIT"SHORT"; C(DI:=DI-5); DS:=4 LIT"LONG";
DS:=14 LIT" CARRIAGE FOR ";
SI*LOC U; SI*SI+1; DS*7 CHR;
DS*4 LIT " ON ";
SI*LOC L; DS*2 DEC;
DS:=LIT LEFTARROW;
END STREAM;
LONGCARRIAGE[L]:=C;
LOGLINE * L OR 512;
SPOUTIT(T,CHRGK+1);
GO TO RETURN;
END;
BOOLEAN PROCEDURE BLASTREAD(LINE,C);
VALUE LINE,C;
REAL LINE,C;
BEGIN
REAL R,T; LABEL EX;
DEFINE BLAM = C,[47:1]#; % CLEAR THE LINE, LEAVING IT IDLE,
LOCK = C,[46:1]#; % THE LINE NEEDS TO BE LOCKED,
IDLE = C,[45:1]#; % WAIT FOR AN IDLE STATUS,
CLEAR = C,[44:1]#; % DISCONNECT THE LINE,
IF LINE>LMAX THEN P(O,RTN);
T:=SPACE(15);
IF BLAM THEN
BEGIN IF NOT LOCK THEN % ALREADY LOCKED, SET ADDRESS
LINETABLE[LINE]:=(+P(DUP))&[R][CTC];
$ SET OMIT = TWXONLY
IF STABLE[LINE],STATIONTYPE=TWX THEN
$ POP OMIT
BEGIN STREAM(T:=T+1); DS:=3 LIT"≤+";
M[T]:=0;
DCWAIT(T,LINE,R,C,60); % NOTE THAT C,[47:1] IS SET,
IF R = 0 THEN %NO RESPONSE = DISCONNECT
BEGIN FORK(P(,QUITTER),LINE&LINE[CTF],-2,96,1);
IF CANDYINX NEQ 0 THEN
BEGIN M[R:=GETAREA(0)],[FF]:=R&6[33:41:7];
QUEVENT(R,CANDYINX);
END;
LINETABLE[LINE]:=ABS(+P(DUP))&DISCON[21:43:5]
&0[CTC]; %UNLOCK LINE
STABLE[LINE],[16:1]:=0; %SET DISCONNECTING
GO TO EX;
END;
$ SET OMIT = TWXONLY
END ELSE
BEGIN M[T]:=0&1[5:47:1];
DCWAIT(T,LINE,R,C AND @76,0);
IF R,[CF]≠0 AND R,[26:17]≠0 THEN
BEGIN R:=0;
CLICK:=CLOCK+P(RTR)+120;
SLEEP([R],0);

```

```

05744000
05745000
05746000
05746500
05747000
05750000
05750500
05751000
05751500
05752000
05752100
05752500
05753000
05753500
05754000
05754500
05755000
05756000
05756500
05757000
05757500
05758000
05758500
05759000
05759500
05760000
05761000
05761100
05761200
05761300
05761900
05762000
05762500
05763000
05763100
05763499
05763500
05763501
05764000
05764500
05764600
05764700
05764800
05764850
05764860
05764870
05764880
05764900
05765000
05765100
05765200
05765300
05765499
05765500
05765550
05765600
05765620
05765640
05765660
05765680

```

```

END;
IF R.[27:1] THEN % WRITE READY
BEGIN M[T]:=0;
M[T+1]:=01274123700000000; % "###"
DCWAIT(T,LINE,R,1,180);
END;
$ POP OMIT
END;
M[T]:=0&1[6:47:1];
DCWAIT(T,LINE,R,0,0);
$ SET OMIT = TWXONLY
DCWAIT(T,LINE,R,IF LINEDISC[LINE]=CONTENTION
THEN 0 ELSE =0,0);
IF LINEDISC[LINE]=CONTENTION THEN
BEGIN STREAM(T:=T+1); DS:=2 LIT "$";
M[T]:=0;
DCWAIT(T,LINE,R,=1,60);
END;
$ POP OMIT
$ SET OMIT = NOT(TWXONLY)
DCWAIT(T,LINE,R,=0,0);
$ POP OMIT
IF R=0 OR (R.[27:2]#0 AND NOT R.[30:1]) THEN GO TO EX;
END;
IF CLEAR OR (R.[30:1] AND SWAPEND#0) THEN
BEGIN STREAM(T);
DS:=29 LIT"##PLEASE CALL BACK LATER##$";
TWXOUT(T,29,=0,LINE);
END;
BLASTREAD:=TRUE;
EX: FORGETSPACE(T);
END BLASTREAD;
DEFINE
MAXSIZ=[1:20]#, TOMAXSIZ=1:28:20#,
SPEED = [23:3]#, TOSPEED= 23:45:3#,
EUNP = [21:1]#, TOEUNP = 21:47:1#,
STARTWRD=[26:12]#, TOSTARTWRD=26:36:12#,
NUMENT=[38:10]#, TONUMENT=38:38:10#, NUMENTM=1023#,
DSIZE=[2:20]#, TODSIZE=2:28:20#,
DEND=[22:26]#, TODEND=22:22:26#,
TOSIZE=8:38:10#, NEUF=[18:15]#,
EUIOFFSET=4 #, % ONE WORD FOR EACH I/O CHANNEL,
AVDIFFMIN=15#, AVDIFFMAX=50#, % AVDIFFMAX GTR AVDIFFMIN GTR 14,
AVTMAX=3900#, % MAX # WORDS ALLOWED FOR AVAILABLE TABLE ON DISK,
% IS REFLECTED IN USERDISKBOTTOM & DISKAVAILTABLEMAX
AVSMIN=90 #, AVSMAX=300#, % MIN AND MAX # WORDS TO READ IN @ 1 TIM
% AVSMAX GTR AVSMIN GTR 85,
% BOTH MUST BE MULTIPLES OF 30,
FIXARRAY(FIXARRAY1,FIXARRAY2,FIXARRAY3)=FIXARRAY1+[M[FIXARRAY2+
SPACE(FIXARRAY3)]]&FIXARRAY3[TOSIZE]#;
$ SET OMIT = NOT(SHAREDISK )
% START OF SHAREDISK DISK FILE MAINTAINANCE DECLATATIONS *****
%
DEFINE SLINK = [1:17] #, % = SCRATCH LINK, SCRATCHVECTOR[I],SLINK
TOSLINK = 1:31:17 #, % IS DISK ADDRESS OF I-TH SCRATCHSEG,
% AND SCRATCHSEG[0],SLINK OF THE I-TH
% SCRATCHSEG IS DISK ADDRESS OF I+1-TH
% SCRATCHSEG.
SNUM = [18:15] #, % = SCRATCH NUMBER, SCRATCHVECTOR[I],
TOSNUM = 18:43:15 #, % SNUM IS NUMBER OF ENTRIES IN I-TH

```

```

05765700
05765710
05765720
05765730
05765740
05765750
05765771
05765800
05765900
05766000
05766099
05766100
05766200
05766300
05766400
05766500
05766600
05766700
05766701
05766799
05766800
05766801
05766900
05767000
05767500
05768000
05768500
05769000
05769500
05769750
05770000
05770500
05780000
05780019
05780020
05780025
05780030
05780040
05780100
05780200
05780300
05780310
05780400
05780500
05780505
05780600
05780605
05780610
05780700
05780800
05800000
05800200
05800300
05800400
05800500
05800600
05800700
05800800
05800900
05801000

```

```

% SCRATCHSEG, AND SCRATCHSEG[0],SNUM IS 05801100
% NUMBER OF ENTRIES IN SCRATCHSEG, 05801200
SRADDR = [23:25] #, % = SCRATCH ROW ADDRESS, SCRATCHSEG[J], 05801300
TOSRADDR = 23:23:25 #, % SRADDR IS DISK ADDRESS OF A SCRATCHROW 05801400
SHADDR = [23:25] #, % = SCRATCH HIGH ADDRESS, SCRATCHVECTOR 05801500
TOSHADDR = 23:23:25 #, % [J],SHADDR IS 1+MAX DISK ADDRESS OF 05801600
% THE DISK ADDRESSES IN THE ENTRIES IN 05801700
% THE I-TH SCRATCHSEG, I.E. SCRATCHVECT=05801800
% OR[I],SHADDR=MAX(SCRATCHSEG[1],SRADDR+05801900
% SCRATCHSEG[1],LENGTH,...,SCRATCHSEG[N]05802000
% J,SRADDR+SCRATCHSEG[I],LENGTH), 05802100
LENGTH= [3:20] #, % = SCRATCH LENGTH, SCRATCHSEG[J], 05802200
TOSLENGTH= 3:28:20 #, % LENGTH IS # SEGMENTS IN THE SCRATCH 05802300
% ROW ADDRESSED BY SCRATCHSEG[J],SRADDR,05802400
SSIZE = [18:10] #, % = SCRATCHVECTOR SIZE, SCRATCHVECTOR[0]05802500
TOSSIZE = 18:38:10 #, % ,SSIZE IS THE NUMBER OF SCRATCH VECTOR05802600
% DOPE ENTRIES, 05802700
MIXFL = [9:6] #, % MIX-INDEX FIELD OF 1-ST LINK (IN-USE),05803300
SUPERSCRATCHMAX = 510 #, % MAX SIZE-1 ALLOWED FOR SCRATCHVECTOR, 05803400
SCRATCHMIN = 19 #, % = LOWER BOUND FOR DELETION CONSOLIDATE05803500
SCRATCHMAX = 11 #, % = UPPER BOUND FOR ADDITION CONSOLIDATE05803600
SCRATCHDOWN = 14 #, % THESE THREE DEFINES ARE USED IN DET= 05803700
SCRATCHOFF = 9 #, % ERMINING WHEN TO ADJUST THE SCRATCH= 05803800
SCRATCHUP = 5 #, % VECs SIZE TO REFLECT ITS ACTUAL NUMBER05803900
% OF ENTRIES, NOTE THAT SCRATCHDOWN MUST05804000
% BE GREATER THAN SCRATCHOFF, ALL ARE >005804100
SCRATCHTYPE = 13 #, % DIALED INTO TYPE-FIELD OF SCRATCHVEC, 05804200
SCRATCHSAVE = 1 #, % SAVE FACTOR OF SCRATCHVEC, 05804300
SERROR(SERROR1) = BEGIN N+SERROR1; GO ERROR END #; 05805500
% * END OF DEFINES * * * 05805900
% SEE START OF MCP FOR SCRATCHDIRECTORYREADY AND SCRATCHDIRECTORYMASK *,05806000
ARRAY SCRATCHVEC[*]; 05806200
05806300
05806400
05806500
PROCEDURE SCRATCHSORT(A,N) ; 05806600
VALUE N ; 05806700
REAL N ; 05806800
ARRAY A[*] ; 05806900
BEGIN % SCRATCHSORT BINARILY SORTS N ELEMENTS OF A--A[1], 05807000
% A[2],...A[N]---IN DESCENDING ORDER 05807100
% SORT TIME APPROX N*LN(N)*95 MICROSECONDS, 05807200
% THE KEY IS ,[23:25], WHICH IS EITHER ,SHADDR OR ,SRADDR,05807400
INTEGER D ; 05807500
REAL I,J,Y ; 05807600
LABEL C1,C2,C3,XT ; 05807700
% ** ** ** ** 05807800
IF (D+1)<N THEN WHILE (D+D+D)<N DO; 05808000
C1: 05808100
IF (D=D DIV 2) LEQ 0 THEN GO XT ELSE I:=1 ; 05808200
C2: 05808300
Y=A[(J+I)+D] ; 05808400
C3: 05808500
IF Y,SHADDR>A[J],SHADDR THEN 05808600
BEGIN 05808700
A[J+D]+A[J] ; 05808800
IF (J+J=D) > 0 THEN GO TO C3 ; 05808900
END ; 05809000
A[J+D]+Y ; 05809100
IF (I+I+1)+D ≤ N THEN GO TO C2 ELSE GO TO C1 ; 05809200
XT;

```

```

END OF SCRATCHSORT ;
05809300

$ SET OMIT = NOT(DEBUGGING) OR OMIT
05809400
PROCEDURE SCRATCHDIRECTORYERROR(A,N) ;
05809500
VALUE A,N ;
05809599
REAL A,N ;
05809600
BEGIN
05809700
% ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
05809800
STREAM(N:=N,A:=A);
05809900
BEGIN
05810000
DS=29 LIT" MCP SCRATCH DIRECTORY ERROR " ;
05810100
SI=LOC N; DS=3 DEC; DS=LIT"+ " ;
05810200
END ;
05810300
SPOUT(A) ;
05810400
COMPLEXSLEEP(-100=NUMESS) ;
05810500
DDT;
05810600
DO UNTIL FALSE ;
05810700
END OF SCRATCHDIRECTORYERROR ;
05810900
05811000
05811100
05811150
05811160
05811170
05811180
05811190
05811200
05811210
05811220
05811230
05811240
05811250
05811260
05811270
05811280
05811281
05811340
05811355
05811360
05811370
05811385
05811400
05811410
05811414
05811415
05811416
05811430
05811445
05811460
05811475
05811490
05811505
05811520
05811535
05811540
05811549
05811550
05811565
05811580
05811595
05811596
05811805

BOOLEAN PROCEDURE SCRATCHCHECK(I,H,S) ;
VALUE I,S ;
REAL I,H,S ;
BEGIN
% ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **^
SCRATCHCHECK=TRUE ;
IF S,SLINK=(IF I#H THEN SCRATCHVEC[I+1] ELSE 0),SLINK THEN H+1
ELSE IF SCRATCHVEC[I],SNUM#I+S,SNUM THEN H+2
ELSE IF I>29 THEN H+3
ELSE IF I<=0 THEN H+4
ELSE SCRATCHCHECK=FALSE ;
END OF SCRATCHCHECK ;
$ POP OMIT
PROCEDURE SCRATCHSPECIALCASE(CN,A,N,CORADDR,SEGADDR,I,H,SCRATCHSEG);
VALUE CN,A,N,SEGADDR,I,H;
REAL CN,A,N,CORADDR,SEGADDR,I,H;
ARRAY SCRATCHSEG[*] ;
BEGIN
REAL POT,MAX,L,Q,J,E,F ;
ARRAY BOTH[*];
$ SET OMIT = NOT(DEBUGGING) OR OMIT
LABEL START,ERROR ;
$ POP OMIT
SUBROUTINE GETNEWSCRATCHVEC ;
BEGIN
P(PIMIX); PIMIX+0; L+GETSPACE(A,SCRATCHTYPE,SCRATCHSAVE) ;
PIMIX+P; M[L],MIXFL+0 ;
MOVE(H+1,SCRATCHVEC,LI=L+2) ;
FORGETSPACE(SCRATCHVEC) ;
SCRATCHVEC:=SCRATCHVEC & L[CTC] & A[TOSIZE] ;
END OF GETNEWSCRATCHVEC ;
% ** ** ** **^
$ SET OMIT = NOT(DEBUGGING) OR OMIT
GO START ;
ERROR:
SCRATCHDIRECTORYERROR(CORADDR,N+100) ;
START:
$ POP OMIT
IF CN=1 THEN

```

```

BEGIN
IF (IF I#1 THEN (L+SCRATCHVEC[I-1],SNUM)<SCRATCHMAX ELSE
L+32) OR (IF I#H THEN (Q+SCRATCHVEC[I+1],SNUM)<SCRATCHMAX
ELSE Q+32) THEN % THEN INSTEAD OF GETTING A NEW SEG AND
BEGIN % GIVING THE I-TH AND NEW SEG EACH 15 ENTRIES,
IF L>Q THEN % WE CAN SPLIT BETWEEN THE I-TH AND Q-TH
BEGIN % SEGS THE SUM OF THE ONE NEW ENTRY AND
L+Q ; % THEIR INDIVIDUAL ENTRIES, WE CAN DO
Q+1+1 ;% THIS BECAUSE THE Q-TH SEG HAS ONLY
END % L<SCRATCHMAX ENTRIES, THIS OPERATION IS
ELSE Q+1-1 ;% DONE TO HELP REDUCE THE NUMBER OF
POT+L+31 ; % SPARSE SEGMENTS.
FIXARRAY(BOTH,J,POT) ;
DISKWAIT(-J,30,MAX+SCRATCHVEC[Q],SLINK) ;
$ SET OMIT = NOT(DEBUGGING) OR OMIT
IF SCRATCHCHECK(Q,H,BOTH[0]) THEN SEKROR(H+21) ;
$ POP OMIT

MOVE(29,[SCRATCHSEG[1]],[BOTH[L+1]]) ;
BOTH[POT:=POT-1]:=A & N[TOSLENGTH] ;
SCRATCHSORT(BOTH,POT) ;
MOVE(H+POT-POT+POT DIV 2,[BOTH[POT+1]],
[SCRATCHSEG[1]]) ;
A:=MAX; N:=SEGADDR ;
IF Q LSS I THEN
BEGIN
POLISH(SCRATCHSEG[0],BOTH[0],
[SCRATCHSEG[0]],:=,[BOTH[0]],:=) ;
POLISH(J,CORADDR,J,+,[CORADDR],+);
POLISH(I,Q,I,:=,Q,:=) ;
POLISH(SEGADDR,MAX,SEGADDR,:=,MAX,:=) ;
END ;
BOTH[0],SNUM:=POT ;
SCRATCHSEG[0],SNUM:=H ;
DISKWAIT(J,30,A) ;
DISKWAIT(CORADDR,30,N) ;
SCRATCHVEC[I]:=P(SCRATCHSEG[1],DUP),SRADDR
+P(XCH),SLENGTH) & H[TOSNUM] & SEGADDR
[TOSLINK] ;
SCRATCHVEC[Q]:=P(BOTH[1],DUP),SRADDR+P(XCH)
,SLENGTH) & POT[TOSNUM] & MAX[TOSLINK] ;
FORGETSPACE(J) ;
END
$ SET OMIT = NOT(DEBUGGING) OR OMIT
ELSE IF H GTR 254 THEN SERROR(49)
$ POP OMIT

ELSE % ELSE GET A NEW SEGMENT AND SPLIT THE I-TH
BEGIN % SEGMENTS 29 ENTRIES PLUS THE NEW ENTRY
% BETWEEN THE NEW AND I-TH SEGMENT.
SCRATCHSEG[30]:=A & N[TOSLENGTH] ;
SCRATCHSORT(SCRATCHSEG,30) ;
IF SCRATCHVEC,SIZE<H+2 THEN % THEN WE MUST GET A
BEGIN % BIGGER SCRATCHVEC,
A:=H+2+SCRATCHUP ;
GETNEWSCRATCHVEC ;
END ;
POT:=I+1 ;
FOR J:=H STEP -1 UNTIL POT
DO SCRATCHVEC[J+1]:=SCRATCHVEC[J] ;
SCRATCHVEC[I+1]:=P(SCRATCHSEG[1],DUP),SRADDR
+P(XCH),SLENGTH)

```

05811820  
05811835  
05811850  
05811865  
05811880  
05811895  
05811910  
05811925  
05811940  
05811955  
05811970  
05811985  
05812000  
05812015  
05812016  
05812030  
05812031  
05812045  
05812060  
05812075  
05812090  
05812105  
05812120  
05812135  
05812150  
05812165  
05812180  
05812195  
05812210  
05812225  
05812240  
05812255  
05812270  
05812285  
05812300  
05812315  
05812330  
05812345  
05812360  
05812375  
05812390  
05812405  
05812406  
05812420  
05812421  
05812435  
05812450  
05812465  
05812480  
05812495  
05812510  
05812525  
05812540  
05812555  
05812630  
05812645  
05812660  
05812675  
05812690  
05812705

```

& 15[TOSNUM] 05812720
& (L+GETESPDISK)[TOSLINK]; 05812735
SCRATCHSEG[0],SNUM:=15 ; 05812750
DISKWAIT(CORADDR,30,L) ; 05812765
SCRATCHVEC[I]:=(P(SCRATCHSEG[16],DUP) 05812780
,SRADDR+P(XCH),SLENGTH) 05812795
& 15[TOSNUM] & SEGADDR[TOSLINK] ; 05812810
SCRATCHSEG[15]:=0 & L[TOSLINK] & 15[TOSNUM] ; 05812825
DISKWAIT(CORADDR+15,16,SEGADDR) ; 05812840
SCRATCHVEC[0],SSIZE+H+1 ; 05812855
END ; 05812870
END 05812885
ELSE 05812900
BEGIN 05812915
IF (IF I#1 THEN (L+SCRATCHVEC[I-1],SNUM)>SCRATCHMAX ELSE 05812930
L+FALSE) OR (IF I#H THEN (Q+SCRATCHVEC[I+1],SNUM)>SCRATCHMAX 05812945
ELSE Q+FALSE) THEN %THEN, INSTEAD OF FORGETTING SEGMENT, SPLIT 05812960
BEGIN % CONTENTS OF Q-TH AND I-TH SEGMENT BETWEEN THEM= 05812975
IF POT:=Q>L THEN BEGIN L+Q; Q+I+1 END ELSE Q+I-1 ; % SELVE 05812990
A+SCRATCHSEG[0] ; 05813005
DISKWAIT(-CORADDR,30,E+SCRATCHVEC[Q],SLINK) ; 05813020
$ SET OMIT = NOT(DEBUGGING) OR OMIT 05813034
IF SCRATCHCHECK(Q,H,HI=SCRATCHSEG[0]) THEN SERROR(12+H) ; 05813035
$ POP OMIT 05813036
SCRATCHSORT(SCRATCHSEG,L) ; 05813050
F:=L-N:=L DIV 2 ; 05813065
SCRATCHVEC[I]:=(P(SCRATCHSEG[(L:=IF POT THEN N ELSE 0) 05813080
+ 1],DUP),SRADDR+P(XCH),SLENGTH) 05813095
& FITOSNUM] & SEGADDR[TOSLINK] ; 05813110
SCRATCHVEC[Q]:=(P(SCRATCHSEG[(POT:=IF POT THEN 0 ELSE F) 05813125
+1],DUP),SRADDR+P(XCH),SLENGTH) 05813140
& N[TOSNUM] & E[TOSLINK] ; 05813155
POLISH(SCRATCHSEG[N],SCRATCHSEG[N-1]) ; 05813170
SCRATCHSEG[L]:=A & F[TOSNUM] ; 05813185
DISKWAIT(CORADDR+L,F+1,SEGADDR) ; 05813200
POLISH([SCRATCHSEG[N-1]],+,[SCRATCHSEG[N]],+) ; 05813215
SCRATCHSEG[POT]:=H & N[TOSNUM] ; 05813245
DISKWAIT(CORADDR+POT,N+1,E) ; 05813260
END 05813275
ELSE % ELSE THERE IS NO SEG ADJ TO I-TH SEG WITH ENOUGH 05813290
BEGIN % ENTRIES TO SPLIT, SO WE DELETE THE I-TH SEGMENT, 05813305
E+SCRATCHSEG[0],SLINK ; 05813320
DISKWAIT(-CORADDR,POT+IF F+I=1 THEN -30 ELSE 30,Q+ 05813335
SCRATCHVEC[I-1],SLINK) ; 05813350
$ SET OMIT = NOT(DEBUGGING) OR OMIT 05813364
IF F AND SCRATCHSEG[SYSNO],SLINK#SEGADDR THEN SERROR(4) ; 05813365
$ POP OMIT 05813366
SCRATCHSEG[IF F THEN SYSNO ELSE 0],SLINK+E ; 05813380
DISKWAIT(CORADDR,POT,Q) ; 05813395
FORGETESPDISK(SEGADDR) ; 05813410
IF I#H THEN MOVE(H=I,[SCRATCHVEC[I+1]],[SCRATCHVEC[I]]) ; 05813425
SCRATCHVEC[0],SSIZE:=H:=H-1 ; 05813440
IF H<(L:=SCRATCHVEC,SIZE)=SCRATCHDOWN THEN %THEN GET A NEW 05813455
BEGIN % SCRATCHVEC CLOSER IN SIZE TO ACTUAL # ENTRIES 05813470
A:=L-SCRATCHOFF ; 05813485
GETNEWSCRATCHVEC ; 05813500
END ; 05813560
END ; 05813575
END ; 05813590
END OF SCRATCHSPECIALCASE ; 05813605

```

```

PROCEDURE SCRATCHDIRECTORYENTER(A,N);
VALUE A,N;
REAL A,N;
BEGIN
    INTEGER I;
    REAL CORADDR,H,L,Q,SEGADDR=Q;
    ARRAY SCRATCHSEG[*];
    LABEL FOUND, SEARCH;
$ SET OMIT = NOT(DEBUGGING) OR OMIT
    LABEL ERROR;
$ POP OMIT
    % ** ** ** **
    FIXARRAY(SCRATCHSEG,CORADDR,31);
    SLEEP([TOGGLE],SCRATCHDIRECTORYMASK); LOCKTOG(SCRATCHDIRECTORYMASK);
    Q:=H:=SCRATCHVEC[0],SSIZE;
    IF SCRATCHVEC[I:=1],SHADDR GTR A THEN GO FOUND;
    IF SCRATCHVEC[I+H],SHADDR<=A THEN GO FOUND;
    L+2; % DO BINARY SEARCH ON I SUCH THAT 1<I<H AND SUCH THAT
SEARCH: % SCRATCHVEC[I],SHADDR GTR A GEQ SCRATCHVEC[I-1],SHADDR.
    IF SCRATCHVEC[I:=(L+Q)/2],SHADDR GTR A THEN
        IF SCRATCHVEC[I-1],SHADDR GTR A THEN Q:=I-1
        ELSE GO FOUND
    ELSE IF SCRATCHVEC[I:=I+1],SHADDR LEQ A THEN L:=I+1
    ELSE GO FOUND;
    GO SEARCH;
$ SET OMIT = NOT(DEBUGGING) OR OMIT
ERROR: % DETECTED MCP AND/OR HARDWARE ERROR VIA BAD CORE/DISK TABLES,
    SCRATCHDIRECTORYERROR(CORADDR,N+50);
$ POP OMIT
FOUND: % FOUND THE DESIRED INDEX I.
    DISKWAIT(-CORADDR,30,SEGADDR:=SCRATCHVEC[I],SLINK);
$ SET OMIT = NOT(DEBUGGING) OR OMIT
    IF SCRATCHCHECK(I,H,SCRATCHSEG[0]) THEN SERROR(37+H);
$ POP OMIT
    IF (L:=SCRATCHVEC[I],SNUM) GEQ 29 THEN %MAKE ROOM FOR NEW ENTRY
        SCRATCHSPECIALCASE(1,A,N,CORADDR,SEGADDR,I,H,SCRATCHSEG)
    ELSE % ELSE MAKE A NORMAL ENTRY INTO THE I-TH SEGMENT
        BEGIN % AND WRITE IT BACK OUT TO DISK,
            SCRATCHVEC[I]+(IF (H=SCRATCHVEC[I],SRADDR)>A+N THEN H
                ELSE A+N) & (L:=L+1)[TOSNUM]
                & SEGADDR[TOSLINK];
            SCRATCHSEG[L]+A & N[TOSLENGTH];
            SCRATCHSEG[0],SNUM+L;
            DISKWAIT(CORADDR,30,SEGADDR);
        END;
    UNLOCKTOG(SCRATCHDIRECTORYMASK);
    FORGETSPACE(CORADDR);
    END SCRATCHDIRECTORYENTER;

PROCEDURE SCRATCHDIRECTORYDELETE(A,N);
VALUE A,N;
REAL A,N;
BEGIN
    INTEGER I;
    REAL L,H,E,F,Q,MAX,POT,CORADDR,SEGADDR;
    ARRAY SCRATCHSEG[*];
    LABEL SEARCH, FOUND;

```

```

05813620
05813635
05813700
05813800
05813900
05814000
05814100
05814200
05814300
05814400
05814449
05814450
05814451
05814500
05814700
05814800
05814850
05816600
05816700
05816800
05816900
05817000
05817100
05817200
05817300
05817400
05817500
05817599
05817600
05817700
05817701
05818000
05818100
05818199
05818200
05818201
05818300
05818400
05825800
05825900
05826000
05826100
05826200
05826300
05826400
05826500
05826600
05826800
05826900
05827000
05827100
05827200
05827300
05827400
05827500
05827600
05827700
05827800
05827900
05828000

```

```

$ SET OMIT = NOT(DEBUGGING) OR OMIT                                05828049
    LABEL ERROR;                                                  05828050
$ POP OMIT                                                         05828051
    % ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **   05828100
    FIXARRAY(SCRATCHSEG,CORADDR,30);                               05828200
    SLEEP([TOGGLE],SCRATCHDIRECTORYMASK); LOCKTOG(SCRATCHDIRECTORYMASK); 05828300
    Q:=H:=SCRATCHVEC[0],SSIZE;                                     05828350
$ SET OMIT = NOT(DEBUGGING) OR OMIT                                05828399
    IF SCRATCHVEC[Q],SHADDR LEQ A OR Q=0 THEN SERROR(H=0);        05828400
$ POP OMIT                                                         05828401
    IF SCRATCHVEC[I]=1],SHADDR GTR A THEN GO FOUND ;             05828500
    LI=2 ; % DO BIN SEARCH ON I SO THAT 1 LSS I LEQ H AND SUCH THAT 05828600
SEARCH: % SCRATCHVEC[I],SHADDR GTR A GEQ SCRATCHVEC[I-1],SHADDR. 05828700
    IF SCRATCHVEC[I]=(L+Q)/2],SHADDR GTR A THEN                   05828800
        IF SCRATCHVEC[I-1],SHADDR GTR A THEN Q:=I-1             05828900
        ELSE GO FOUND                                           05829000
    ELSE IF SCRATCHVEC[I]=I+1],SHADDR LEQ A THEN LI:=I+1         05829100
        ELSE GO FOUND ;                                         05829200
    GO SEARCH ;                                                  05829300
$ SET OMIT = NOT(DEBUGGING) OR OMIT                                05829399
ERROR: % DETECTED MCP AND/OR HARDWARE ERROR VIA BAD CORE/DISK TABLES, 05829400
    SCRATCHDIRECTORYERROR(CORADDR,N) ;                             05829500
$ POP OMIT                                                         05829501
FOUND: % FOUND THE DESIRED INDEX I,                                05829800
    DISKWAIT(=CORADDR,30,SEGADDR+(POT+SCRATCHVEC[I]),SLINK) ;    05829900
$ SET OMIT = NOT(DEBUGGING) OR OMIT                                05829999
    IF SCRATCHCHECK(I,H,SCRATCHSEG[0]) THEN SERROR(8+H) ;        05830000
$ POP OMIT                                                         05830001
    MAX=POT,SHADDR ;                                             05830100
    Q:=0 ;                                                         05830150
    POLISH(H) ;                                                  05830175
    FOR L=POT+POT,SNUM STEP -1 UNTIL 1 DO % FIND E, THE INDEX OF THE 05830200
        BEGIN % DELETABLE ENTRY,AND ALSO RECORD Q, THE VALUE OF THE 05830300
            IF (H+SCRATCHSEG[L],SRADDR) LEQ A % SECOND MAX,      05830400
                AND (F:=H+SCRATCHSEG[L],SLENGTH) GTR A THEN    05830450
$ SET OMIT = NOT(DEBUGGING) OR OMIT                                05830499
    IF E#0 THEN SERROR(29) ELSE                                   05830500
$ POP OMIT                                                         05830501
    E:=L;                                                         05830510
$ SET OMIT = NOT(DEBUGGING) OR OMIT                                05830549
    IF H=0 OR H=F THEN SERROR(30) ;                                05830550
$ POP OMIT                                                         05830551
    IF F#MAX AND Q<F THEN Q=F ;                                   05830600
    END ;                                                         05830700
    HI=POLISH ;                                                  05830750
    LI=(F:=SCRATCHSEG[E]),SRADDR+(F:=F,SLENGTH);                 05830800
$ SET OMIT = NOT(DEBUGGING) OR OMIT                                05830809
    IF (E=0) OR (A+N)#L THEN SERROR(2+E=0);                       05830810
$ POP OMIT                                                         05830811
    IF F GTR N THEN                                               05830860
        BEGIN % ONLY DELETE THE END=PART OF AN ENTRY            05830865
            SCRATCHSEG[E],SLENGTHI=F-N ;                          05830870
            IF MAX=L THEN SCRATCHVEC[I],SHADDR:=L-N ;           05830875
            DISKWAIT(CORADDR,30,SEGADDR) ;                        05830877
            END                                                    05830879
        ELSE % ELSE COMPLETELY DELETE THE ENTRY,                 05830880
            IF POT=1 THEN % THEN ENTRY TO DELETE IS ONLY ENTRY IN SEGMENT, 05830900
                SCRATCHSPECIALCASE(2,A,N,CORADDR,SEGADDR,I,H,SCRATCHSEG) 05831000
            ELSE % ELSE NORMAL CASE, WHERE AN ENTRY IS DELETED FROM 05836400
                BEGIN % A SEG AND SEG IS WRITTEN BACK OUT TO DISK, 05836500

```



```

SCRATCHSEG[E]*SCRATCHSEG[POT] ; 05836600
SCRATCHSEG[0],SNUM*POT-1 ; 05836700
DISKWAIT(CORADDR,30,SEGADDR) ; 05836800
SCRATCHVEC[I]*(IF A+N=MAX THEN @ ELSE MAX) & (POT-1)[TOSNUM] 05836900
& SEGADDR[TOSLINK] ; 05837000
END ; 05837100
UNLOCKTOG(SCRATCHDIRECTORYMASK); 05837300
FORGETSPACE(CORADDR) ; 05837400
END OF SCRATCHDIRECTORYDELETE ; 05837500

PROCEDURE SCRATCHCLEAN(ARY,BC,LINK) ; 05837600
VALUE BC ; % SCRATCHCLEAN SORTS THE SCRATCH DIRECTORY SEGMENT, AS 05837700
REAL BC,LINK ; % STORED IN ARRAY ARY, AND THEN IT GOES THROUGH THIS 05837800
ARRAY ARY[*] ; % SORTED ARRAY CONSOLIDATING ADJACENT AREAS AND DOING 05837900
BEGIN % FORGETUSERDISKS ON THESE (CONSOLIDATED) AREAS, 05838000
REAL I,K,S,T; 05838100
% ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** * * * * * 05838200
IF (LINK*ARY[0],SLINK)*0 THEN DISKIO(T,"BC,30,LINK) ; 05838300
SCRATCHSORT(ARY,K:=ARY[0],SNUM) ; 05838400
ARY[0]:=I:=K; % ARY[0] IS JUST A STOPPER 05838500
DO BEGIN 05838600
WHILE (S*ARY[K]),SLENGTH+S,SRADDR = ARY[K-1],SRADDR DO K*K-1; 05838700
FORGETUSERDISK(I*ARY[I],SRADDR,S,SRADDR+S,SLENGTH-I) ; 05838800
END 05838900
UNTIL (I*K*K-1)<1; 05839000
IF LINK*0 THEN SLEEP([T],IOMASK) ; 05839100
END OF SCRATCHCLEAN ; 05839200
$ POP OMIT 05839300

REAL PROCEDURE PETUSERDISK(N,T); VALUE N,T; REAL N,T ; 05839301
% N IS THE NUMBER OF SEGMENTS REQUESTED, AND T IS THE EU# OR THE SPEED#, 05839390
% GETUSERDISK WILL RETURN -1, 0, OR THE ABSOLUTE DISK SEGMENT ADDRESS OF 05839395
% THE RESULTANT AREA, SEE T,[2:1] FOR THE -1, AND N,[2:1] FOR THE 0. 05839400
% T>0 => T IS A PREFERRED SPEED#: T=1,2,3,4,...., OR 31, 05839500
% T<0 => T IS A PREFERRED EU#: T=-1,-2,-3,-4,...., OR -20. 05839600
% T=0 => DONT CARE ABOUT SPEED# OR EU#, USE EU WITH LEAST EU I/O, 05839700
% T,[2:1]=1 => IF CANT GET PREFERRED SPEED# OR EU#, RETURN -1, 05839800
% T,[2:1]=0 => IF CANT GET PREFERRED SPEED# OR EU#, TREAT AS T=0 (ABOVE) 05839900
% N>0 => MAKE A SCRATCHDIRECTORY ENTRY. 05840000
% N<0 => DONT MAKE A SCRATCHDIRECTORY ENTRY. 05840100
% N=0 => IMMEDIATELY RETURN WITH A 0, 05840200
% N,[2:1]=0 => IF CANT FIND ANY USERDISK, AND T,[2:1]=0, NO=USER=DISK, 05840300
% N,[2:1]=1 => IF CANT FIND ANY USERDISK, AND T,[2:1]=0, RETURN 0, 05840400
BEGIN 05840500
INTEGER K:=+1, % K IS ALSO "GETUSERDISK"; DONT USE K ABOVE LABEL D, 05840600
Z:=K+1,NS:=Z+1,I:=NS+1,Q:=I+1, 05840700
$ SET OMIT = NOT(SHAREDISK ) 05840800
L:=Q+1,H:=L+1,J:=H+1,R:=J+1,DL:=R+1; 05840900
REAL M1:=DL+1, M2:=M1+1; ARRAY U=M2+1[*]; DEFINE UT=U # ; 05841000
$ POP OMIT 05841100
$ SET OMIT = SHAREDISK 05841200
R:=Q+1, AVS:=R+1, J:=AVS+1, L:=AVS, H:=NT6 ; 05841300
REAL M1:=NT5, M2:=NT4; ARRAY UT=J+1[*]; DEFINE U=AVTABLE # ; 05841350
$ POP OMIT 05841380
05841400
05841500
05841502
05841610
05841615
05841620
05841621

```

```

LABEL A,B,C,D,E,F,G,W ; 05841650
DEFINE GETUSERDISK=PETUSERDISK#;*****05841700
%*****05841800
%*****05841900
IF N=0 THEN GO W ; 05842000
P(T,[2:1],ABS(N),1,0,0,0,0,0); 05842100
$ SET OMIT = NOT(SHAREDISK ) 05842200
P(O,D,DUP,0) ; 05842205
$ POP OMIT 05842210
A; 05842211
SLEEP([TOGGLE],USERDISKMASK); LOCKTOG(USERDISKMASK); 05842250
$ SET OMIT = NOT(SHAREDISK) 05842250
FIXARRAY(U,R,AVS); DISKWAIT(-R,-AVS,USERDISKBOTTOM); 05842310
$ POP OMIT 05842390
$ SET OMIT = SHAREDISK 05842400
M1:=M2:=P(D) ; 05842400
$ POP OMIT 05842401
L:=NEUP,NEUF ; 05842405
IF T LSS 0 THEN IF U[J]=IF -T GTR L THEN L+1 ELSE -T],MAXSIZ GEQ NS 05842410
THEN GO E ELSE IF Z THEN GO C ; 05842411
B; IF U[I],MAXSIZ>NS THEN 05842450
BEGIN 05842475
P(EUIO[(NT1:=I-1)+EUIOFFSET]+PEUIO[NT1],,NT2,SND,DUP) ; 05842500
IF P LSS M1 THEN BEGIN M1:=NT2; H:=NT1 END ; 05842700
IF P LSS M2 THEN IF U[I],SPEED=T THEN BEGIN M2:=NT2;J:=NT1 END; 05842800
END ; 05842900
IF (I:=I+1) LEQ L THEN GO B ; 05842930
IF P(D)#M1 THEN 05843000
BEGIN 05843100
IF M2#M2:=P(D) THEN IF Z AND T NEQ 0 THEN 05843200
BEGIN GETUSERDISK←-1; GO G END 05843300
ELSE J←H ; 05843400
J:=J+1; GO E ; 05843500
END ; 05843600
IF Z THEN GO C ; 05843700
IF N,[2:1] THEN GO G ; 05843800
$ SET OMIT = NOT(SHAREDISK ) 05843900
UNLOCK(USERDISKBOTTOM);USERDISKSPECIALCASE(I:=1,R,U,NS); 05843950
IF (Q:=R) THEN GO W;GO A; 05844000
$ POP OMIT 05844050
$ SET OMIT = SHAREDISK 05844070
FIXARRAY(UT,R,30); USERDISKSPECIALCASE(I←1,R,UT,NS); GO TO A; 05844072
$ POP OMIT 05844073
D:=@0777777777777777 ; 05844090
$ SET OMIT = NOT(SHAREDISK ) 05844110
E; IF (K←(I←(M1+U[J]),STARTWRD)+T+M1 AND NUMENTM)>AVS THEN 05844111
BEGIN 05844200
IF (L:=K+I)=(DL:=IF I LSS 30 THEN 0 ELSE IF (U[0] AND 05844290
NUMENTM) LEQ 30 THEN 30 ELSE 60)+(H:=I) MOD 30) MOD 30) NEQ 0 05844300
THEN L:=30=L ; 05844400
IF (L:=K+L) GTR AVS THEN 05844500
BEGIN USERDISKSPECIALCASE(O,R,U,L&DL[CTF]); U←FLAG(P) END; 05844530
IF P(DL#0,DUP) THEN P(UT[DL-1],XCH) ; 05844540
DISKWAIT(-(R+DL),L:=L-DL,Z:=USERDISKBOTTOM+H DIV 30) ; 05844570
IF P THEN UT[DL-1]:=P(XCH) ; 05844600
END ; 05844650
P(K=1); NT3:=K:=M1,MAXSIZ; NT2:=0; 05844700
$ POP OMIT 05844750
$ SET OMIT = SHAREDISK 05844800
05844900
05844901
05844915

```

```

E: IF (AVS:=K:=U[J] AND NUMENTM)+I:=Z:=U[J],STARTWRD) MOD 30) MOD05844920
    30) NEQ 0 THEN AVS:=30-AVS; AVS:=AVS+K; P(M2); 05844925
    FIXARRAY(UT,R,AVS); DISKWAIT(-R,AVS,Z+Z DIV 30+USERDISKBOTTOM); 05844930
    M2:=P; P(K-1); NT2:=0; NT3:=K:=U[J],MAXSIZ; 05844935
$ POP OMIT 05844936
F: IF (NT1+UT[I],DSIZE)>NT2 THEN IF NT1#K THEN NT2+NT1 ELSE K:=0; 05845000
    IF NT1#NS THEN IF NT1<M2 THEN BEGIN M2+NT1; H+I END; 05845100
    IF P(DUP) GTR I:=I+1 THEN GO F; 05845200
    UT[H],DSIZE+NS+M2-NS; 05845300
    IF M1:=M2=NT3 THEN U[J],MAXSIZ:=IF NT2>NS THEN NT2 ELSE NS; 05845400
    GETUSERDISK+UT[H],DEND=M2; I:=P; 05845500
$ SET OMIT = NOT(SHAREDISK) 05845590
    IF N>0 THEN SCRATCHDIRECTORYENTER(K,N); 05845600
$ POP OMIT 05845601
    IF N+NS=0 THEN BEGIN MOVE(I=H,[UT[H+1]],[UT[H]]);U[J],NUMENT+T-1END;05845700
$ SET OMIT = NOT(SHAREDISK) 05845790
    IF Z GTR 1 THEN 05845800
        BEGIN 05845900
            IF DL NEQ 0 AND (M1 OR N) THEN DISKWAIT(R,DL,USERDISKBOTTOM); 05846100
            DISKWAIT(DL+R,L,Z); 05846110
        END 05846200
    ELSE DISKWAIT(R,AVS,USERDISKBOTTOM); 05846300
$ POP OMIT 05846301
$ SET OMIT = SHAREDISK 05846350
    DISKWAIT(R,AVS,Z); 05846355
$ POP OMIT 05846356
    IF Q,[FF] THEN 05846361
        BEGIN 05846362
            M[I:=GETAREA(0)]:=(*P(DUP))&LOGLINE[CTF]&16[18:41:7]; 05846363
            M[I+1]:=0; 05846364
            QUEVENT(I,CANDYINX); 05846365
        END; 05846366
$ SET OMIT = NOT(SHAREDISK) 05846370
G: FORGETSPACE(R); UNLOCK(USERDISKBOTTOM); UNLOCKTOG(USERDISKMASK); 05846380
$ POP OMIT 05846381
$ SET OMIT = SHAREDISK 05846385
    FORGETSPACE(R); 05846390
G: UNLOCKTOG(USERDISKMASK); 05846395
$ POP OMIT 05846396
W: END OF GETUSERDISK ; 05846500
05846590
05846595
05847000
05847010
05847020
05847030
05847040
05847050
05847060
05847070
05847080
05847090
05847100
05847101
05847110
05847120
05847121
05847130
05847140
05847150

```

```

PROCEDURE FORGETUSERDISK(A,N); VALUE A,N; REAL A,N ;

```

```

% A IS THE ABSOLUTE DISK SEGMENT ADDRESS OF AN AREA N SEGMENTS LONG
% WHICH IS TO BE MADE AVAILABLE AGAIN,
% N<0 => MAKE A SCRATCHDIRECTORY DELETION,
% N>0 => DONT MAKE A SCRATCHDIRECTORY DELETION,
% N=0 => IMMEDIATELY GO AWAY ;

```

```

    BEGIN
$ SET OMIT = NOT(SHAREDISK)
    INTEGER DL,F; ARRAY U[*]; DEFINE UT=U #;
$ POP OMIT
$ SET OMIT = SHAREDISK
    INTEGER AVS,F=AVS; ARRAY UT[*]; DEFINE U=AVTABLE #;
$ POP OMIT
    REAL E; INTEGER B,C,D,I,J,R,S,H=NT7,K=NT6,L=NT5,G=NT4,T=NT3,Q=JUNK;
    LABEL V,W,X,Y,Z,AZ,BZ,CZ,DZ ;
    SUBROUTINE SETSHIFT ;

```

```

        BEGIN
        S:=P(XCH) ;
$ SET OMIT = NOT(SHAREDISK )
        U[J],STARTWRD:=G:=I+S; IF B>1 THEN G:=D+S+DL; D:=30 ;
$ POP OMIT
$ SET OMIT = SHAREDISK
        U[J],STARTWRD:=I+S; G:=D+S ;
$ POP OMIT
        K:=G+C-1 ;
        END OF SETSHIFT ;

% * * * * *

        IF N=0 OR (J+A DIV 1000000)≥NEUP,NEUF OR A<USERDISKBOTTOM+
        DISKAVALTABLEMAX THEN GO BZ ;
$ SET OMIT = NOT(SHAREDISK)
        FIXARRAY(U,R,AVS);
$ POP OMIT
        SLEEP([TOGGLE],USERDISKMASK); LOCKTOG(USERDISKMASK);
$ SET OMIT = NOT(SHAREDISK )
        DISKWAIT(-R,-AVS,USERDISKBOTTOM);
$ POP OMIT
        IF (D:=U[0],MAXSIZ) NEQ 0 AND N GTR 0 THEN IF (TWO(J) AND D) NEQ 0
        THEN BEGIN USERDISKSPECIALCASE(3,N,U,A); IF NOT P THEN GO DZ END ;
        J:=J+1 ;
V: D*(I+(E+U[J]),STARTWRD) MOD 30 ;
$ SET OMIT = NOT(SHAREDISK )
        IF (S:=K:=(C:=E AND NUMENTM)+L:=I) GEQ (F:=AVS) THEN
        BEGIN
        P((DL:=IF I LSS 30 THEN 0 ELSE IF (U[0] AND NUMENTM) LEQ 30
        THEN 30 ELSE 60)+D,DUP,C,+,DUP,DUP) ;
        IF (F:=P-P MOD 30+30) GEQ AVS THEN
        BEGIN USERDISKSPECIALCASE(0,R,U,F&DL[CTF]); U+FLAG(P) END;
        IF I+F>AVTMAX+D+DL THEN GO Y ;
        IF P(DL≠0,DUP) THEN P(UT[DL=1],XCH) ;
        DISKWAIT(-(R+DL),F=DL,B:=USERDISKBOTTOM+I DIV 30) ;
        IF P THEN UT[DL=1]:=P(XCH); K:=P; L:=P ;
        END ;
$ POP OMIT
$ SET OMIT = SHAREDISK
        AVS:=30*(S:=(C:=E AND NUMENTM)+D) MOD 30+S ;
        FIXARRAY(UT,R,AVS); DISKWAIT(-R,AVS,B:=I DIV 30+USERDISKBOTTOM) ;
        K:=S; L:=D; S:=I+C ;
$ POP OMIT
        G:=I-(NT2:=(P(U[J-1],DUP) AND NUMENTM)+P(XCH),STARTWRD) ;
        S:=U[J+1],STARTWRD=S; H+K+K-1; IF UT[T-L].DEND≥A THEN GO X ;
W: IF UT[T-(H+L+1)DIV 2].DEND≥A THEN IF UT[H+T-1].DEND≥A THEN GO W ELSE
        ELSE IF UT[T+1].DEND<A THEN BEGIN L+T+1; GO W END ;
X: IF (L:=A+ABS(N)) GEQ H:=P(UT[Q:=T],DUP).DEND=P(XCH).DSIZE THEN GO Z;
        IF S=0 THEN
        BEGIN
$ SET OMIT = NOT(SHAREDISK )
        IF G=0 THEN GO Y; P((G+1) DIV 2) ;
        IF B>1 THEN
        BEGIN IF D=0 THEN GO Y; IF P(DUP) GTR D THEN P(DEL,D) END;
$ POP OMIT
$ SET OMIT = SHAREDISK
        IF G=0 OR D=0 THEN GO Y; IF P((G+1)DIV 2,DUP)>D THEN P(DEL,D);
$ POP OMIT
        P(SSN); SETSHIFT; MOVE(C,[UT[G=S]],UT[G]); T:=Q:=T+S ;

```

```

05847160
05847170
05847180
05847190
05847191
05847200
05847210
05847211
05847220
05847230
05847240
05847250
05847260
05847270
05847271
05847279
05847280
05847281
05847290
05847299
05847300
05847301
05847310
05847320
05847330
05847340
05847350
05847360
05847370
05847380
05847390
05847400
05847410
05847420
05847430
05847440
05847450
05847460
05847461
05847470
05847480
05847490
05847500
05847501
05847510
05847520
05847530
05847540
05847550
05847560
05847570
05847580
05847590
05847600
05847610
05847611
05847620
05847630
05847631
05847640

```

```

        END ;
        FOR H=K STEP -1 UNTIL T DO UT[H+1]=UT[H]; H=ABS(N); GO AZ ;
Y: USERDISKSPECIALCASE(2,E,UT,J) ;
$ SET OMIT = NOT(SHAREDISK )
  UT=FLAG(P); RI=UT,[CF]; B=0 ;
$ POP OMIT
  GO V ;
Z: IF P(UT[Q+Q+1],DUP),DEND=P(XCH),DSIZE≤L THEN GO Z ;
  IF P(UT[NT1+Q-1],DEND,DUP) LSS L THEN P(DEL,L) ;
  HI=(LI=P)-(IF A LSS H THEN A ELSE H) ;
  IF NT1 GTR T THEN MOVE(K-NT1,[UT[Q]],[UT[T+1]]) ;
AZ: UT[T]=L&H[TODSIZE]; C=(Q-T-Q+1)+C ;
  IF(S+S=Q)>T+IF AVDIFFMAX>T+C DIV 2 THEN AVDIFFMAX ELSE T THEN IF J=1
  OR S+G>T+(IF AVDIFFMAX>T+NT2 DIV 2 THEN AVDIFFMAX ELSE T) THEN GO Y
  ELSE BEGIN
    IF (NT1=F-1-K)=0 THEN GO Y ;
    IF P((S+G) DIV 2,DUP)>NT1 THEN P(DEL,NT1); SETSHIFT ;
    FOR NT1=K STEP -1 UNTIL G DO UT[NT1]=UT[NT1-S] ;
    END ;
    U[J]=(NT1+U[J])&C[TONUMENTJ&(IF E+(NT1+NT1,MAXSIZ)<H THEN H ELSE
    NT1)[TOMAXSIZ] ;
$ SET OMIT = NOT(SHAREDISK )
  IF B GTR 1 THEN
    BEGIN
      IF DL NEQ 0 AND (E OR Q NEQ 0 OR D=30)
      THEN DISKWAIT(R,DL,USERDISKBOTTOM) ;
      DISKWAIT(R+DL,F=DL,B) ;
    END
  ELSE DISKWAIT(R,AVS,USERDISKBOTTOM) ;
$ POP OMIT
$ SET OMIT = SHAREDISK
  DISKWAIT(R,AVS,B) ;
$ POP OMIT
$ SET OMIT = NOT(SHAREDISK )
  IF N<0 THEN SCRATCHDIRECTORYDELETE(A,-N) ;
DZ: FORGETSPACE(R); UNLOCKTOG(USERDISKMASK); UNLOCK(USERDISKBOTTOM);
$ POP OMIT
$ SET OMIT = SHAREDISK
  FORGETSPACE(R) ;
DZ: UNLOCKTOG(USERDISKMASK);
$ POP OMIT
BZ: END OF FORGETUSERDISK ;
PROCEDURE KRUNCHER(H); ARRAY H[*]; FORWARD;
PROCEDURE SCHEDLOOK(KTR,TYPE); VALUE KTR,TYPE; REAL KTR,TYPE;%
BEGIN REAL T1,T2,H1,H2,MS,S,BUFF=S,LL=H2;
  INTEGER I=NT1, J=NT2;
  LABEL NXT,ENDIT,ESETC,LOOP,LOOPD,ERROR;
  LABEL STRM;

  IF KTR NEQ 0 THEN GO TO ESETC; %ES,XS,SV,RY
  IF SCHEDTOG THEN SLEEP([SCHEDWRD],0);
  SCHEDWRD=ABS(SCHEDWRD);
  IF FRSTSCHED=SCHEDNUM THEN GO ENDIT;
  T1=(FRSTSCHED INX "FIL0000")&SYSNO[24:42:16];
  NXT: IF (H1=DIRECTORYSEARCH(T1,"SCHEDUL",4)) LSS 64 THEN
    BEGIN T1=0; GO ENDIT; END;
  IF T2 NEQ 0 THEN
    BEGIN M[H2+29]=T1; HEADERUNLOCK(T2,"SCHEDUL",H2) END;
  M[H1+28]=T2; T2=T1; H2=H1; S=S+1; T1=M[H2+29];
  IF TYPE THEN IF T1=0 THEN GO ENDIT ELSE GO NXT;

```

```

05847650
05847660
05847670
05847680
05847690
05847691
05847700
05847710
05847720
05847730
05847740
05847750
05847760
05847770
05847780
05847790
05847800
05847810
05847820
05847830
05847840
05847850
05847860
05847870
05847880
05847890
05847900
05847910
05847920
05847921
05847930
05847940
05847941
05847950
05847960
05847980
05847981
05847985
05847990
05848000
05848001
05848010
XR5005849900
05850000
05850100
05850200
05850300
05850310
05850400
05850500
05850600
05850700
05850800
05850900
05851000
05851100
05851200
05851300
05851400
05851500

```

```

MS:=SPACE(10);
I:=M[H2+6] DIV 3600; J:=I MOD 60; II:=I DIV 60;
STREAM(T2,U:=M[H2+2],I,J,T1=(I GTR 0),MS);
BEGIN SI:=LOC T2; SI:=SI+5; DS:=6LIT" TASK#"; DS:=3CHR;
  DS:=LIT" "; SI:=SI+1; DS:=7CHR;
  T(DS:=7LIT" AFTER "; DS:=2DEC; DS:=LIT" "; DS:=2DEC);
  DS:=LIT" ";
END;
SPOUT(MS); IF T1 NEQ 0 THEN GO TO NXT;
ENDIT; IF T2=0 THEN FRSTSCHED:=SCHEDNUM
ELSE
BEGIN LSTSCHED:=T2; M[H2+29]=0;
  HEADERUNLOCK(T2,"SCHEDUL",H2);
END;
CHANGEDATE(0); SCHEDWRD:=NABS(SCHEDWRD);
IF TYPE LSS 0 AND S GTR 0 THEN
BEGIN; STREAM(S,T1:=T1:=SPACE(10));
  BEGIN DS:=11LIT" THERE ARE"; T1:=DI; SI:=LOC S;
    DS:=4 DEC; DS:=15LIT" TASKS ON DISK";
    DI:=T1; DS:=3 FILL;
  END;
  SPOUT(T1);
END;
IF TYPE=0 THEN % TS MESSAGE
BEGIN T1:=SPACE(5);
  IF S=0 THEN
  BEGIN STREAM(T1); DS:=20 LIT" NULL TASK SCHEDULE"; END
  ELSE STREAM(T1); DS:=7 LIT"END TS";
  SPOUT(T1);
END;
P(XIT);
ESETC; BUFF:=KTR;
LOOP;
STREAM(T:=0,LL:=0,KTR:Z:=(TYPE LSS 2),BUFF);
BEGIN SI:=KTR;
  L1: IF SC="*" THEN GO TO L4;
    IF SC LSS "0" THEN BEGIN SI:=SI+1; GO TO L1 END;
    IF SC GTR "9" THEN BEGIN SI:=SI+1; GO TO L1 END;
    2(40(IF SC="*" THEN JUMP OUT 2 TO L2; DS:=CHR));
  L2: DS:=LIT" "; DI:=LOC LL; SI:=BUFF;
    3(IF SC LSS "0" THEN JUMP OUT;
      IF SC GTR "9" THEN JUMP OUT;
      TALLY:=TALLY+1; SI:=SI+1);
    SI:=BUFF; BUFF:=TALLY;
    Z(DI:=DI+8; DI:=DI-BUFF; DS:=BUFF CHR; JUMP OUT TO L3);
    DS:=BUFF OCT;
  L3: IF SC=" " THEN BEGIN SI:=SI+1; GO TO L3 END;
    IF SC="," THEN SI:=SI+1;
    IF SC="*" THEN
  L4: BEGIN TALLY:=1; T:=TALLY END;
    KTR:=SI;
END;
KTR:=P; LL:=P; T1:=P;
IF LL=0 THEN GO TO ERROR;
IF TYPE LSS 2 THEN %ES OR XS
BEGIN
  FORK(P(.SCHEDIDLE),(NOT 0)&LL[CTC]&TYPE[2:32:16],0,160,0);
  GO TO LOOPD;
END;
IF LL GTR LMAX THEN GO TO ERROR;

```

```

05851600
05851700
%09805851800
05851900
05852000
05852100
05852200
05852300
05852400
05852500
05852600
05852700
05852800
05852900
05853000
05853100
05853200
05853300
05853400
05853500
05853600
05853700
05853800
05853900
05853950
05854000
05854050
05854100
05854200
05854300
05854400
05854500
05854600
05854700
05854800
05854900
05854925
05854950
05855000
05855100
05855200
05855300
05855400
05855500
05855600
05855650
05855700
05855800
05855900
05856000
05856100
05856200
05856300
05856400
05856500
05856600
05856700
05856800
05856900
05857000

```

IF NOT SCHEDLINE[LL] THEN	05857150
ERROR: BEGIN NT1:=" INVALID"; NT2:="D+ "; GO TO STRM END;	05857160
SCHEDBUSY[LL]:=TYPE;	05857170
NT1:=IF TYPE THEN " SAVED+" ELSE " READY+";	05857180
NT2:=0;	05857190
STRM:	05857500
STREAM(LL,NT1,NT2,MS:=MS:=SPACE(5));	05857550
BEGIN DS:=6LIT" LINE "; SI:=LOC LL; DS:=3 DEC;	05857600
2(SI:=SI+1; DS:=7 CHR); DI:=DI-17; DS:=2 FILL;	05857700
END;	05857800
SPOUT(MS); T2:=T2+1;	05857900
LOOPD: IF NOT T1 THEN GO TO LOOP;	05858000
IF (TYPE=2) AND (T2 GTR 0) THEN	05858400
IF SCHEDNUM NEQ FRSTSCHED THEN FORK(PC,SCHEDIDLE),0,0,160,0);	05858500
END SCHEDLOOK;	05858600
PROCEDURE SCHEDIDLE(ADR); VALUE ADR; REAL ADR;	05858700
BEGIN	05858800
REAL RCW:=+0;	05858900
ARRAY A[*];	05859000
REAL LL,NXTSCHED, T1,T2,T,TIME;	05859100
INTEGER I=NT1;	05859200
LABEL LINKIT, LUKAGN, LINKED, INIT1, FORGT1, TERMNATE,	05859300
REMOVIT, EXITR;	05859400
LABEL INIT, STRT, RPT, RSTRT, FORGET, GOTO, EXIT;	05859500
REAL N1,N2,SCH,FILE;	05859600
REAL CHARGE; INTEGER AFTER;	%09805859610
DEFINE PT=A[30]#	05859700
, IO=A[31]#	05859800
, DELTA=A[32]#	05859900
, N=A[33]#	05860000
, S=A[34]#	05860100
, R=A[35]#	05860200
, H=A[36]#	05860300
, NR=A[37]#	05860400
, MR=A[38]#	05860500
,	05860600
	05860700
DEFINE FORGETDISK =	05860800
BEGIN	05860900
FOR T1:=(M[T+9]+9) STEP =1 UNTIL 10	05861000
DO IF M[T1+T] NEQ 0 THEN FORGETUSERDISK(M[T1+T],M[T+8]);	05861100
END#;	05861200
LABEL DLOUT;	05861300
SUBROUTINE DELINK;	05861400
BEGIN	05861500
IF M[T+28]=0 THEN	05861600
BEGIN IF M[T+29]=0 THEN	05861700
BEGIN FRSTSCHED:=SCHEDNUM; CHANGEDATE(0); GO DLOUT END	05861800
ELSE BEGIN FRSTSCHED:=M[T+29]; CHANGEDATE(0) END	05861900
END ELSE	05862000
IF (T2=DIRECTORYSEARCH(M[T+28],SCH,4)) GEQ 64 THEN	05862100
BEGIN M[T2+29]:=M[T+29];	05862200
HEADERUNLOCK(M[T+28],SCH,T2);	05862300
END;	05862400
IF M[T+29]=0 THEN BEGIN LSTSCHED:=M[T+28]; CHANGEDATE(0) END	05862500
ELSE	05862600
IF (T2=DIRECTORYSEARCH(M[T+29],SCH,4)) GEQ 64 THEN	05862700
BEGIN M[T2+28]:=M[T+28];	05862800
HEADERUNLOCK(M[T+29],SCH,T2);	05862900
END;	05863000

DLOUT:	05863100
END:	05863200
DEFINE MYSTACK =	05863300
([RCW] INX NOT 3),[CF]#;	05863400
LL:=ADR,[CF]; SCH:="SCHEDUL"; FILE:="FIL0000"&SYSNO[24:42:6];	05863500
IF ADR,[1:1] THEN GO TERMNATE;	05863600
IF ADR,[2:1] THEN GO LINKIT;	05863700
IF LL=0 THEN GO TO INIT;	05863800
% "GO TO RECORD"... MAY BE RESTART ALSO,	05863900
SYSDISKIO(1,-LL,A:=(IOQUE&(GETAREA(1)+1))[CTC]);	05864000
N1:=A[1]; T1:=A[2]; N2:=A[3];	05864100
FORGETAREA(1,A INX NOT 0);	05864200
NXTSCHED:=(T1,[CF]);	05864300
IF (T1=SEQARRAY[LL],[CF]) LSS 512 THEN GO TO RSTRT;	05864310
IF SCHEND[LL] THEN GO EXIT;	05864400
A:=[M[T]]&80[8:38:10];	05864500
IF NOT(A[72],[1:1]) THEN SLEEP([A[72]],-0);	05864510
A[72]:=ABS(*P(DUP));	05864600
IF NOT(A[32],[1:1]) THEN SLEEP([A[32]],-0);	05864700
A[32]:=ABS(*P(DUP));	05864800
A[37]:=N2,[CF]; A[77]:=N2,[FF];	05864900
GO TO GOTO;	05865000
% LINK IN NEW SCHEDULE TASK	05865100
LINKIT:;	05865200
IF SCHEDTOG THEN SLEEP([SCHEDWRD],-0);	05865300
SCHEDWRD:=ABS(SCHEDWRD);	05865400
NXTSCHED:=SCHEDNUM INX FILE;	05865500
P(DIRECTORYSEARCH(NXTSCHED,SCH,6),DEL);	05865600
IF (T1=DIRECTORYSEARCH(M[ADR+5],-M[ADR+2],4)) LSS 64 THEN	05865700
BEGIN T:=ADR; FORGETDISK; FORGETSPACE(T); GO EXITR END;	05865800
M[T+6]:=(NOT 0)&NXTSCHED[CTC];	05866000
HEADERUNLOCK(M[ADR+5],M[ADR+2],T);	05866100
IF FRSTSCHED=SCHEDNUM THEN GO TO LINKED;	05866200
T1:=LSTSCHED INX FILE;	05866300
IF (TIME:=M[ADR+6])=0 THEN	05866400
IF (TIME:=XCLOCK) LSS 1728000 THEN TIME:=5184000;	05866500
IF (N1:=DIRECTORYSEARCH(T1,SCH,4)) LSS 64 THEN	05866600
BEGIN IF T2 NEQ 0 THEN HEADERUNLOCK(T2,SCH,N2);	05866700
SCHEDWRD:=NABS(SCHEDWRD); SCHEDLOOK(0,1);	05866800
T2:=0; GO LINKIT;	05866900
END;	05867000
IF M[N1+6] GTR TIME THEN	05867100
BEGIN IF T2 NEQ 0 THEN HEADERUNLOCK(T2,SCH,N2);	05867200
T2:=T1; N2:=N1;	05867300
IF (T1:=M[N2+28]) NEQ 0 THEN GO LUKAGN;	05867400
END;	05867500
LINKED:;	05867600
M[ADR+28]:=T1; M[ADR+29]:=T2;	05867700
ENTERUSERFILE(-NXTSCHED,SCH,ADR,[CF]=1);	05867800
T:=SCHEDNUM;	05867900
STREAM(A1:=1, A2:=T);	05868000
BEGIN SI:=LOC A1; DS:=8 ADD END;	05868100
SCHEDNUM:=IF T,[CF]=0 THEN 1 ELSE T;	05868200
IF T1 NEQ 0 THEN	05868300
BEGIN M[N1+29]:=NXTSCHED; HEADERUNLOCK(T1,SCH,N1) END	05868400
ELSE FRSTSCHED:=NXTSCHED;	05868500
IF T2=0 THEN LSTSCHED:=NXTSCHED;	05868600
CHANGEDATE(0);	05868700
IF T2 NEQ 0 THEN	05868800
	05868900



```

BEGIN MIN2+28]:=NXTSCHED; HEADERUNLOCK(T2,SCH,N2) END; 05869000
FORGETSPACE(ADR); 05869100
ADR:=0; 05869200
GO TO INIT1; 05869300
% END OF LINK IN NEW SCHEDULE TASK 05869400
INIT1: IF SCHEDTOG THEN SLEEP([SCHEDWRD],-0); 05869500
SCHEDWRD:=ABS(SCHEDWRD); 05869600
INIT1: IF (NXTSCHED:=FRSTSCHED)=SCHEDNUM THEN GO EXITR; %R8905869700
LL:=0; %R8905869750
IF (TIME:=XCLOCK) LSS 1728000 THEN TIME:=5184000; 05869800
STRT: IF LINEDISC[LL:=LL+1] # SCHED THEN GO TO EXITR; 05869900
IF STABLE[LL],DIALEDUP THEN GO TO STRT; 05870000
IF SCHEDBUSY[LL] THEN GO TO STRT; 05870100
IF SEQARRAY[LL] NEQ 0 THEN GO TO STRT; 05870200
SEQARRAY[LL]:=1; 05870300
RPT: IF NXTSCHED=0 THEN 05870400
BEGIN SEQARRAY[LL]:=0; GO EXITR END; 05870500
T1:=NXTSCHED INX FILE; N1:=N2:=0; 05870600
RSTRT: IF (T:=DIRECTORYSEARCH(T1,SCH,4)) LSS 64 THEN 05870700
IF NXTSCHED LSS 0 THEN 05870800
BEGIN A:=[M[SPACE(10)]]&10[8:38:10]; 05870900
GO TO FORGT1; 05871000
END 05871100
ELSE 05871200
BEGIN SEQARRAY[LL]:=0; LL:=0; 05871300
%TRY TO RESTORE QUEUE 05871400
SCHEDWRD:=NABS(SCHEDWRD); SCHEDLOOK(0,1); 05871500
GO TO INIT; 05871600
END; 05871700
IF NXTSCHED GEQ 0 THEN 05871800
BEGIN NXTSCHED:=M[T+29],[CF]; 05871900
IF TIME LSS M[T+6] THEN 05872000
BEGIN HEADERUNLOCK(T1,SCH,T); 05872100
SEQARRAY[LL]:=0; GO TO EXITR 05872200
END; 05872300
IF M[T+6],[2:1] THEN %ITS BEING TERMINATED %09805872400
BEGIN HEADERUNLOCK(T1,SCH,T); GO TO RPT END; 05872500
DELINK; 05872600
END; 05872700
A:=[M[SPACE(80)]]&80[8:38:10]; 05872800
A[0]:=0; MOVE(79,A,A INX 1); %ZERO OUT ARRAY 05872900
DISKWAIT(-(A INX 40),30,M[T INX 10]); %FILL BUFF 05873000
A[70]:=T1,[CF]; %SCHEDULE NUMBER 05873100
A[71]:=IOMASK; 05873200
A[74]:=M[T INX 8];%ROW LENGTH OF IP FILE 05873300
A[75]:=M[T INX 10];%ADDRESS OF 1ST ROW 05873400
A[76]:=T,[FF]; %ADDRESS OF HEADER 05873500
A[77]:=N2,[FF]; 05873600
A[78]:=M[T INX 7]+2; %NUMBER OF RECORDS 05873700
A[37]:=N2,[CF]; 05873800
N1:=M[T INX 2]; %GET USERCODE 05873900
N2:=M[T INX 5]; %GET O/P FILE NAME 05874000
AFTER:=ABS(M[T INX 6]) DIV 3600; %TIME AFTER IN MIN %09805874010
CHARGE:=M[T INX 27]; %CHARGE CODE %09805874020
FORGETSPACE(T); 05874100
IF (T:=DIRECTORYSEARCH(N2,-N1,2)) LSS 64 THEN 05874200
BEGIN 05874300
FORGET: T:=DIRECTORYSEARCH(T1,SCH,8); 05874400
FORGETDISK; FORGETSPACE(T); 05874500
IF NXTSCHED GEQ 0 THEN 05874600

```

	BEGIN FORGETSPACE(A); GO TO RPT; END;	05874700
FORGT1:	SYSDISKIO(3,=LL,A);	05874800
	A[0],DIALEDUPI=A[1]=0;	05874900
	SYSDISKIO(0,=LL,A); FORGETSPACE(A);	05875000
	GO TO EXIT;	05875100
	END;	05875200
	IF NXTSCHED GEQ 0 THEN	05875300
	IF M[T+4],[36:6]=TYPEINFO THEN	05875400
	BEGIN FORGETSPACE(T);	05875500
	FORGETSPACE(DIRECTORYSEARCH(N2,N1,12));	05875600
	GO TO FORGET	05875700
	END ELSE BEGIN	05875800
	DISKWAIT(=T,[CF],30,T,[FF]);	05875850
	M[T+4],[36:6]=TYPEINFO;	05875900
	DISKWAIT(T,[CF],30,T,[FF]);	05876000
	STREAM(A);	05876100
	BEGIN DS:=21 LIT"YOUR TASK IS RUNNING,";	05876200
	DS:=51 LIT " "; DS:=8 LIT"00000001";	05876300
	20(DS:=8LIT " ");	05876400
	END;	05876500
	A[32]:=10; A[37]:=1;	05876600
	END;	05876700
	A[31]:=IOMASK;	05876800
	A[34]:=M[T INX 8];	05876900
	A[35]:=M[T INX 10];	05877000
	A[36]:=T,[FF];	05877100
	A[38]:=A[34]xM[T INX 9],[40:8]x3; % MAX NUM OF OUP T RECS	05877200
	FORGETSPACE(T);	05877300
	IF NXTSCHED LSS 0 THEN	05877400
	BEGIN	05877500
GOTO1:	IF A[37] GEQ A[38] THEN A[37]:=A[38]-1;	05877600
	IF (N2:=(A[37]) DIV 3) NEQ A[33] THEN	05877700
	BEGIN A[37]:=(A[33]+N2)x3; A[32]:=0;	05877800
	IF (A[31] AND IOMASK)=0 THEN	05877900
	SLEEP([A[31]],IOMASK);	05878000
	DISKWAIT(=A,[CF],30,A[36]);	05878100
	IF (N2:=A[(I:=N2 DIV A[34])+10])=0 THEN	05878200
	GO TO FORGET;	05878300
	A[35]:=N2-(IxA[34]);	05878400
	END;	05878500
	IF A[77] GEQ A[78] THEN A[77]:=A[78]	05878600
	ELSE	05878700
	BEGIN A[72]:=(I:=(A[77]-1) MOD 3)x10;	05878800
	IF (N2:=(A[77]-1) DIV 3) NEQ A[73] THEN	05878900
	BEGIN A[73]:=N2;	05879000
	IF (A[71] AND IOMASK)=0 THEN	05879100
	SLEEP([A[71]],IOMASK);	05879200
	DISKWAIT(=(A INX 40),30,A[76]);	05879300
	IF (N2:=A[(I:=N2 DIV A[74])+50])=0 THEN	05879400
	A[77]:=A[78]	05879500
	ELSE BEGIN	05879600
	A[75]:=N2-(IxA[74]);	05879700
	DISKIO(A[71],=(A INX 39),30,N2);	05879800
	END;	05879900
	END;	05880000
	END;	05880100
	IF SEQARRAY[LL],[FF]=0 THEN	05880200
	BEGIN STATABLE[LL],DIALEDUPI=1;	05880300
	A[32]:=NABS(*P(DUP)); A[72]:=NABS(*P(DUP));	05880400
	IF SEQARRAY[LL] GEQ 0 THEN GO TO EXIT;	05880500

```

                SEQARRAY[LL] := ABS(*P(DUP));
                FORK(P(,SCHEDIO), (-LL), 0, 0, MYSTACK);
                GO TO NOTHINGTODO;
            END;
        END;
    END;
CONNECT(LL);
SEQARRAY[LL]:=A.[CF];
T := GETAREA(0);
M[T]:=0&9[18;41;7]&LL[25;40;8];
M[T+1]:=N1&1[2;47;1]&NXTSCHED[1;1;1];
IF (N2:=DIRECTORYSEARCH(T1 INX FILE,SCH,5)) NEQ 0 THEN
    BEGIN
        M[T+2] := M[N2 INX 5];
        FORGETSPACE(N2);
    END
ELSE M[T+2] := NOT 0;
M[T+3]:=CHARGE;
QUEVENT(T,[CF],CANDEMIX[LL]);
A[32]:=NABS(*P(DUP)); A[72]:=NABS(*P(DUP));
T:=SPACE(30); I:=AFTER MOD 60;
STREAM(N1,LL,CHARGE,AFT:=(AFTER DIV 60),I,
        X:=(NOT CHARGE),[1;1],Y:=(AFTER NEQ 0),T);
BEGIN DS:=LIT" "; SI:=LOC N1; SI:=SI+1; DS:=7 CHR;
        DS:=5LIT" ON "; T:=DI; DS:=3DEC; DS:=2LIT"=S";
        X(DS:=LIT"("; SI:=SI+1; DS:=7CHR; DS:=LIT")");
        Y(DS:=LIT"["; SI:=LOC AFT; DS:=2 DEC;
        DS:=LIT"i"; DS:=2 DEC; DS:=LIT"]");
        DS:=LIT LEFTARROW; DI:=T; DS:=2 FILL;
    END;
LOGLINE:=@1000+LL; SPOUTIT(T,INOUTK);
IF NXTSCHED LSS 0 THEN GO TO EXIT;
SYSDISKIO(3,-LL,(A:=A&(GETAREA(1)+1)[CTC]));
A[0],DIALEDUP:=1; A[1]:=N1; A[2]:=T1; A[3]:=1;
SYSDISKIO(0,-LL,A); FORGETAREA(1,A INX NOT 0);
GO TO STRT;
% END INITIATE/GOTO
TERMNATE:= IF ADR,[FF] NEQ 0 THEN GO REMOVI;
% TERMINATE A RUNNING TASK
IF SCHEND[LL] THEN GO EXIT; SCHEND[LL]:=1;
A:=[M[SEQARRAY[LL]]]&80[8;38;10];
IF NOT(A[72],[1;1]) THEN SLEEP([A[72]],=-0);
A[72]:=ABS(*P(DUP));
IF NOT(A[32],[1;1]) THEN SLEEP([A[32]],=-0);
A[32]:=ABS(*P(DUP));
IF DELTA NEQ 0 THEN DISKIO(10,A,[CF]=1,30,R+N);
IF (A[71] AND IOMASK)=0 THEN SLEEP([A[71]],IOMASK);
SYSDISKIO(3,-LL,40 INX A);
T1:=A[42]; %SAVE NAME OF SCHEDULE IP
A[40],DIALEDUP:=A[41]:=A[42]:=A[43]:=SEQARRAY[LL]:=0;
STATABLE[LL],DIALEDUP:=0;
SYSDISKIO(0,-LL,40 INX A);
T:=DIRECTORYSEARCH(T1,SCH,8);%DELETE DIR ENTRY
N1:=M[T INX 2]; N2:=M[T INX 5];
FORGETDISK; FORGETSPACE(T);
IF (10 AND IOMASK)=0 THEN SLEEP([10],IOMASK);
FORGETSPACE(DIRECTORYSEARCH(N2,N1,12));
IF (T:=DIRECTORYSEARCH(N2,N1,4)) GEQ 64 THEN
    BEGIN DISKWAIT(-A,[CF],30,(T1:=M[T INX 10]));
        STREAM(A); 9(DS:=8 LIT " ");
        DISKWAIT(A,[CF],30,T1); T1:=NR-1; FORGETSPACE(A);
    END

```

```

05880600
05880700
05880800
05880900
05881000
05881100
05881200
05881300
05881400
05881500
05881510
05881520
05881530
05881540
05881550
05881560
05881570
05881600
05881700
%09805881800
%09805881810
%09805881820
05881900
%09805882000
%09805882020
%09805882040
%09805882060
%09805882100
05882200
05882300
05882400
05882500
05882600
05882700
05882800
05882900
05883000
05883100
05883200
05883300
05883400
05883500
05883600
05883700
05883800
05883900
05884000
05884100
05884200
05884300
05884400
05884500
05884600
05884700
05884800
05884900
05885000
05885100
05885200
05885300

```

```

A.[CF]:=T; A[7]:=T1; A[5]:=A[6]:=0; 05885400
T1:=A[1]; A[1]:=A[8]; 05885420
$ SET OMIT = NOT SHAREDISK 05885430
A[4]:=(*P(DUP)) OR 1; 05885440
$ POP OMIT 05885450
KRUNCHER(A); A[1]:=T1; 05885460
$ SET OMIT = NOT SHAREDISK 05885470
A[4]:=(*P(DUP)) AND NOT 1; 05885480
$ POP OMIT 05885490
HEADERUNLOCK(N2,N1,T); 05885500
END ELSE FORGETSPACE(A); 05885600
ADR:=0; 05885700
IF FRSTSCHED NEQ SCHEDNUM THEN GO TO INIT; 05885800
GO TO EXIT; 05885900
% END TERMINATE A RUNNING TASK 05886000
REMOVIT;; 05886100
% TERMINATE A SCHEDULED TASK 05886200
% OR LINK IT IN FRONT (XS) 05886300
IF SCHEDTOG THEN SLEEP([SCHEDWRD],-0); 05886400
SCHEDWRD:=ABS(SCHEDWRD); 05886500
T1:=ADR INX FILE; 05886600
IF (T1=DIRECTORYSEARCH(T1,"SCH,4)) GEQ 64 THEN 05886700
BEGIN 05886800
IF (N1:=ADR,[16:2])NEQ 3 THEN % ES OR XS KEYIN 05886900
BEGIN; STREAM(N1,T1,N2:=N2:=SPACE(10)); 05887000
BEGIN DS:=6LIT" TASK#"; SI:=LOC T1; 05887100
SI:=SI+5; DS:=3 CHR; 05887200
N1(DS:=7LIT" RESET#"; JUMP OUT TO L); 05887300
DS:=9LIT" REMOVED#"; 05887400
L: END; 05887500
SPOUT(N2); 05887600
IF N1 THEN % XS == RELINK IT 05887700
BEGIN M[T+6]:=NABS(*P(DUP)); %09805887800
IF T1.[CF]=FRSTSCHED THEN 05887900
HEADERUNLOCK(T1,SCH,T) 05888000
ELSE 05888100
BEGIN DELINK; M[T+28]:=0; 05888200
M[T+29]:=T2:=FRSTSCHED INX FILE; 05888300
HEADERUNLOCK(T1,SCH,T); 05888400
FRSTSCHED:=T1; CHANGEDATE(0); 05888500
IF (T1=DIRECTORYSEARCH(T2,SCH,4)) 05888600
GEQ 64 THEN BEGIN M[T+28]:=T1; 05888700
HEADERUNLOCK(T2,SCH,T); 05888800
END; 05888900
END; 05889000
ADR:=0; 05889100
GO TO INIT1; 05889200
END; 05889300
END; 05889400
DELINK; 05889500
IF (N1=DIRECTORYSEARCH(M[T+5],M[T+2],4)) GEQ 64 THEN 05889600
BEGIN IF (M[N1+7] OR M[N1+4],[36:16])=0 THEN 05889700
BEGIN; STREAM(X:=ADR,[17:1],N2:=N2:=SPACE(10)); 05889800
BEGIN DS:=19 LIT"TASK TERMINATED BY "; 05889900
X(DS:=8 LIT"USER "; JUMP OUT TO L); 05890000
DS:=8 LIT"OPERATOR"; 05890100
L: DS:=53 LIT" "; 05890200
END; 05890300
DISKWAIT(N2,10,M[N1+10]); 05890400
FORGETSPACE(N2); 05890500

```

```

                END;
                M[N1+4],[3616] := 63; % "ABORTED"
                M[N1+5]:=M[N1+6]:=0;
                HEADERUNLOCK(M[T+5],M[T+2],N1);
            END;
            FORGETSPACE(DIRECTORYSEARCH(T1,SCH,8));
            FORGETDISK; FORGETSPACE(T);
        END
    ELSE IF (N1:=ADR,[16:2])NEQ 3 THEN % ES OR XS KEYIN
    BEGIN; STREAM(N1,T1,Z:=T,[1:1],N2:=N2:=SPACE(10));
        BEGIN DS:=6LIT",TASK#"; SI:=LOC T1;
            SI:=SI+5; DS:=3CHR; DS:=5LIT" NOT ";
            N1(DS:=5LIT"RESET"; JUMP OUT TO L);
            DS:=7LIT"REMOVED";
        L: Z(DS:=9LIT"(IN USE)+"; JUMP OUT TO L1);
            DS:=14LIT"(NOT ON DISK)+";
        L1:
        END;
        SPOUT(N2);
    END;
    % END TERMINATING OF A SCHEDULED TASK
    EXITR:= SCHEDWRD:=NABS(SCHEDWRD);
    EXIT:= KILL([RCW] INX NOT 2);
END SCHEDULE;
PROCEDURE SCHEDIO(NUM,TYPE,ADR); %IF FORQUED THEN ONLY 1 PARAM.
VALUE ADR,NUM,TYPE; REAL ADR,NUM,TYPE;
BEGIN
    REAL RCW:=+0;
    ARRAY A[*];
    REAL LL,T;
    REAL LSIZ=LL, K=T+1, LAS=K+1; % USED ONLY FOR WRITING.
    BOOLEAN FOLDING=LAS+1; % USED ONLY FOR WRITING.
    LABEL RDSTRT, FRSTRD, RD, WRITAGN, UPDATE, UNLOCKIT, EXIT;
    DEFINE PT=A[30]#
        ,IO=A[31]#
        ,DELTA=A[32]#
        ,N=A[33]#
        ,S=A[34]#
        ,R=A[35]#
        ,H=A[36]#
        ,NR=A[37]#
        ,MR=A[38]#
    ;

    IF (LL:=ADR,[FF])=0 THEN
    BEGIN LL:=ADR,[CF]; ADRI:=0; GO TO FRSTRD END;
    IF SCHEND[LL] THEN P(XIT);
    AI=[M[SEQARRAY[LL]]]&80[8:38:10];
    IF ADR,[1:1] THEN %INPUT
    % READ
    BEGIN AI:=40 INX A; GO TO RD;
    RDSTRT:= IF NOT SCHEDLINE[LL:=LL+1] THEN GO TO EXIT;
        IF STABLE[LL],DIALEDUP THEN
        IF SEQARRAY[LL],[1:2]=2 THEN
        BEGIN SEQARRAY[LL],[1:1]:=LOGLINE:=0;
    FRSTRD:= AI=[M[SEQARRAY[LL] INX 40]]&80[8:38:10];
        GO TO RD;
        END;
        GO TO RDSTRT;
    RD: IF NOT(DELTA,[1:1]) THEN SLEEP([DELTA],-0);

```

```

05890600
05890700
05890800
05890900
05891000
05891100
05891200
05891300
05891400
05891500
05891600
05891700
05891800
05891900
05892000
05892100
05892200
05892300
05892400
05892500
05892600
05892700
05892800
05892900
05893000
05893100
05893200
05893300
05893400
05893500
05893525
05893550
05893600
05893700
05893800
05893900
05894000
05894100
05894200
05894300
05894400
05894500
05894600
05894700
05894800
05894900
05895000
05895100
05895200
05895300
05895400
05895500
05895600
05895700
05895800
05895900
05896000
05896100
05896200
05896300

```

```

DELTA:=ABS(*P(DUP));                                05896400
IF (IO AND IOMASK)=0 THEN SLEEP([IO],IOMASK);        05896500
IF (NR:=NR+1) GEQ MR THEN %EOF                        05896600
IF (ADR:=ADR,[CF])=0 THEN %C&E REQUEST              05896700
  IF NR GEQ MR+2 THEN                                05896800
    A[DELTA]:=022702537373737373737373737373737 %BYE..... 05896900
    ELSE %SEND PLAIN LEFT ARROW FIRST IN CASE        05897000
    A[DELTA]:=NOT 0 %WE ARE IN SEQ MODE...          05897100
  ELSE BEGIN                                         05897200
    DELTA:=NABS(*P(DUP)); TERMINATE(P1MIX);          05897300
    TERMINALMESSAGE(48)                              05897400
  END;                                               05897500
A[DELTA+9]:=0;                                       05897600
IF ADR NEQ 0 THEN %NORMAL STATE PROGRAM              05897605
IF NUM GTR 0 THEN %QUESTION MARK REQUIRED            05897610
BEGIN T:=01437000000000000000;                     05897612
  SCHEDIO(2,08,[CTF],[T]INX(0&LL[CTF]));           05897614
END ELSE NUM:=ABS(NUM);                               05897616
SCHEDIO(72,1 OR M,(A INX DELTA)&LL[CTF]); %WRITE IT OUT 05897700
IF ADR=0 THEN %C&E REQUEST                           05897800
BEGIN ADR:=GETAREA(2);                               05897900
  M[ADR]:=0&LL[10:40:8]&9[2:44:4]; %SIZE AND MESSEND 05898000
  STREAM(SS:=A[DELTA],DD:=ADR&7[CTF]);              05898100
  BEGIN SI:=SS; DI:=DI+1;                            05898200
  L: IF SC=LEFTARROW THEN                            05898300
    BEGIN DI:=DD; DI:=DI+1;                          05898400
      DSI=CHR;                                        05898450
    END ELSE                                          05898500
    BEGIN IF SC NEQ " " THEN DD:=DI;                 05898600
      DSI=CHR; GO TO L;                              05898700
    END;                                             05898800
  END;                                               05898950
  GIVEAWAY(ADR);                                     05899000
  ADR:=0;                                            05899100
END ELSE %NOT C&E REQUEST                             05899200
BEGIN STREAM(NN:=IF (T:=NUM GTR 9) THEN 9 ELSE NUM,  05899300
  T:=IF T THEN NUM-9 ELSE 0, SSI=(A INX DELTA),      05899400
  ADR:=ADR,[CF]);                                    05899500
  BEGIN SI:=SS; DSI=NN WDS; T(DSI=8 LIT" ") END;    05899600
END;                                                 05899700
IF (DELTA:=DELTA+10)=30 THEN%END OF DISK SEGMENT    05899800
BEGIN IO:=DELTA:=0;                                  05899900
  IF (N:=N+1) MOD 5 = 0 THEN %NEW ROW                05900000
  BEGIN DISKWAIT(-A,[CF],30,H);                      05900100
    R:=A[(N DIV 5)+10]=N;                            05900200
  END;                                               05900300
  DISKIO(IO,(1-A,[CF]),30,R+N)                       05900400
END;                                                 05900500
DELTA:=NABS(*P(DUP));                                05900600
IF ADR=0 THEN GO TO RDSTRT;                          05900700
P(XIT);                                              05900800
END;                                                 05900900
% WRITE                                              05901000
P(0,0,0); % K, LAS, FOLDING                          05901020
IF TYPE,[1:1] THEN P(XIT); % 1/2ASCII STUFF NOT ALLOWED 05901040
IF NOT (DELTA,[1:1]) THEN SLEEP([DELTA],0);         05901060
DELTA:=ABS(*P(DUP));                                05901080
ADR:=ADR,[CF];                                       05901100
IF (LAS:=TYPE,[2:1]) OR TYPE,[FF] OR NUM GTR 72=PT THEN 05901120
BEGIN K:=T:=ADR; % SCAN FOR + OR END OF DATA 05901140

```

```

DO BEGIN                                05901160
  STREAM(LAS, T, K; I:=IF NUM GTR 63 THEN 63 ELSE NUM); 05901180
  BEGIN SI:=K;                                05901200
    I(CI:=CI+LAS; GO TO L;                    05901220
      IF SC=" " THEN                          05901240
        BEGIN T:=SI; TALLY:=1;              05901260
          JUMP OUT;                            05901280
        END;                                    05901300
      L: IF SC=" " THEN BEGIN SII:=SI+1; T:=SI END ELSE 05901320
        SII:=SI+1);                            05901340
        LASI=TALLY; KI:=SI;                    05901360
      END;                                    05901380
      KI:=P; T:=P;                              05901400
    END UNTIL P OR (NUM:=NUM-63) LEQ 0;        05901420
    NUM:=T.[30:3]&(T=ADR)[30:33:15];          05901440
  END;                                        05901460
WRITAGN: IF NR GEQ MR THEN GO UNLOCKIT;      % DONT WRITE IF FULL 05901500
IF (IO AND IOMASK)=0 THEN SLEEP([IO],IOMASK); 05901550
IF (DELTA)=0 THEN                            05901600
BEGIN; STREAM(A:=A.[CF]);                     05901700
  BEGIN DS:=8 LIT" "; SI:=A; DS:=29 WDS END;  05901800
END;                                          05901900
NR:=NR+1; T:=A INX DELTA;                    05902000
STREAM(NN:=NR, T1:=T+9);                     05902100
BEGIN SI:=LOC NN; DS:=8 DEC; END;            05902200
IF NR+10=MR THEN % CONSIDER THE FILE FULL    05902300
BEGIN                                          05902400
  STREAM(T); DS:=22 LIT"***END OF SCHEDULE TANK"; 05902500
  IF A[77] LSS A[78] THEN A[77]:=A[78]; % FORCE EOF ON INPT 05902600
  IF P1MIX GTR 0 AND P1MIX#CANDYINX THEN     05902700
  IF DAT[P1MIX],NDSABLE THEN BREAK[LLJ]=1 ELSE 05902800
  TERMINATE(P1MIX&67[CTF]);                 05902900
END                                          05903000
ELSE BEGIN                                    05903100
  IF FOLDING THEN % INDENT THE LINE          05903150
  BEGIN STREAM(T); BEGIN DII:=DI+10; DSI:=2 LIT"xx"; END; 05903200
  PT:=12;                                     05903250
END;                                          05903300
IF FOLDING:=((KI=NUM) GTR (LSIZ:=72-PT)) THEN 05903350
BEGIN STREAM(ADR: Q:=LSIZ-2, Q1:=P(DUP).[39:3], 05903400
  I:=IF NUM>LSIZ+34 THEN 24 ELSE LSIZ+58=NUM, D:=T+9); 05903450
  BEGIN SI:=ADR;                              05903500
    Q1(SI:=SI+32; SII:=SI+32); SII:=SI+Q;    05903550
    DII:=DI-2; TALLY:=2;                      05903600
    I(IF SC=" " THEN JUMP OUT TO L;          05903650
      TALLY:=TALLY+1; SII:=SI-1; DII:=DI-1); 05903700
    DII:=DI+I; TALLY:=2;                     05903750
  L: DSI:=2 LIT"xx"; ADR:=TALLY;            05903800
  END;                                        05903850
  KI:=LSIZ-P(XCH);                            05903900
END;                                          05903950
STREAM(ADR: K, I:=K GTR 64, D:=T+PT.[40:5]&PT[30:45:3]); 05904000
BEGIN SI:=ADR; DS:=K CHR;                    05904050
  I(DS:=32 CHR; DS:=32 CHR);                 05904100
  ADR:=SI;                                    05904150
END;                                          05904200
ADR:=P;                                       05904250
IF FOLDING THEN                              05904300
BEGIN                                          05904350
  UPDATE: IF (DELTA:=DELTA INX 10)=30 THEN 05904400

```

```

        BEGIN IO:=DELTA:=0;                                05904450
          DISKIO(IO,A,[CF]=1,30,R+N);                      05904500
          IF ((N:=N+1) MOD S)=0 THEN % END OF THE ROW    05904550
          BEGIN R:=GETUSERDISK(-S)=N;                     05904600
            SLEEP([IO],IOMASK);                            05904650
          LOCKDIRECTORY;                                    05904690
          DISKWAIT("A,[CF],-30,H);                        05904700
          A[(N DIV S)+10]:=R+N;                            05904710
          DISKIO(IO,A,[CF]=1,-30,H);                      05904750
          UNLOCKDIRECTORY;                                 05904800
          DISKWAIT("A,[CF],-30,H);                        05904850
          UNLOCKDIRECTORY;                                 05904890
          UNLOCKDIRECTORY;                                 05904900
          UNLOCKDIRECTORY;                                 05904910
          UNLOCKDIRECTORY;                                 05904950
          UNLOCKDIRECTORY;                                 05904950
          UNLOCKDIRECTORY;                                 05905000
          UNLOCKDIRECTORY;                                 05905050
          UNLOCKDIRECTORY;                                 05905100
          UNLOCKDIRECTORY;                                 05905150
          UNLOCKDIRECTORY;                                 05905200
          UNLOCKDIRECTORY;                                 05905300
          UNLOCKDIRECTORY;                                 05905400
          UNLOCKDIRECTORY;                                 05905500
          UNLOCKDIRECTORY;                                 05905600
          UNLOCKDIRECTORY;                                 05905700
          UNLOCKDIRECTORY;                                 05905800
          UNLOCKDIRECTORY;                                 05905900
          UNLOCKDIRECTORY;                                 05906000
          UNLOCKDIRECTORY;                                 05906100
          UNLOCKDIRECTORY;                                 05906200
          UNLOCKDIRECTORY;                                 05906300
          UNLOCKIT;                                        05906350
          DELTA:=NABS(*P(DUP));                             05906400
          P(XIT);                                           05906500
          EXIT:: KILL([RCW] INX NOT 2);                     05906600
          UNLOCKIT;                                        05906700
        END SCHEDIO;
        PROCEDURE DKBUSINESS(BUFF); VALUE BUFF; REAL BUFF;
        BEGIN
          REAL RCW=+0,
            MID=RCW+1,
            FID=MID+1,
            TMID=FID+1,
            TFID=TMID+1,
            A=TFID+1,
            B=A+1;
          INTEGER N=B+1;
          ARRAY HD=N+1[*];
          BOOLEAN RDT=HD+1;
          INTEGER C=RDT+1,D=C+1,I=D+1,J=I+1,R=J+1,S=R+1,
            LA=S+1,SA1=NT2,
            H=NT7,K=NT6,L=NT5,G=NT4,T=NT3,Q=JUNK;
          REAL E=LA+1;
          REAL KTR=B;
          REAL TYPE=C;
          REAL WORD=D;
          REAL HA=J;
          REAL HEADER=R;
          ARRAY HDR=E[*];
          BOOLEAN FILTOG=E+1;
          REAL SEGS=FILTOG+1;

```

```

%028-05950000
%028-05950200
%028-05950400
%028-05950600
%028-05950800
05950900
05950950
05951000
%028-05951200
%028-05951400
%028-05951600
05951700
05951800
05951900
%028-05952000
05952200
05952210
05952220
05952230
05952240
05952250
05952260
05952270
05952300

```



\$ SET OMIT = SHAREDISK	05952399
ARRAY UT=HD[*]; INTEGER AVS=SEGS+1; DEFINE U=AVTABLE#;	05952400
INTEGER SLEEPER=AVS+1;	05952500
\$ POP OMIT	05952501
\$ SET OMIT = NOT(SHAREDISK)	05952505
INTEGER BB=SEGS+1,DL=BB+1,F=DL+1,SLEEPER=F+1; ARRAY U=HD[*];	05952510
DEFINE UT=U#;	05952520
\$ POP OMIT	05952521
LABEL V,W,X,Y,Z,AZ,BZ,CZ,INUSE,EXIT;	05952600
LABEL FILEID,XDFILE,CONFLICT,FOUND,MSG,FINIS;	05952620
\$ SET OMIT = NOT(SHAREDISK)	05952690
LABEL YZ;	05952700
\$ POP OMIT	05952701
REAL SUBROUTINE DECWORD;	05952705
BEGIN	05952710
STREAM(T=0;W=[WORD]);	05952715
BEGIN	05952720
SI=W; DI=LOC T; DS=8DEC;	05952725
END STREAM;	05952730
DECWORD=P;	05952735
END DECWORD;	05952740
SUBROUTINE SCAN;	05952745
BEGIN	05952750
STREAM(KTR,TYPE=0;T=0,W=[WORD]);	05952755
BEGIN	05952760
SI=KTR;	05952765
L0:    IF SC=" " THEN BEGIN SI=SI+1; GO L0; END;	05952770
IF SC="" THEN * STRING IDENTIFIER	05952775
BEGIN	05952780
SI=SI+1; DS=LIT"0";	05952785
IF SC="" THEN	05952790
BEGIN	05952795
SI=SI+1;	05952800
IF SC="" THEN DS=CHR ELSE DS=LIT" ";	05952805
DS=6LIT" ";	05952810
END ELSE	05952815
BEGIN	05952820
7(IF SC="" THEN DS=CHR ELSE DS=LIT" ");	05952825
L1:          IF SC="" THEN BEGIN SI=SI+1; GO L1; END;	05952830
SI=SI+1;	05952835
END;	05952840
GO T1;	05952845
END;	05952850
IF SC=ALPHA THEN IF SC LSS "0" THEN	05952855
BEGIN * IDENTIFIER	05952860
ID:    DS=LIT"0";	05952865
7(IF SC=ALPHA THEN DS=CHR ELSE DS=LIT" ");	05952870
L2:          IF SC=ALPHA THEN BEGIN SI=SI+1; GO L2; END;	05952875
T1:          TALLY+1;	05952880
GO EXT;	05952885
END;	05952890
IF SC=ALPHA THEN IF SC LEQ "9" THEN	05952895
BEGIN * NUMBER	05952900
SI=SI+1; TALLY+1;	05952905
7(IF SC=ALPHA THEN IF SC LSS "0" THEN	05952910
BEGIN T=TALLY; SI=SI-T; JUMP OUT TO ID; END	05952915
ELSE IF SC LEQ "9" THEN	05952920
BEGIN SI=SI+1; TALLY=TALLY+1; END);	05952925
T=TALLY; SI=SI-T; DS=T OCT;	05952930
TALLY+2;	05952935

	GO EXT;	05952940
	END;	05952945
	IF SC#"" THEN TALLY*3 ELSE TALLY*5;	05952950
	DS*7 LIT"0"; DS*CHR;	05952955
EXT:	TYPE*TALLY;	05952960
	KTR*SI;	05952965
	END STREAM;	05952970
	P(,TYPE,STD,,KTR,STD);	05952975
	END SCAN;	05952980
	SUBROUTINE MLOGIT;	05952985
	BEGIN	05952990
	S*GETSPACE(15,73,0)+2;	05952995
	STREAM(B:DATE,D*S+1);	05953000
	BEGIN	05953005
	SI*LOC DATE; DS*8 OCT; DI*DI+8;	05953010
	SI*B;	05953015
	2(63(IF SC#"" THEN DS*CHR ELSE JUMP OUT 2 TO LL));	05953020
LL:	DS*LIT"0"; DI*DI-1; B*DI;	05953025
	END STREAM;	05953030
	LA* P INX 0;	05953035
	M[S]*(LA*S) DIV 5;	05953040
	M[S+2]*IF FILTOG THEN =N ELSE SEGS;	05953045
	LINKUP(18,S);	05953050
	END MLOGIT;	05953055
	SUBROUTINE ENTERFILE;	05953060
	BEGIN	05953065
	FIXARRAY(HD,B,30);	05953070
	MOVE(30,HD-1,HD);	05953075
	HD[0]*@3600036000101;	05953080
	STREAM(,XCLOCK,H*HD INX 3);	05953085
	BEGIN	05953090
	SI*LOC DATE; DS*8OCT;	05953095
	DI*DI-20; SI*SI+4; DS*4CHR;	05953100
	DI*DI-7; SI*H; SI*SI+5; DS*3CHR;	05953105
	DI*H; DS*2LIT"+#"; SI*SI-3; DS*3CHR;	05953110
	END STREAM;	05953115
	HD[4],[42:1]:=1; % MAKE FILE NON-MOVEABLE	05953117
	HD[7]*(HD[8]+N)=(HD[9]+1);	05953120
	HD[10]*A;	05953125
	ENTERUSERFILE(MID,FID,[6:42],B=1);	05953130
	STREAM(MID,FID,N,TMID,TFID,FILTOG,	05953135
	B*IF FILTOG THEN B ELSE BUFF);	05953140
	BEGIN	05953145
	SI*LOC N; DI*LOC N; DS*8DEC;	05953150
	DI*LOC N; DS*7FILL; DI*B;	05953155
	DS*LIT" "; SI*LOC MID; SI*SI+1; DS*7CHR;	05953160
	DS*LIT"/"; SI*SI+1; DS*7CHR;	05953165
	DS*6LIT" SEGS="; DS*8CHR; DS*8LIT" CREATED";	05953170
	FILTOG(DS*6LIT" FROM "; SI*SI+1; DS*7CHR;	05953175
	DS*LIT"/"; SI*SI+1; DS*7CHR);	05953180
	DS*LIT"0";	05953185
	END STREAM;	05953190
	IF FILTOG THEN	05953195
	BEGIN	05953200
	MLOGIT;	05953205
	SPOUT(B);	05953210
	END ELSE	05953215
	FORGETSPACE(B);	05953220
	END ENTERFILE;	05953225
	P(0,0,0,0,0,BUFF,DUP); BUFF*P,[15:15]=1; P(0,0,B LSS 0);	05953350

```

P(0,0,0,0,0,0,0,0,0,0,0,0);
$ SET OMIT = NOT(SHAREDISK);
P(0,0);
$ POP OMIT
IF B.[CF]#0 THEN% MAKE RESERVE/DISK
BEGIN MID:="RESERVE"; FID:="DISK ";
  IF (A:=DIRECTORYSEARCH(=MID,FID,5))#0 THEN
  BEGIN STREAM(BUFF);
    DS:=30LIT" RESERVE/DISK ALREADY PRESENT*";
    GO TO EXIT;
  END;
  IF (A+GETUSERDISK((N+RESERVEDISKSZ)&1[2:47:1]))#0 THEN
  BEGIN STREAM(BUFF);
    DS:=32LIT"***NO USER DISK FOR RESERVE/DISK*";
    GO TO EXIT;
  END;
  GO TO CZ;
END;
IF RDT THEN
BEGIN P(B); A:=M[BUFF INX 0]; N:=M[BUFF INX 1]; END ELSE
BEGIN
SCAN;
IF TYPE=1 THEN % IDENTIFIER
BEGIN
  TMID=WORD;
  SCAN; IF WORD#"/" THEN GO EXIT;
FILEID;
  SCAN; IF NOT(TYPE=1 OR TYPE=2) THEN GO EXIT;
  TFID=IF TYPE=2 THEN DECWORD ELSE WORD;
  FILTOG=TRUE;
  SCAN;
END;
IF TYPE=2 THEN % NUMBER
BEGIN
  A=WORD;
  SCAN;
  IF TYPE=3 THEN IF WORD#"/" THEN
  BEGIN
    WORD=A;
    A=0;
    TMID=DECWORD;
    GO FILEID;
  END ELSE SCAN;
  IF TYPE=2 THEN N=WORD;
END;
END;
SEGS=N+N+(N#0);
IF A#0 THEN
BEGIN
STREAM(A,DI=[FID]);
BEGIN SI:=LOC A; DS:=8 DEC; END;
IF (J:=A DIV 100000) GEQ NEUP,NEUF OR A LSS DIRECTORYTOP+4 THEN
VI BEGIN STREAM(FID,BUFF);
  BEGIN DS:=22LIT" INVALID DISK ADDRESS ";
    SI:=LOC FID; DS:=8CHR; DS:=LIT"+";
    DI:=DI-9; DS:=7 FILL;
  END;
  GO TO EXIT;
END;
IF WAITIO([FID]INX@100000000,@64,18+FID.[5:1]),[42:1] THEN GO TO V;

```

```

05953360
05953369
05953370
05953371
%028-05953400
%028-05953600
%028-05953800
%028-05954000
%028-05954200
%028-05954400
%028-05954600
05954800
%028-05955000
%028-05955200
%028-05955400
%028-05955600
%028-05955800
%028-05956000
05956250
05956300
05956350
05956400
05956450
05956500
05956550
05956600
05956650
05956700
05956750
05956800
05956850
05956900
05956950
05957000
05957050
05957100
05957150
05957200
05957250
05957300
05957350
05957400
05957450
05957500
05957550
05957600
05957650
05957700
05957750
%028-05958600
%028-05958800
05959000
%028-05959200
%028-05959400
%028-05959600
%028-05959800
%028-05960000
%028-05960200
%028-05960400
05960600

```

```

IF (R1=FID.[12:6]) GEQ 2 THEN % CHECK FOR 40 MIL ADDRESS 05960650
    IF NOT WAITIO([FID]INX @140000000,@64,18+FID.[5:1]),[43:1] 05960660
        THEN GO TO V ELSE IF R GEQ 4 THEN GO TO V;% INV ADD 05960670
END; 05960675
IF FILTOG THEN GO XDFILE; 05960680
IF A=0 THEN GO EXIT; 05960685
SLEEP([TOGGLE],USERDISKMASK); LOCKTOG(USERDISKMASK); 05960700
$ SET OMIT = NOT(SHAREDISK) 05960705
FIXARRAY(U,R,AVS); DISKWAIT(-R,-AVS,USERDISKBOTTOM); 05960710
$ POP OMIT 05960711
J+J+1 05960800
BZ: DI=(I:(E:=U[J]),STARTWRD) MOD 30; %028-05961000
$ SET OMIT = NOT(SHAREDISK) 05961005
IF (S*K+(C+E AND NUMENTM) + L+I) ≥ (F+AVS) THEN 05961010
    BEGIN 05961020
        P((DL + IF I<30 THEN 0 ELSE IF (U[0] AND NUMENTM) ≤ 30 05961030
            THEN 30 ELSE 60)+D,DUP,C+,DUP,DUP); 05961040
        IF (F + P = P MOD 30 + 30) ≥ AVS THEN 05961050
            BEGIN USERDISKSPECIALCASE(0,R,U,F&DL[CTF]); U=FLAG(P);END; 05961060
        IF I+F>AVTMAX+D+DL THEN GO Y; 05961070
        IF P(DL≠0,DUP) THEN P(UT[DL-1],XCH); 05961080
        DISKWAIT(-(R+DL),F=DL,BB+ USERDISKBOTTOM+I DIV 30); 05961085
        IF P THEN UT[DL-1] + P(XCH); K+P; L+P; 05961090
    END; 05961095
$ POP OMIT 05961099
$ SET OMIT = SHAREDISK 05961199
AVS:=30-(S:(C+E AND NUMENTM)+D)MOD 30+S; %028-05961200
FIXARRAY(UT,R,AVS); DISKWAIT(-R,AVS,BI=I DIV 30+USERDISKBOTTOM); 05961400
K:=S; LI=D; SI=I+C; %028-05961600
$ POP OMIT 05961601
GI=I-(NT2:(P(U[J-1],DUP) AND NUMENTM)+P(XCH),STARTWRD); %028-05961800
SI=U[J+1],STARTWRD=S; HI=K:=K-1; IF UT[T:=L],DEND GTR A THEN GO X; 05962000
W: IF UT[T+(H+L+1) DIV 2],DEND > A THEN IF UT[H+T-1],DEND > A THEN GO W 05962200
ELSE ELSE IF UT[T+T+1],DEND ≤ A THEN BEGIN L+T+1; GO W END; %028-05962400
X: IF A GEQ L:=(HI=UT[T],DEND)-(Q:=UT[T],DSIZE) THEN 05962600
    IF (LAI=(A+N)) LEQ H THEN GO AZ%AREA AVAILABLE 05962700
    ELSE IF LA LEQ SA1:=(UT[T+1],DEND-UT[T+1],DSIZE) THEN 05962800
        NI=LA-A:=H ELSE NI=SA1-A:=H ELSE IF (LAI=A+N) GTR L THEN 05962900
        NI=L-A ELSE RDT:=RDT OR @100000; 05963000
    GO INUSE; 05963100
Y: TMID:=IF RDT THEN "DKTEST " ELSE "BADISK "; 05963800
$ SET OMIT = NOT(DKBNODFX AND NOT DFX) 05963809
IF FID.[5:1] THEN TMID.[42:6]:=@22; % NAME CHANGE DKB 05963810
$ POP OMIT 05963811
STREAM(TMID,FID,N,MID,B,BUFF); 05964000
BEGIN DSI=LIT " "; SII=LOC TMID; SII=SI+1; DSI=7 CHR; 05964200
    DSI=LIT "/" ; SII=SI+1; DSI=7 CHR; 05964400
    DSI=13 LIT " NOT CREATED(" ; SII=SI+8; SKIP SB; 05964500
    IF SB THEN ELSE 05964600
        BEGIN SII=LOC N; DSI=7 DEC; NI=DI; DII=DI-7; DSI=7 FILL; 05964800
            DII=N; DSI=5 LIT " SEGS"; SII=SI+1; 05964900
        END; DSI=11 LIT " IN USE BY "; DSI=7 CHR; DSI=LIT "/" ; 05965000
        SII=SI+1; DSI=7 CHR; 05965200
        DSI=2 LIT ")+"; 05965400
    END; 05965600
FORGETSPACE(R); %028-05966100
GO EXIT; 05966110
INUSE: % SEARCH THE DIRECTORY TO FIND THE NAME OF THE CONFLICTING 05966200
% FILE, SINCE USERDISK REMAINS LOCKED, DISK ALLOCATION 05966210
% CANNOT CHANGE, HENCE, THE DIRECTORY NEED NOT BE LOCKED, 05966220

```

```

FORGETSPACE(R);
FIXARRAY(UT,R,480);
FOR J:=DIRECTORYTOP+4 STEP 16 WHILE TRUE DO
BEGIN DISKWAIT(=R,480,J);
  FOR I:=14 STEP -1 UNTIL 0 DO
  BEGIN E:=UT[450+2*I];
    IF(E EQV @114)=NOT 0 THEN
    BEGIN MID:="SYSTEM "; B:=FID; GO Z; END;
    IF (E EQV @14) NEQ NOT 0 THEN
    BEGIN B:=UT[30*I+9] AND 31;
      FOR K:=1 STEP 1 UNTIL B DO
      IF (C:=UT[30*I+9+K])NEQ 0 THEN
      IF A GEQ C THEN IF A LSS
      SA1:=(C+D:=UT[30*I+8]) THEN
      BEGIN MID:=E&((LA LEQ SA1) AND
      (RDT,[18;15]))[1;47;1];
      IF A+N GTR SA1 THEN N:=SA1-A;
      B:=UT[451+2*I];
      GO TO Z;
      END;
    END;
  END;
END;
Z:
$ SET OMIT = NOT SHAREDISK
UNLOCK(USERDISKBOTTOM);
$ POP OMIT
UNLOCKTOG(USERDISKMASK);
GO TO Y;
AZ: IF A NEQ L AND LA NEQ H THEN
BEGIN IF S=0 THEN
$ SET OMIT = NOT(SHAREDISK)
BEGIN IF G=0 THEN GO YZ;
P((G+1) DIV 2);
IF BB>1 THEN
BEGIN IF D=0 THEN
YZ: BEGIN USERDISKSPECIALCASE(2,E,UT,J);
UT=FLAG(P); R=UT.[CF];BB=0; GO BZ;
END;
IF P(DUP)>D THEN P(DEL,D);
END;
S = P;
U[J],STARTWRD + G+I=S; IF BB>1 THEN G + DL+D=S;
K + G+C-1; D + IF BB>1 THEN D+DL ELSE I;
$ POP OMIT
$ SET OMIT = SHAREDISK
BEGIN IF G=0 OR D=0 THEN
BEGIN USERDISKSPECIALCASE(2,E,UT,J); GO TO BZ END;
SI=IF P((G+1) DIV 2,DUP) > D THEN P(DEL,D) ELSE P;
U[J],STARTWRD:=I-S; GI=D-S; KI=G+C-1;
$ POP OMIT
MOVE(C,[UT[D]], [UT[G]]); T:=T-S;
END;
FOR GI=K STEP -1 UNTIL T DO UT[G+1]:=UT[G];
UT[T]:=A&(A=L)[TODSIZE];
UT[T+1]:=H&(H=LA)[TODSIZE];
C:=C+1;
K = K+1;
END ELSE
IF A=L AND LA=H THEN

```

```

05966400
%028-05966600
%028-05967000
%028-05967200
%028-05967400
%028-05967600
05967800
05967900
%028-05968000
%028-05968200
%028-05968400
%028-05968600
05968800
05968900
05969000
05969100
05969150
05969200
%028-05969400
%028-05969600
%028-05969800
%028-05970000
%028-05970200
05970300
05970390
05970400
05970410
05970500
%028-05970600
05970800
%028-05971000
05971005
05971010
05971012
05971015
05971020
05971025
05971030
05971040
05971050
05971060
05971070
05971080
05971090
05971091
05971095
%028-05971200
%028-05971400
%028-05971600
%028-05971800
05971801
%028-05972000
%028-05972200
%028-05972400
%028-05972600
05972800
%028-05973000
05973100
%028-05973200
05973400

```

BEGIN C:=C-1; MOVE(K=T,[UT[T+1]],[UT[T]]); K:=K-1 END	%028-05973600
ELSE UT[T]:=(IF A=L THEN H ELSE A)&(Q=N)[TODSIZE];	%028-05973800
U[J],NUMENT:=C;	%028-05974000
IF Q=U[J],MAXSIZ THEN	%028-05974200
BEGIN Q:=UT[H:=K*C+1],DSIZE;	05974400
FOR H:=H STEP 1 UNTIL K DO	05974600
IF P(UT[H],DSIZE,DUP) GTR Q THEN Q:=P ELSE P(DEL);	%028-05974800
U[J],MAXSIZ:=Q;	%028-05975000
END;	%028-05975200
MID:=IF RDT THEN "DKTEST " ELSE "BADISK ";	05975400
\$ SET OMIT = NOT(DKBNODFX AND NOT DFX)	05975404
IF FID,[5:1] THEN MID,[42:6]:=@22; % NAME CHANGE DKB	05975405
\$ POP OMIT	05975406
\$ SET OMIT = NOT(SHAREDISK)	05975410
SCRATCHDIRECTORYENTER(A,N);	05975440
IF BB>1 THEN	05975450
BEGIN	05975460
IF DL#0 AND U[J]#E THEN DISKWAIT(R,DL,USERDISKBOTTOM);	05975500
DISKWAIT(R+DL,F=DL,BB);	05975550
END ELSE DISKWAIT(R,AVS,USERDISKBOTTOM);	05975560
UNLOCK(USERDISKBOTTOM);	05975570
\$ POP OMIT	05975571
\$ SET OMIT = SHAREDISK	05975595
DISKWAIT(R,AVS,B);	%028-05975600
\$ POP OMIT	05975601
UNLOCKTOG(USERDISKMASK);	05975610
FORGETSPACE(R);	05975620
CZ: ENTERFILE;	05975630
GO EXIT;	05975640
XDFILE:	05975700
IF (HEADER:=DIRECTORYSEARCH(TMID,NFLAG(=TFID OR M),4)) LSS 64 THEN	05975750
BEGIN	05975800
TYPE:=HEADER;	05975850
GO MSG;	05975900
END;	05975950
HA=HEADER,[FF];	05976000
HDR=[M[HEADER+HEADER INX 0]] & 30[8:38:10];	05976050
MID="BADISK ";	05976100
S=HDR[8]; % SEGMENTS PER ROW	05976150
IF A#0 THEN	05976200
BEGIN	05976250
FOR I=HDR[9] STEP =1 UNTIL 1 DO	05976300
IF (LA=HDR[I+9])#0 THEN	05976350
IF A GEQ LA AND A LSS LA+S THEN % FOUND ROW	05976400
IF A+N LEQ LA+S THEN GO FOUND ELSE GO CONFLICT;	05976450
TYPE=4;	05976500
IF FALSE THEN	05976550
BEGIN	05976600
CONFLICT: TYPE=3;	05976650
SEGS=A+N-LA-S;	05976700
END;	05976750
HEADERUNLOCK(TMID,TFID,HEADER&HA[CTF]);	05976800
GO MSG;	05976850
FOUND:	05976900
HDR[I+9]=0;	05976950
DISKWAIT(HEADER,30,HA);	05977000
IF (I+A=LA) GTR 0 THEN FORGETUSERDISK(LA,I);	05977050
IF (I+LA+S=(LA+A+N)) GTR 0 THEN FORGETUSERDISK(LA,I);	05977100
\$ SET OMIT = NOT(DKBNODFX AND NOT DFX)	05977124
IF FID,[5:1] THEN MID,[42:6]:=@22; % NAME CHANGE DKB	05977125

```

$ POP OMIT
    ENTERFILE;
    GO FINIS;
END;
N+S; SEGS+0;
FOR I+HDR[9] STEP -1 UNTIL 1 DO
    IF (A+HDR[I+9])#0 THEN
        BEGIN
            HDR[I+9]+0;
            DISKWAIT(HEADER,30,HA);
            WORD+A; FID+DECWORD;
$ SET OMIT = NOT(DKBNODFX AND NOT DFX)
            MID,[42:6];=IF FID,[5:1] THEN @22 ELSE @60; % NAME CHANGE DKB
$ POP OMIT
            ENTERFILE;
            SEGS+SEGS+N;
        END;
FINIS;
FORGETSPACE(HEADER);
P(DIRECTORYSEARCH(-TMID,TFID,6),DEL);
TYPE+5;
MSG;
STREAM(TMID,TFID,SEGS,A,TYPE,BUFF);
BEGIN
    SI+LOC SEGS; DI+LOC SEGS; DS+8DEC; DS+8DEC;
    DI+LOC SEGS; DS+8FILL; DI+LOC A; DS+8 FILL; DI+BUFF;
    DS+LIT", "; SI+LOC TMID; SI+SI+1; DS+7CHR;
    DS+LIT"/ "; SI+SI+1; DS+7CHR;
    DS+11 LIT" NOT XD=ED(";
    CI+CI+TYPE;
    GO T0; GO T1; GO T2; GO T3; GO T4; GO T5;
T0:   DS+11 LIT"NOT ON DISK"; GO EXT;
T3:   DS+8 CHR; DS+6 LIT" SEGS ";
T1:   DS+6 LIT"IN USE"; GO EXT;
T2:   DS+11 LIT"SYSTEM FILE"; GO EXT;
T4:   SI+SI+8; DS+8 CHR;
    DS+12 LIT" NOT IN FILE"; GO EXT;
T5:   DI+DI-11;
    DS+6 LIT" SEGS="; DS+8 CHR; DS+7 LIT" XD=ED=";
    TYPE+DI; DI+BUFF; DS+LIT" "; DI+TYPE; GO EXT;
EXT:  DS+2 LIT")+";
    END STREAM;
    A+1; N+SEGS; % FOR LOGGING
    GO EXIT;
EXIT:
    IF A#0 THEN
        BEGIN
            B+BUFF;
            MLOGIT;
        END;
    IF RDT THEN M[SLEEPER INX 0] :=1 ELSE SPOUT(BUFF);
END;
SAVE PROCEDURE DISKIO(LOCIOD,CORE,SIZE,DISK);%
    VALUE CORE,SIZE,DISK;%
    REAL LOCIOD;%
    INTEGER CORE,SIZE,DISK;%
    BEGIN REAL IOD, OLAYIO, FIN;
    OLAYIO := SIZE,[3:1]; SIZE,[3:1] := 0;
    IF DISK,[1:1] THEN % DRUM OR AUXMEM
        BEGIN

```

```

05977126
05977150
05977200
05977250
05977300
05977350
05977400
05977450
05977500
05977550
05977600
05977624
05977625
05977626
05977650
05977700
05977750
05977800
05977850
05977900
05977950
05978000
05978050
05978100
05978150
05978200
05978250
05978300
05978350
05978400
05978450
05978500
05978550
05978600
05978650
05978700
05978750
05978800
05978850
05978900
05978950
05979000
05979050
05979100
05979310
05979320
05979330
05979340
05979350
05979360
05979400
%028-05979600
06000000
06001000
06002000
06003000
06004000
06004010
06005000
06007000

```

```

$ SET OMIT = NOT(AUXMEM)                                06007099
  NT1 := 0;                                              06007100
  IOD := CORE,[CF] & DISK[CTF] & CORE[2:1:1] &        06007200
  (SIZE INX 1)[8:38:10] &                               06007300
  ((SIZE := DISK.[32:1])+1)[4:46:2]                    06007400
$ POP OMIT                                              06007401
$ SET OMIT = AUXMEM                                     06007402
  BYBY("INVALID AUXMEM I/O+",19);                       06007450
$ POP OMIT                                              06007499
  END                                                    06007500
  ELSE                                                  06011000
    BEGIN                                              06011100
      IOD := ABS(CORE) & SIZE[8:38:10] &              06011200
      ((NT1:=SIZE) INX 29) DIV 30 + @1000)[CTF] &      06011300
      CORE[24:1:1] & 3[5:46:2];                       06012000
$ SET OMIT = NOT(SHAREDISK)                             06012499
  DISK.[1:1]:=SIZE.[1:1];                               06012500
$ POP OMIT                                              06012501
  STREAM(DISK,D:=CORE,[CF]);                            06013000
  BEGIN SI ← LOC DISK; DS ← 8 DEC END;%                06014000
  SIZE ← 2;%                                            06015000
  END;%                                                 06016000
  FIN:=IF OLAYIO THEN IOD&DISK[CTC]&DISK[8:21:12] ELSE IOD; 06016100
  % ACTUAL DISK ADDRESS IN FINALQUE FOR OLAY I/O=S     06016200
  IOREQUEST(NABS(FIN)&@377[25:40:8],IOD,[LOCIOD]&%     06017000
  NT1[10:2:1]&                                          06017500
  (SIZE+16)[12:42:6]&OLAYIO[9:47:1]);                 06018000
  LOCIOD ← 0;%                                         06019000
  END DISKIO;%                                         06020000
PROCEDURE FORGETESPDISK(S); VALUE S; REAL S; FORWARD;  06020500
REAL PROCEDURE GETESPDISK;%                             06021000
  BEGIN REAL T=NT1;                                    06022000
  IF ESPCOUNT=0 THEN                                  06022100
  BEGIN                                                06022200
    STREAM(D:=T:=SPACE(2));                             06022300
    DS:=12 LIT"NO ESPDISK";                             06022400
    SPOUT(T);                                           06022500
    SLEEP([ESPCOUNT],NOT 0);                            06022600
  END;                                                  06022700
  STREAM(T←0,A←ESPTAB[X←0]);                             06023000
  BEGIN SI←A;                                           06024000
  L1: IF SC="" THEN BEGIN SI←SI+1; GO TO L1 END;        06025000
    A←SI; DI←A;                                         06026000
  L2: IF SB THEN                                        06027000
    BEGIN TALLY←TALLY+1; SKIP SB; SKIP DB; GO TO L2 END; 06028000
    T←TALLY; DS←SET;                                    06029000
  END;                                                  06030000
  GETESPDISK:=((P(DUP),[CF]=ESPTAB)×8                 06031000
  +P(XCH),[30:3])×6+P+ESPDISKBOTTOM;                   06032000
  ESPCOUNT:=ESPCOUNT-1;                               06033000
  END;                                                  06035000
PROCEDURE FORGETESPDISK(SEGMENT); VALUE SEGMENT; REAL SEGMENT;% 06036000
  BEGIN REAL S,T;                                       06037000
  IF SEGMENT LSS ESPDISKBOTTOM OR                     06037100
  SEGMENT GTR ESPDISKTOP THEN                          06037200
  BEGIN S ← FLAG("ESPDISK"); T ← FLAG("ERROR,+");     06037300
    S ← WAITIO([SEGMENT] INX 2,0,25);                   06037400
    SLEEP(0,0);                                         06037500
  END;                                                  06037600
  T:=(S:=(T:=SEGMENT-ESPDISKBOTTOM) DIV 6)×6-T;      06037700

```



```

S+S,[30:15]&S[30:45:3]+ESPTAB; 06038000
STREAM(T,S); BEGIN SKIP T DB; DS+RESET END; 06038100
ESPCOUNT+ESPCOUNT+1; 06038200
END;% 06045000
$ SET OMIT = NOT(DEBUGGING) 06045999
PROCEDURE DISKBUG;% 06046000
BEGIN REAL T;% 06047000
;STREAM(T+P(,DBADR),DBARRAY); 06048000
BEGIN DS := 20 LIT "DISKBUG... TRY +"; DI := DI+4; 06048100
SI := LOC T; SI := SI+6; SKIP 3 SB; 06048200
3(DS := 3 RESET; 06048300
3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB)); 06048400
END STREAMING; 06048500
T + WAITIO(DBARRAY,[33:15],@377,25);% 06049000
DDT;% 06050000
WHILE DBADR # 0 DO% 06051000
BEGIN DISKWAIT(=(CDBARRAY INX 1)&DBADR[1:1:1]), 06052000
30, ABS(DBADR)); 06052100
DDT;% 06054000
END;% 06055000
END;% 06056000
$ POP OMIT 06056001
$ SET OMIT = NOT(DFX OR SHAREDISK) 06056099
REAL LASTEU; COMMENT USED FOR ALLOCATING DISK FROM DIFFERENT EUS, %DFX06056100
IF POSSIBLE; %DFX06056200
$ POP OMIT 06056201
$ SET OMIT = NOT(SHAREDISK) 06057000
COMMENT CLEANOUT PERFORMS THE FOLLOWING TASKS: 06057020
1, REMOVES ALL CONTENTION BITS THAT WERE SET BY "SYS", 06057040
2, UNLOCKS ALL ADDRESSES THAT WERE LOCKED BY "SYS", 06057060
3, RETURNS ALL OF THE DISKSPACE THAT IS IN THE SCRATCHDIRECTORY 06057080
OF THE SYSTEM THAT IS BEING CLEARED, 06057100
4, REMOVES ALL FILES THAT WERE BEING LOADED BY THE SYSTEM THAT 06057120
IS BEING CLEARED, 06057140
5, CLOSES ALL FILES THAT THE OFFENDING SYSTEM HAD OPENED, 06057160
6, REMOVES ALL ENTRIES IN THE HOLDLIST THAT WERE MADE BY THE 06057180
SYSTEM BEING CLEARED, 06057200
7, WAKES UP ALL PROCESSES IN OTHER SYSTEMS THAT WERE WAITING 06057220
FOR A FILE THAT WAS IN USE (BY ANY SYSTEM), 06057240
CLEANOUT IS CALLED WHEN THE KEYBOARD MESSAGE "CLSYN" 06057260
IS ENTERED AND WHEN A SYSTEM IS HALT/LOADED AND 06057280
THERE ARE OTHER SYSTEMS RUNNING. 06057300
END COMMENT; 06057320
PROCEDURE CLEANOUT(SYS); 06057340
VALUE SYS; REAL SYS; 06057360
BEGIN 06057380
REAL I,J,K,F,N,B; 06057400
REAL T,T1,FOURMASK,NINEMASK; 06057420
ARRAY NB[*],FH[*],BP[*]; 06057440
REAL KLUDGE,HOLDER,NEXTSLOT,BYPASS; 06057450
LABEL AGAIN,FOUND,ZOTIT,CLOSEIT,QUIT,FM,NM; 06057460
IF SYS#SYSNO THEN % CL = LET THE OPTR 06057462
BEGIN STREAM(S:=SYS+17, DI=N:=SPACE(5)); % KNOW THAT WE ARE 06057464
BEGIN DS:=17 LIT " CLEARING SYSTEM "; % WORKING ON IT, 06057466
SI:=LOC S; SI:=SI+7; DS:=CHR; 06057468
DS:=15 LIT " - PLEASE WAIT-"; 06057470
END; 06057472
SPOUT(N); 06057474
END; 06057476
T:=@4060&SYS[30:46:2]; % CLEAR ALL CONTENTION BITS 06057480

```

```

P(WAIT10([T] INX @100000000,0,18),DEL); 06057500
T,[3711];=1; % UNLOCK ALL ADDRESSES 06057520
P(WAIT10([T] INX @100000000,0,18),DEL); 06057540
NB:=[M[N:=SPACE(480)]]&480[8:38:10]; 06057560
FH:=[M[F:=N+33]]&31[8:38:10]; 06057580
DISKWAIT(-F,-30,B+DIRECTORYTOP+1); 06057600
IF (I:=FH[SYS],SLINK) NEQ 0 THEN 06057620
  BEGIN %CLEAN OUT SCRATCH DIRECTORY 06057640
    DISKWAIT(-F,30,I) ; 06057660
    FH[1]:=J:=0 ; 06057680
    DO IF J:=NOT J THEN SCRATCHCLEAN(FH,N=1,I) 06057700
      ELSE SCRATCHCLEAN(NB,F=1,I) 06057720
    UNTIL I=0 ; 06057740
    DISKWAIT(-F,30,B); FH[SYS]=0; 06057750
  END ; 06057760
DISKWAIT(F,-30,B); 06057770
KI:=SYS=1; % LOD IS 1 WORD PAST LITC UNLESS IT IS 06057780
FOURMASK:=P(,FM,K,+,LOD); % LAST SYL, IN WHICH CASE C IS COUNTED 06057790
NINEMASK:=P(,NM,K,+,LOD); % UP WHILE THE SUM POPS FROM B TO A, 06057800
FOR I:=DIRECTORYTOP+4 STEP 16 WHILE TRUE DO 06057820
  BEGIN 06057840
    DISKWAIT(-N,480,I); 06057860
    FOR J:=478 STEP -2 UNTIL 450 DO 06057880
      IF (NB[J] EQV @114)=NOT 0 THEN GO QUIT ELSE 06057900
      IF (NB[J] EQV @14) NEQ NOT 0 THEN 06057920
      BEGIN 06057940
        F:=15*(J-450)+4; % NB[F] POINTS TO HEADER[4] 06057960
        IF NB[F],[1:1] AND NB[F],[4:2]=SYS AND NB[F],[12:4]=0 THEN 06057980
        BEGIN % FILE MUST BE REMOVED FROM DIRECTORY 06058000
          BP:=[M[B:=SPACE(60)]]&60[8:38:10]; 06058060
          DISKWAIT(-B,-60,T1:=T:=SCRAMBLE(NB[J],NB[J+1])); 06058080
          FOR K:=0 STEP 3 UNTIL 57 DO 06058100
            IF (BP[K] EQV NB[J])=NOT 0 THEN 06058120
            IF (BP[K+1] EQV NB[J+1])=NOT 0 THEN GO TO FOUND; 06058140
            IF (T:=BP[2],[FF])#0 THEN 06058160
            BEGIN DISKWAIT(-B,60,T); 06058180
              GO AGAIN; 06058200
            END; 06058220
            STREAM(SI:=NB[J], DI:=T:=SPACE(10)); 06058225
            BEGIN SI:=S; SI:=SI+1; DS:=LIT" "; DS:=7 CHR; 06058230
              DS:=LIT"/"; SI:=SI+1; DS:=7 CHR; 06058235
              DS:=34 LIT" NOT IN BYPASS, SHOULD COOL/START"; 06058240
            END; 06058245
            SPOUT(T); % TELL OPERATOR WHATS WRONG AND 06058250
            GO ZOTIT; % THEN REMOVE FILE AS PLANNED, 06058255
          FM: @ 4004370000000000, @ 4002007600000000, 06058260
            @ 4001000174000000, @ 4000400003700000; 06058265
          NM: @ 210760000000000000, @ 104017400000000000, 06058270
            @ 420003700000000000, @ 2100000760000000; 06058275
          FOUND: BP[K]:=@14; 06058280
          DISKWAIT(B,60,T); 06058300
          ZOTIT: DISKWAIT(-([HOLDER],[CF]),-3,DIRECTORYSEG); 06058320
            P(NB[449]); 06058340
            DISKWAIT(-([T:=NB[450]],[CF]),-30,I+15); 06058360
            NB[J]=@14; NB[J+1]=NEXTSLOT; NEXTSLOT=(J-450)/2+1; 06058380
            DISKWAIT(T,-30,I+15); 06058400
            DISKWAIT([HOLDER],[CF],-3,DIRECTORYSEG); 06058420
            UNLOCK(T1); 06058440
            P([NB[449]],STD); 06058460
            FORGETSPACE(B); 06058480

```

```

        IF NB[F],[2:1] THEN % FILE WAS BEING LOADED FROM TAPE06058500
        FOR K:=F+6+NB[F+5],[43:5] STEP =1 UNTIL F+6 DO 06058520
            IF NB[K]#0 THEN FORGETUSERDISK(NB[K],NB[F+4]); 06058540
        END ELSE 06058560
        IF NB[F],[4:2]=SYS AND NB[F],[44:1] THEN 06058562
        BEGIN % START LIBMAIN FOR ZEROING 06058564
            STREAM(A:=NB[J],B:=NB[J+1],T:=T:=GETSPACE(10,64,0)+4); 06058566
            BEGIN DS:=10LIT"CC REMOVE "; SI:=LOC A; SI:=SI+1; 06058568
                DS:=7CHR; DS:=LIT"/"; SI:=LOC B; SI:=SI+1; 06058570
                DS:=7CHR; DS:=6LIT";END,+"; 06058572
            END; 06058574
            CCARD(T&31[2:42:6]&1[8:47:1]); 06058576
        END ELSE 06058580
        BEGIN % CLOSE ALL OPEN FILES 06058584
            IF NB[F],[2:1] THEN 06058600
            IF NB[F],[4:2]=SYS THEN GO CLOSEIT; 06058620
            IF (NB[F] AND FOURMASK)#0 THEN GO CLOSEIT; 06058630
            IF (NB[F+5] AND NINEMASK)#0 THEN 06058640
            BEGIN 06058650
                DISKWAIT(=(T:=SPACE(30)),=-30,K:=(J-450)/2+1); 06058660
                IF (M[T+4]!=(+P(DUP)) AND NOT FOURMASK),[4:2] 06058670
                    = SYS THEN M[T+4],[2:1]:=0; 06058680
                M[T+9]!=(+P(DUP)) AND NOT NINEMASK; 06058690
                DISKWAIT(T,-30,K); 06058700
                FORGETSPACE(T); 06058705
            END; 06058710
            IF SYS=SYSNO THEN 06058720
            BEGIN 06058740
                PBCOUNT+PBCOUNT+(((NB[J] EQV "PBD ")=NOT 0 06058760
                    OR (NB[J] EQV "PUD ")=NOT 0) 06058770
                    AND NB[J+1],[CF]=1); 06058780
            END; 06058840
        END; 06058860
    END; 06058880
END; 06058920
QUIT: 06059000
DISKWAIT(=[HOLDER],[CF]),=3,DIRECTORYSEG); 06059020
IF (I:=HOLDER,[FF])#0 THEN% REMOVE ALL ENTRIES FOR THIS SYSTEM 06059040
BEGIN % AND WAKE UP ALL OTHERS 06059060
    DISKWAIT(=N,I,HOLDER,[CF]); 06059080
    FOR J:=0 STEP 1 UNTIL I=1 DO 06059100
        IF NB[J],[2:2]=SYS THEN 06059120
        BEGIN % REMOVE ENTRY FOR SYSTEM BEING CLEARED 06059140
            MOVE(I=J-1,[NB[J+1]],,[NB[J]]); 06059160
            I:=I-1; 06059180
            J=J-1; 06059190
        END ELSE 06059200
        IF NB[J],[2:2]#SYSNO THEN NB[J]:=P(DUP,LOD,SSN) ELSE 06059220
        IF NB[J] GEQ FENCE THEN BRINGBACK(NB[J],[10:8]) ELSE 06059230
            M[NB[J],[FF]]:=1; 06059240
        IF I#0 THEN DISKWAIT(N,I,HOLDER,[CF]); 06059260
        HOLDER,[FF]:=I; 06059280
    END; 06059300
% SET OMIT = NOT STATISTICS OR OMIT 06059305
    BYPASSBOTTOM:=BYPASS,[CF]; 06059310
% POP OMIT 06059315
    DISKWAIT([HOLDER],[CF]),=3,DIRECTORYSEG); 06059320
    IF SYS=SYSNO THEN 06059340
    BEGIN 06059360
        IF PBCOUNT#0 AND NOT AUTOPRINT THEN 06059400

```

```

BEGIN
    STREAM(PBCOUNT,DI=N);
    BEGIN DS:=11 LIT" THERE ARE"; DI:=DI; SI:=LOC PBCOUNT;
        DS:=4 DEC; DS:=18 LIT" PB FILES ON DISK";
        DI:=DI; DS:=3 FILL
    END;
    SPOUT(N);
END ELSE FORGETSPACE(N);
STREAM(N:=N:=SPACE(5)); DS:=19LIT"CLEANOUT COMPLETED";
SPOUT(N);
END ELSE
BEGIN
    DISKWAIT(-N,-30,0);
    NB[I:=13+5*SYS]:=NB[I+1]:=0;
    DISKWAIT(N,-30,0);
    STREAM(S:=SYS+17,N);
    BEGIN DS:=18 LIT"#SYSTEM  CLEARED"; DI:=DI-10;
        SI:=LOC S; SI:=SI+7; DS:=CHR;
    END;
    SPOUT(N);
END;
END; % OF PROCEDURE CLEANOUT
$ POP OMIT
SAVE PROCEDURE DISKWAIT(CORE,SIZE,DISK);
VALUE CORE,SIZE,DISK;
REAL CORE,SIZE,DISK;
BEGIN REAL T;
    DISKIO(T,(ABS(CORE)-1)&CORE[1:1:1],SIZE,DISK);
    SLEEP([T],IOMASK);
END;
PROCEDURE DISKSQUASH(BUFF);
VALUE BUFF; REAL BUFF;
BEGIN
REAL B, E, F, R, HI, LO,
    CNT, USE, TOG, IOD;
REAL T, SUM=T;
REAL A1, A2, A3, A4, A5; % ARRAY VARIABLES
REAL X1, X2, X3, X4, X5; % SCRATCH VARIABLES
REAL LOCIOD=X4, HICNT=X4, LSTCNT=X5;
BOOLEAN CONFLICT, PASSTWO, EUNOTSQUASHED,
    FILEOK, SQALL;
INTEGER C, D, I, S, EU, AV,
    AVSIZE, DISKAV, SQSIZE;
ARRAY UT[*], MV[*], DIR[*], EUS[*];
REAL PRTADDR, PRTVALUE;
$ SET OMIT = NOT SHAREDISK
ARRAY U[*];
REAL R1;
$ POP OMIT
LABEL SCAN, SPOUTERR, CK, OKINUSE, NOTOK, OKBOUNDS, MVEMORE, MVE,
    ENDMVE, AGAIN, OK, NEXT, SQIT, STOPSQ, STOPIT, SDXIT, OUT, FIXMV;
DEFINE
$ SET OMIT = SHAREDISK
    U          = AVTABLE#,
$ POP OMIT
    LINK       = [12:10]#,
    ASIZE      = [3:19]#,
    LOCKED     = [2:1]#,
    FACTOR     = 10000#,
    MINSIZE    = 10#,

```

```

06059420
06059440
06059460
06059480
06059500
06059520
06059540
06059560
06059562
06059564
06059580
06059600
06059620
06059640
06059660
06059680
06059700
06059720
06059740
06059760
06059800
06059820
06060001
06061500
06062000
06063000
06064000
06065000
06066000
06067000
06068000
06068100
06068200
06068300
06068400
06068500
06068600
06068700
06068800
06068900
06069000
06069100
06069200
06069300
06069400
06069500
06069600
06069700
06069800
06069900
06070000
06070100
06070200
06070300
06070400
06070500
06070600
06070700
06070800
06070900

```

```

MAXMVSZ = 900#,
KEYINMASK = [18:15]#;
COMMENT
FACTOR: THE MAXIMUM SEPARATION, IN SEGMENTS, ALLOWED
        BETWEEN TWO AVAILABLE AREAS WHICH ARE TO BE
        SQUASHED, IN GENERAL, FACTOR SHOULD NOT BE MADE
        LARGER THAN THE CAPACITY OF A 20 ML SUBMOD, I.E.,
        10,000 SEGMENTS,
MINSIZE: THE MINIMUM SIZE, IN SEGMENTS, ALLOWED FOR AN
        AVAILABLE AREA TO BE CONSIDERED AS A CANDIDATE
        FOR SQUASHING, MINSIZE MAY BE MADE AS SMALL AS
        ONE, BUT AS SQUASH TIME VARIES INVERSLY WITH
        MINSIZE, SMALLER VALUES WILL INCREASE SQUASH-
        ING TIME PROPORTIONALLY, MINSIZE LIMITA-
        TIONS MAY BE OVERRIDEN BY THE LOOKAHEAD
        FACILITY,
MAXMVSZ: LIMITS THE NUMBER OF INDIVIDUAL AREAS IN AN
        IN-USE AREA TO BE AT MOST MAXMVSZ/3 AREAS
        FOR SQUASHING TO OCCUR.
NOTE:
      1) MAXMVSZ MUST BE LESS THAN 1024,
      2) MAXMVSZ MUST BE A MULTIPLE OF 3,

```

```

DEFINE CELL = M[PRADDR]#,
STOP = M[PRADDR]#,
STOPCK = IF M[PRADDR] THEN GO STOPSQ#,
MOVEABLE = NOT DIR[X3+4],[42:1]#,
TEMPDSK = MV[I+2],[1:1]#;
SUBROUTINE SQUASHMESS;
BEGIN
  IF (X1:=P(XCH))>1 THEN X3:=IF SQSIZE#0 THEN SQSIZE
  ELSE EUS[EU-1],DSIZE;
  STREAM(A:=EU-1,B:=X1,C:=X3,C1:=0,C2:=0,CX:=0,
        NOSQ:=EUNOTSQUASHED, X2:=X2:=SPACE(10));
  BEGIN
    C1:=C1; GO TO L0;
    S1:=LOC A; DS:=4 LIT" EU "; DS:=2 DEC;
    A:=D1; D1:=D1-2; DS:=FILL; D1:=A; C1:=CX;
  L0: C2:=C1; GO TO L2; DS:=4 LIT"NULL"; C1:=CX;
  L1: DS:=7 LIT" SQUASH"; C1:=CX;
  L2: C1:=C1+B;
    GO TO LL0; GO TO LL0; GO TO LL2; GO TO LL2;
  LL0: CX:=C1; C1:=C1;
    B(NOSQ(DS:=LIT" "; CX:=C1; C1:=C2));
    CX:=C1; GO TO L1;
    B(NOSQ(JUMP OUT 2 TO LL1)); DS:=2 LIT"ED";
    JUMP OUT TO LL1;
    DS:=3 LIT"ING";
  LL1: GO TO EXT;
  LL2: DS:=LIT" "; CX:=C1; C1:=C2;
    CX:=C1; GO TO L1;
    S1:=B; 2(S1:=S1-8); B:=S1;
    B(CX:=C1; C1:=C1);
    DS:=2 LIT" ("; S1:=LOC C;
    DS:=6 DEC; C:=D1; D1:=D1-6; DS:=5 FILL; D1:=C;
    DS:=19 LIT" SEGMENTS AVAILABLE";
    B(JUMP OUT TO LL3); DS:=4 LIT" ON ";
    CX:=C1; C1:=C1;
  LL3: DS:=LIT")";
  EXT: DS:=LIT" ";
  END;

```

```

06071000
06071100
06071200
06071300
06071400
06071500
06071600
06071700
06071800
06071900
06072000
06072100
06072200
06072300
06072400
06072500
06072600
06072700
06072800
06072900
06073000
06073100
06073200
06073300
06073400
06073500
06073600
06073700
06073800
06073900
06074000
06074100
06074200
06074300
06074400
06074500
06074600
06074700
06074800
06074900
06075000
06075100
06075200
06075300
06075400
06075500
06075600
06075700
06075800
06075900
06076000
06076100
06076200
06076300
06076400
06076500
06076600
06076700
06076800
06076900

```

```

      SPOUT(X2);
END PRINTING MESSAGES;
SUBROUTINE SCANMESSAGE;
BEGIN
  X1:=(X5:=NEUP,[FF])=1; X2:=BUFF,[30:18];
  FIXARRAY(EUS,A5,X5);
  MOVE(X5,A5=1,A5);
  X5:=1; % WILL BE GEQ ZERO AFTER FIRST PASS THRU SCAN
SCAN:
  STREAM(A:=0,SIZ:=0,EU1:=-1,EU2:=-1,ERRTOG:=0,INO:=0,
        B:=X5<0,EU:=@2564000000000000,CX:=0,C1:=0,
        C2:=0,KTR:=X2);
  BEGIN
    C1:=C1; GO TO L2;
    IF SC<0 THEN
  A0: BEGIN TALLY:=1; NO:=TALLY; C1:=CX END;
    IF SC=12 THEN GO TO A0;
    DI:=LOC SIZ;
  L1: IF SC GEQ 0 THEN IF SC<12 THEN
      BEGIN
        TALLY:=TALLY+1;
        SI:=SI+1;
        GO TO L1;
      END;
    NO:=TALLY;
    SI:=SI-NO;
    DS:=NO OCT;
    TALLY:=0; NO:=TALLY;
    C1:=CX;
  L2: C2:=C1; GO TO STR;
    TALLY:=1; DI:=LOC EU;
    IF 2 SC=DC THEN % AN EU SPECIFIED
      BEGIN
        CX:=C1; GO TO L3;
        IF SC GEQ 0 THEN IF SC<12 THEN
          BEGIN
            SI:=SI+1; DI:=LOC EU1;
            IF SC GEQ 0 THEN IF SC<12 THEN
              TALLY:=2 ELSE GO TO A1;
            SI:=SI-1; NO:=TALLY;
            DS:=NO OCT; TALLY:=0;
          END ELSE GO TO A1;
        END;
        NO:=TALLY; C1:=A;
        C1:=A;
  L3: IF SC=" " THEN BEGIN SI:=SI+1; GO TO L3 END; C1:=CX;
  STR: SI:=KTR; C1:=C1+B; GO TO L5; GO TO L4;
  L4: IF SC="+" THEN GO TO EXT;
    CX:=C1; C1:=C1; % SIZE CHECK
    NO(JUMP OUT TO L5);
    CX:=C1; GO TO L3;
    IF SC#"+" THEN
  A1: GO TO ERR;
    GO EXT;
  L5: A1:=C1; C1:=C2; % EU CHECK
    NO(JUMP OUT TO ERR);
    IF SC="=" THEN
      BEGIN
        SI:=SI+1; CX:=C1; GO TO L3;
        CX:=C1; C1:=C1; % SIZE CHECK

```

```

06077000
06077100
06077200
06077300
06077400
06077500
06077600
06077700
06077800
06077900
06078000
06078100
06078200
06078300
06078400
06078500
06078600
06078700
06078800
06078900
06079000
06079100
06079200
06079300
06079400
06079500
06079600
06079700
06079800
06079900
06080000
06080100
06080200
06080300
06080400
06080500
06080600
06080700
06080800
06080900
06081000
06081100
06081200
06081300
06081400
06081500
06081600
06081700
06081800
06081900
06082000
06082100
06082200
06082300
06082400
06082500
06082600
06082700
06082800
06082900

```

NO(JUMP OUT TO L6); GO TO L7;	06083000
L6: TALLY:=EU1; EU2:=TALLY;	06083100
A:=C1; C1:=C2; % EU CHECK	06083200
NO(JUMP OUT TO ERR);	06083300
END;	06083400
L7: A:=TALLY; % ZERO OUT A	06083500
IF SC="+" THEN GO TO EXT;	06083600
IF SC="," THEN	06083700
BEGIN S1:=S1+1; A:=S1; GO EXT END;	06083800
ERR: TALLY:=1; ERRTOG:=TALLY;	06083900
EXT:	06084000
END;	06084100
IF P THEN % ERROR IN INPUT MESSAGE	06084200
BEGIN	06084300
SPOUTERR:	06084400
SPOUT(BUFF,[15:15]=1);	06084500
FORGETSPACE(A5);	06084600
P(XIT);	06084700
END;	06084800
IF (X3:=P) GEQ 0 THEN % AN EU RANGE SPECIFIED,	06084900
BEGIN	06085000
IF (X4:=P)>X1 OR X3>X1 THEN GO SPOUTERR;	06085100
FOR I:=X3 STEP 1 UNTIL X4 DO EUS[I]:=1;	06085200
P(DEL); GO CK;	06085300
END;	06085400
X5:=P(XCH); % SIZE OF SQUASH	06085500
IF (X4:=P) GEQ 0 THEN IF X4>X1 THEN GO SPOUTERR ELSE	06085600
EUS[X4]:=1&X5(TODSIZE) ELSE IF X5=0 THEN SQALL:=1	06085700
ELSE SQSIZE:=X5;	06085800
CK: IF (X2:=P)≠0 THEN GO SCAN; % NOT FINISHED YET	06085900
END SCANNING INPUT MESSAGE;	06086000
SUBROUTINE FIXANDWRITEHEADER;	06086100
BEGIN	06086200
M[A4+9+X2,[28:5]]:=C;	06086300
DISKWAIT(A4,30,X2,[CF]);	06086400
END WRITING NEW HEADER;	06086500
SUBROUTINE BOUNDARYCK;	06086600
BEGIN	06086700
LSTCNT:=0; M[A2=1]:=-1;	06086800
MVEMORE:	06086900
X3:=HICNT:=0; STOPCK;	06087000
FOR I:=CNT STEP -3 UNTIL 0 DO	06087100
IF P(MV[I],DUP),DEND>X3 AND P(XCH)>0 THEN	06087200
BEGIN X3:=MV[I],DEND; HICNT:=I END;	06087300
IF X3=0 THEN % RE=ORDERING OF MV ARRAY COMPLETE	06087400
BEGIN	06087500
MV[LSTCNT+2],LINK:=@1777;	06087600
GO OKBOUNDS;	06087700
END;	06087800
IF M[A2=1]<0 THEN M[A2=1]:=HICNT ELSE MV[LSTCNT+2],LINK:=HICNT;	06087900
MV[LSTCNT:=HICNT]:=NABS(*P(DUP));	06088000
MV[HICNT+1],[2:26]:=HI;	06088100
HI:=HI-(X3:=MV[HICNT],DSIZE);	06088200
IF X3 LEQ UT[AV+1],ASIZE THEN	06088300
OK: BEGIN	06088400
MV[HICNT+2]:=0;	06088500
GO MVEMORE;	06088600
END ELSE	06088700
BEGIN % LOOKING FOR TEMPORARY STORAGE	06088800
FOR I:=S-2 STEP -1 UNTIL D DO	06088900

IF X3 LEQ UT[I],ASIZE THEN	06089000
IF NOT UT[I],LOCKED THEN        % OK FOR TEMP STORAGE	06089100
BEGIN	06089200
MV[HICNT+2]:=UT[I].DEND&[[2:38:10];	06089300
GO MVEMORE;	06089400
END;	06089500
END;	06089600
IF PASSTWO THEN % NON-PROTECTED FILE TRANSFER	06089700
BEGIN	06089800
DISKWAIT(=A4.30,MV[HICNT+2],[CF]);	06089900
STREAM(A:=[M[A4+MV[HICNT+2],[FF]]],X2:=X2:=SPACE(6));	06090000
BEGIN	06090100
DS:=27 LIT" #FILE INTEGRITY CONFLICT: "; SI:=A;	06090200
SI:=SI+1; DS:=7 CHR; DS:=LIT"/"; SI:=SI+1;	06090300
DS:=7 CHR; DS:=LIT"@";	06090400
END;	06090500
SPOUT(X2); CELL,KEYINMASK:=7;	06090600
SLEEP((PRTADDR INX M),@77777); STOPCK;	06090700
IF CELL=2 THEN BEGIN CELL:=0&1[CTF]; GO TO OK END;	06090800
END ELSE CONFLICT:=TRUE;	06090900
TOG:=0;	06091000
OKBOUNDS;	06091100
END BOUNDARY AND CONFLICT CHECKING;	06091200
BOOLEAN SUBROUTINE INUSEOK;	06091300
BEGIN	06091400
UT[AV+1],[1:1]:=NOT PASSTWO; TOG:=1; CNT:=0;	06091500
FOR X1:=DIRECTORYTOP+4 STEP 16 WHILE TRUE DO	06091600
BEGIN STOPCK;	06091700
DISKWAIT(=A1.480,X1);	06091800
FOR I:=14 STEP -1 UNTIL 0 DO	06091900
BEGIN STOPCK;	06092000
IF ((E:=DIR[450+P(I,DUP,+)]) EQV @114)≠NOT 0 THEN	06092100
GO TO NOTOK;	06092200
IF (E EQV @14)≠ NOT 0 THEN	06092300
BEGIN FILEOK:=FALSE; % INITIATE STATUS CHECKING	06092400
B:=DIR[(X3:=30×I)+9],[43:5];	06092500
FOR X2:=1 STEP 1 UNTIL B DO	06092600
IF (C:=DIR[X3+9+X2])≠0 THEN	06092700
IF P(C,DUP)<HI AND P(XCH)>LO THEN	06092800
IF FILEOK THEN GO FIXMV ELSE        % CHECK STATUS	06092900
IF NOT SYSTEMFILE(E,DIR[450+P(I,DUP,+)+1]) AND	06093000
DIR[X3+4],[12:4]=0 THEN        % NOT SYSTEM FILE	06093100
IF (P(DIR[X3+4],DUP),[1:3] OR P(XCH),[16:20] OR	06093200
DIR[X3+9],[1:28])=0 THEN        % FILE NOT IN USE	06093300
IF MOVEABLE THEN                    % NOT PERMANENT	06093400
BEGIN	06093500
FILEOK:=TRUE; % ELIMINATE STATUS CHECKING	06093600
FIXMV:    USE:=USE-(MV[CNT]:=C&DIR[X3+8][TODSIZE])	06093700
,DSIZE;	06093800
MV[CNT+1]:=(X1+I)&X2[CTF]; % HEADER INFO	06093900
IF PASSTWO THEN % SAVE LOC OF FIDS	06094000
MV[CNT+2]:=(X1+15)&(I×2)[CTF];	06094100
IF USE=0 THEN % FOUND ALL USERS OF IN-USE AREA	06094200
BEGIN	06094300
BOUNDARYCK;	06094400
GO OKINUSE;	06094500
END;	06094600
IF USE<0 THEN GO TO NOTOK; % DIRECTORY ERROR	06094700
IF (CNT:=CNT+3) MOD 150 = 0 THEN	06094800
BEGIN	06094900



```

                                IF CNT=MAXMVSZIE THEN GO TO NOTOK;
                                FIXARRAY(MV,X4,(CNT+150));
                                MOVE(CNT,A2,X4);
                                FORGETSPACE(A2);
                                A2I=X4;
                                END;
                                END ELSE GO TO NEXT ELSE GO TO NEXT;
                                END;
NEXT;      END;
                                END;
NOTOK;
                                TOG:=0;
OKINUSE;
                                INUSEOK:=TOG;
END SEARCHING IN USE AREAS;
SUBROUTINE MOVEANDFIX;
BEGIN
    I:=M[A2-1]; STOPCK;
    WHILE I<@1777 DO
    BEGIN
        DISKWAIT(=A4,30,(X2:=MV[I+1]),[CF]); % READ IN HEADER
        MVE: X1:=30; F:=P(MV[I],DUP),DEND+(B:=P(XCH),ASIZE);
            IF P(MV[I+2],DEND=0,DUP) THEN C:=MV[I+1],[2:26] ELSE
            MV[I],DEND:=(C:=MV[I+2],DEND)-B;
            WHILE (X1:=X1+30)<B DO
            BEGIN
                IF STOP THEN % STOP SQUASH BUT BE CAREFUL
                BEGIN
                    IF TEMPDSK THEN UT[MV[I+2],[2:10]]:=(P(DUP))-B;
                    UT[AV+1],DEND:=MV[I+1],[2:26];
                    C:=MV[I],DEND; FIXANDWRITEHEADER;
                    GO STOPSQ;
                END;
                E:=IF P((B-X1),DUP)<30 THEN P ELSE P(DEL,30);
                DISKIO(T,1=A3,E*30,F:=F-E);
                IOD:=IOD&(E*30)[8:38:10]&E[27:42:6];
                LOCIOD:=0; SLEEP([T],IOMASK);
                STREAM(A:=C:=C-E,B:=A3-1);
                BEGIN SI:=LOC A; DS:= 8 DEC END;
                IOREQUEST(NABS(IOD)&@357[25:40:8],IOD,
                    [LOCIOD]&18[12:42:6]);
                SLEEP([LOCIOD],IOMASK);
                IF LOCIOD.[28:1] THEN % WRITE LOCKOUT OCCURED
                BEGIN
                    UT[IF P THEN AV+1 ELSE MV[I+2],[2:10]],LOCKED:=1;
                    UT[AV+1],DEND:=MV[I+1],[2:26]; GO ENDMVE;
                END;
            END;
        END;
        FIXANDWRITEHEADER;
        IF NOT P THEN % TEMPORARY DISK STORAGE WAS USED,
        BEGIN
            MV[I+2],DEND:=0;
            TEMPDSK:=1;
            GO TO MVE;
        END;
        I:=MV[I+2],LINK;
    END;
    * WILL NOW RECONFIGURE THE AVAILABLE TABLE
    UT[AV]:=HI&(UT[AV],ASIZE+UT[AV+1],ASIZE)[2:28:20];
    MOVE(S=AV,P([UT[AV+2]],DUP),NOT 0 INX P(XCH));

```

```

06095000
06095100
06095200
06095300
06095400
06095500
06095600
06095700
06095800
06095900
06096000
06096100
06096200
06096300
06096400
06096500
06096600
06096700
06096800
06096900
06097000
06097100
06097200
06097300
06097400
06097500
06097600
06097700
06097800
06097900
06098000
06098100
06098200
06098300
06098400
06098500
06098600
06098700
06098800
06098900
06099000
06099100
06099200
06099300
06099400
06099500
06099600
06099700
06099800
06099900
06100000
06100100
06100200
06100300
06100400
06100500
06100600
06100700
06100800
06100900

```



```

$ POP OMIT
FIXARRAY(DIR,A1,480); FIXARRAY(MV,A2,150);
A3:=SPACE(900);
IOD:=@1400001000000000&(A3=1)[CTC];
IF NOT SQALL THEN FOR EU:=1 STEP 1 UNTIL NEUP,[FF] DO
IF (CELL:=(P(SQSIZE,DUP)≠0 AND P(XCH) LEQ U[EU],[1:20]))
THEN BEGIN P(2); SQUASHMESS; GO STOPIT END;
FOR EU:=1 STEP 1 UNTIL NEUP,[FF] DO &
IF NOT (E:=U[EU]),EUNP THEN % NOT A DUMMY EU
IF EUS[EU=1] OR SQALL OR SQSIZE≠0 THEN & SQUASH THIS EU
BEGIN
EUNOTSQUASHED:=TRUE;
IF NOT SQALL THEN % CHECK IF SQUASH IS NECESSARY
IF (P(EUS[EU=1],DSIZE,DUP) LEQ E,[1:20] AND P(XCH)≠0)
THEN BEGIN P(3); SQUASHMESS; GO STOPIT END;
CELL:=O&1[CTF];
P(0); SQUASHMESS;
D:=(I:=E,STARTWRD) MOD 30;
AVSIZE:=30*(S:=(E AND NUMENTM)+D) MOD 30+S;
FIXARRAY(UT,R,AVSIZE);
DISKAV:=I DIV 30+USERDISKBOTTOM;
$ SET OMIT = NOT SHAREDISK
IF DISKAV=USERDISKBOTTOM THEN
BEGIN
IF AVSIZE>30 THEN DISKWAIT(-R=30,AVSIZE=30,DISKAV+1);
MOVE(30,R1,R);
END ELSE
$ POP OMIT
DISKWAIT(-R,AVSIZE,DISKAV);
AGAIN: SUM:=USE:=0;
FOR I:=S=3 STEP -1 UNTIL D DO
BEGIN STOPCK;
IF (UT[I+1]<0)=PASSTWO THEN % NOT CHECKED THIS PASS
IF ((X1:=UT[I],ASIZE)+(X2:=UT[I+1],ASIZE)) GEQ SUM
THEN IF (X3:=(((X4:=UT[I+1],DEND)-1)-UT[I+1],ASIZE)-
X5:=(UT[I],DEND-1)) LEQ FACTOR THEN IF MINSIZE LEQ X2
THEN IF MINSIZE LEQ X1 THEN
BEGIN
SQIT: USE:=X3; AV:=I;
SUM:=X1+X2; % SUM OF CURRENT AVAILABLE AREAS
HI:=X4; LO:=X5;
END ELSE IF I≠0 THEN % LOOK AHEAD TO NEXT AREA
IF ((MINSIZE LEQ UT[I-1],ASIZE) AND (((X5-X1)-
UT[I-1],DEND-1) LEQ FACTOR)) THEN GO SQIT;
END;
IF USE≠0 THEN % FOUND A POSSIBLE SQUASH SITUATION
BEGIN
IF INUSEOK THEN MOVEANDFIX;
GO AGAIN;
END ELSE % TIME TO WRAP IT UP FOR THIS EU UNLESS...
IF CONFLICT THEN IF NOT PASSTWO THEN % .CONFLICTS EXIST
BEGIN
PASSTWO:=TRUE;
GO AGAIN;
END ELSE
BEGIN % CLEAN-UP PASS AFTER CONFLICTS RESOLVED,
PASSTWO:=CONFLICT:=0;
GO AGAIN;
END;
STOPSQ: FOR I:=D STEP 1 UNTIL S DO UT[I]:=ABS(P(DUP,LUD)&O[2:2:1]);

```

```

06107100
06107200
06107300
06107400
06107900
06108000
06108100
06108200
06108300
06108400
06108500
06108600
06108700
06108800
06108900
06109000
06109100
06109200
06109300
06109400
06109500
06109600
06109700
06109800
06109900
06110000
06110100
06110200
06110300
06110400
06110500
06110600
06110700
06110800
06110900
06111000
06111100
06111200
06111300
06111400
06111500
06111600
06111700
06111800
06111900
06112000
06112100
06112200
06112300
06112400
06112500
06112600
06112700
06112800
06112900
06113000
06113100
06113200
06113300
06113400

```

```

IF NOT EUNOTSQUASHED THEN
$ SET OMIT = NOT SHAREDISK
  IF DISKAV=USERDISKBOTTOM THEN
  BEGIN
    MOVE((IF S>30 THEN 30 ELSE S)=D,[UT[D]],R1+D);
    IF AVSIZE>30 THEN DISKWAIT( R+30,AVSIZE=30,DISKAV+1);
  END ELSE
$ POP OMIT
  DISKWAIT( R,AVSIZE,DISKAV);
  FORGETSPACE(R);
  P(1); SQUASHMESS;
STOPIT: IF STOP THEN GO OUT; % STOPCK GOT US HERE
  END EU LOOP;
OUT:
$ SET OMIT = NOT SHAREDISK
  DISKWAIT( R1,=30,USERDISKBOTTOM); % NOTE WRITE UNLOCK
  FORGETSPACE(R1);
$ POP OMIT
  FORGETSPACE(A1); FORGETSPACE(A2);
  FORGETSPACE(A3); FORGETSPACE(A5);
SDXIT:
  FORGETSPACE(A4);
  CELL:=PRTVALUE;
  STREAM(A:=BUFF,[15:15]=1); DS:=13 LIT" END SQUASH,+";
  SPOUT(BUFF,[15:15]=1);
$ SET OMIT = SHAREDISK
  UNLOCKDIRECTORY;
$ POP OMIT
  UNLOCKTOG(USERDISKMASK);
  NOPROCESSTOG:=NOPROCESSTOG=1;
  KILL([BUFF] INX NOT 1);
END SQUASHING;

SAVE PROCEDURE INITIALIZE; FORWARD;
REAL ACTDATE=INITIALIZE;
SAVE REAL PROCEDURE COREND; FORWARD;
REAL WEEKDAY=COREND;
PROCEDURE USERDISKSPECIALCASE(Q,R,UT,J) ;
VALUE Q,J; REAL R,J; INTEGER Q; ARRAY UT[*] ;
BEGIN
  REAL BUFF=Q,N=J,Z=UT,E=R ,WEWONTGO=R;
$ SET OMIT = NOT(SHAREDISK )
  INTEGER NT=J ;
  REAL NEU=AVS; DEFINE U=UT #, UA=UT #, NEU1=NEU+J #, NEU2=NEU+NEU#;
$ POP OMIT
$ SET OMIT = SHAREDISK
  REAL NEU,NT; ARRAY UA[*] ;
  DEFINE U=AVTABLE #, AVS=B #, NEU1=J-1 #, NEU2=NT=1 #;
$ POP OMIT
  INTEGER NT1,NT3,NT4,B ;
  LABEL L1,L2,L3,UP,PU,BD,WHY,M1,T10 ;
  LABEL UNLOADER;
  SWITCH SW=L1,L2,L3 ;
  IF Q#0 THEN GO SW[Q=1] ;
$ SET OMIT = NOT(SHAREDISK )
  IF (B1=J,[CF])>1023 THEN GO BD; Q:=SPACE(B); MOVE(J,[FF],R,Q) ;
  FORGETSPACE(R); P(((R1=Q)&B[TOSIZE]) OR M,RTN) ;
$ POP OMIT
L1: BUFF:=R; Z:=0; UNLOCKTOG(USERDISKMASK);
  WEWONTGO := -1;

```

```

06113500
06113600
06113700
06113800
06113900
06114000
06114100
06114200
06114300
06114400
06114500
06114600
06114700
06114800
06114900
06115000
06115100
06115200
06115300
06115400
06115500
06115600
06115700
06115800
06115900
06115990
06116000
06116010
06116100
06116200
06116300
06116400
06179000
06179400
06179500
06179600
06179700
06350000
06350300
06350600
06351000
06351050
06351053
06351055
06351056
06351100
06351104
06351105
06351106
06351250
06351500
06351600
06351800
06352000
06352490
06352500
06352600
06352601
06353500
06353600

```

```

IF N LEQ RESERVEDISKSIZ THEN          % CALL OUT THE RESERVES      06353605
IF (Z:=DIRECTORYSEARCH("RESERVE","DISK  ",6))#0 THEN              06353610
BEGIN FORGETSPACE(Z);                                           06353615
  WEWONTGO := 0;                                               06353620
  IF N GTR 0 THEN                                             06353625
$ SET OMIT = PACKETS                                           06353629
  IF NOT (LIBMSG AND (LOGLINE.[33:7]=0 OR CANDYMESS)) THEN      06353630
$ POP OMIT                                                       06353631
  BEGIN STREAM(Z:=Z:=SPACE(3));                                  06353640
    DS:=23 LIT"***RESERVE DISK REMOVED*";                       06353650
    SPOUTER(Z,0,(NOT LIBMSG) AND 1);                             06353660
  END;                                                         06353670
  GO TO UNLOADER;                                             06353680
END OF RESERVE CALL UP;                                        06353690
IF NOT N.[2:1] THEN                                           06353700
BEGIN IF P1MIX#0 THEN                                          06354000
WHY:  STREAM(J:=JARROW[P1MIX],P1MIX,N,BUFF);                  06355000
  BEGIN DS:=14 LIT"#NO USER DISK:";                             06356000
    SI+J; SI+SI+1; DS+7 CHR;                                    06357000
    DS+LIT "/"; SI+SI+1; DS+7 CHR;                              06358000
    SI+LOC P1MIX; DS+LIT "="; DS+2 DEC;                        06359000
    J:=DI; DI:=DI-2; DS:=FILL; DI:=J; DS:=LIT"-";             06359500
    SI:=LOC N; DS:=8 DEC; DS:=7 LIT" SEGS,+";                 06360000
    DI:=DI-15; DS:=7 FILL;                                     06360500
  END ELSE                                                    06361000
  STREAM(N,BUFF);                                             06361500
  BEGIN DS:=20 LIT"#NO USER DISK:MCP = ";                       06362000
    SI:=LOC N; DS:=8 DEC;                                       06362500
    DS:=6 LIT" SEGS,+";                                         06363000
    DI:=DI-14; DS:=7 FILL;                                     06363500
  END;                                                         06364000
  SPOUT(BUFF);                                               06364500
END ELSE FORGETSPACE(BUFF);                                   06365000
UNLOADER:                                                     06368000
IF AUTOUNLD THEN                                             06369000
BEGIN P(P1MIX); AUTOUNLD:=P1MIX:=0;                            06369500
  STREAM(ACTDATE,Z:=Z:=SPACE(10)+2);                          06370000
  BEGIN DS:=24 LIT"CC UNLOAD EXPIRED TO EXP";                 06370500
    SI+LOC ACTDATE; SI+SI+2; DS+4 CHR;                         06371000
    DS:=9 LIT" =/=;END,";                                     06371500
  END;                                                         06372000
  CCARD(Z&31[3:43:5]);                                       06372500
  P1MIX:=P;                                                  06373000
  IF N GEQ 0 THEN                                             06373500
  BEGIN STREAM(Z:=Z:=SPACE(10));                               06374000
    DS:=18 LIT"19 AUTOUNLD RESET*";                             06375000
    SPOUT(Z);                                                 06376000
  END END AUTOMATIC UNLOADING;                                06377000
IF WEWONTGO THEN                                             06377100
IF NOT N.[2:1] THEN                                           06377200
  BEGIN IF WEWONTGO.[1:1] THEN                                06377300
    IF LOGLINE GTR 0 OR P1MIX=0 THEN                          06377400
      BEGIN M[Z:=GETAREA(0)]:=(*P(DUP))&LOGLINE[CTF]&        06377500
        15[18:41:7];                                         06377600
        M[Z+1] := 0;                                         06377700
        M[Z+2] := N;                                         06377800
        @EVENT(Z,CANDYINX);                                  06377900
        WEWONTGO := @100001;                                 06378000
      END;                                                     06378100
    IF P1MIX=0 THEN                                           06378200

```

BEGIN CLICK=CLOCK+P(RTR)+1800;%WAIT 30 SECS,	06378210
SLEEP(0,0);	06378220
END ELSE	06378230
BEGIN PRTRROW[P1MIX],[7:1]:=1;	06378300
IF OUTWAIT(FALSE) THEN	06378400
BEGIN BUFF:=SPACE(10);	06378500
GO TO WHY;	06378600
END;	06378700
PRTRROW[P1MIX],[7:1] := 0;	06378800
END;	06378900
WEWONTGO,[CF] := 0;	06379000
END WAITING FOR DISK TO SHOW UP;	06379100
P(XIT);	06379200
L2: U[J]:=E; E:=NEU:=(NT:=NEUP,NEUF)+2+(NT+1)DIV 2; P(NT); J:=1;	06380100
\$ SET OMIT = SHAREDISK	06380120
NT1:=NT+NT+NT; FORGETSPACE(UT); FIXARRAY(UA,NT2,NT1); E1=0;	06380140
\$ POP OMIT	06380141
UP: IF (NT4:=E MOD 30) LSS (NT3:=(NT1:=U[J],STARTWRD) MOD 30)	06380150
THEN NT4:=NT3 ;	06380200
IF (NT2:=(Q:=U[J] AND NUMENTM)+NT4) GTR 1023	06380250
OR ((Q+E+1) DIV 30+1=F DIV 30) GTR 34 THEN	06380300
BD: BYBY("ODISK IS TOO CHECKERED,..PLEASE COMPACT IT*",43) ;	06380350
DISKWAIT("((UA[NEU1]:=UA[NEU2+J]:=SPACE(NT2))+NT4)-NT3),Q+NT3,	06380400
USERDISKBOTTOM+NT1 DIV 30) ;	06380450
\$ SET OMIT = NOT(SHAREDISK )	06380490
IF J=1 THEN MOVE(NEU,[UI0]),B:=UA[NEU2+1]) ;	06380500
\$ POP OMIT	06380501
\$ SET OMIT = SHAREDISK	06380520
IF J=1 THEN B:=UA,[CF]+NT+NT-1 ;	06380525
\$ POP OMIT	06380526
M[B+J]:=U[J]&E[TOSTARTWRD] ;	06380550
IF (NT1:=Q DIV 4) LSS AVDIFFMIN THEN NT1:=AVDIFFMIN ;	06380600
IF (E1=E+Q+NT1) GTR AVTMAX THEN GO TO BD;	06380650
IF P(DUP) GEQ J:=J+1 THEN GO UP; E:=E-NT1; J:=1 ;	06380700
PU: NT2:=(NT3:=P(M[B+J],DUP),STARTWRD)+NT5:=P(XCH) AND NUMENTM ;	06380750
IF P(DUP)≠J THEN IF (NT2-1)DIV 30=(NT4+M[B+J+1],STARTWRD)DIV 30 THEN	06380800
MOVE(NT1-NT2 MOD 30,UA[NEU1]+NT5-NT1,NT1+UA[NEU1+1]-NT4 MOD 30);	06380850
DISKWAIT(UA[NEU1]-NT1-NT3 MOD 30,NT1+NT5,USERDISKBOTTOM+NT3 DIV 30);	06380900
\$ SET OMIT = NOT(SHAREDISK)	06380924
IF J NEQ 1 THEN %SHAREDISK ONLY	06380925
\$ POP OMIT	06380926
FORGETSPACE(UA[NEU2+J]);	06380950
IF P(DUP) GEQ J:=J+1 THEN GO PU ;	06381000
\$ SET OMIT = SHAREDISK	06381020
MOVE(NT,[UA[NT+NT]],[AVTABLE[1]]) ;	06381070
\$ POP OMIT	06381071
\$ SET OMIT = NOT(SHAREDISK )	06381075
FORGETSPACE(B) ;	06381080
\$ POP OMIT	06381081
FORGETSPACE(UA) ;	06381085
\$ SET OMIT = NOT(SHAREDISK )	06381095
Q1:=SPACE(AVS:=(AVS:=IF E LSS AVSMIN THEN AVSMIN ELSE IF E GTR AVSMAX	06381100
THEN AVSMAX ELSE E)+30-(IF (E:=AVS MOD 30)≠0 THEN E	06381150
ELSE 30)) ;	06381155
DISKWAIT("Q,AVS,USERDISKBOTTOM) ;	06381200
\$ POP OMIT	06381201
P(DEL,Q&AVS[TO SIZE] OR M,RTN) ;	06381250
L3: P(U[NEUP,NEUF+2+(Q1=J DIV P(M1)) DIV 2],IF Q THEN P,[8:20] ELSE	06381300
P,[28:20]) ;	06381310
IF U[Q+1],SPEED = 2 THEN	06381320

BEGIN % 40=MILL MASK CONSTRUCTION,	06381330
Q1=P ;	06381335
STREAM(S:=0;Q);	06381340
BEGIN	06381345
SI:=LOC Q; SKIP 28SB; DI:=LOC S; SKIP 8DB ;	06381350
5(4(IF SB THEN DS:=SET ELSE SKIP DB;SKIP SB); SKIP 4 DB);	06381355
SI:=LOC Q; SKIP 28 SB; DI:=LOC S; DI:=DI+2;	06381360
5(4(IF SB THEN DS:=SET ELSE SKIP DB;SKIP SB); SKIP 4 DB);	06381365
END STREAM ;	06381380
END ;	06381390
STREAM(MSK:=0;V:=47-(J:=((Q:=J MOD P(M1))+ABS(R)-1) DIV P(T10)),	06381395
W:=1+J-Q DIV P(T10));	06381400
BEGIN DI:=LOC MSK; SKIP V DB; DS:=W SET; END;	06381405
P(LND,LNG,0,LNG,*,RTN);	06381410
M1::: @3641100; % DECIMAL 1000000,	06381450
T10::: @23420; % DECIMAL 10000,	06381500
END OF USERDISKSPECIALCASE ;	06381550
PROCEDURE GETMOREOLAYDISK(MIX);%	06400000
VALUE MIX;%	06401000
INTEGER MIX;%	06402000
BEGIN INTEGER I:=+1,%	06403000
J:=+2,%	06404000
T:=+3;%	06405000
ARRAY A:=+4[*];%	06406000
REAL RCW:=+0;%	06407000
LABEL EXIT;%	06408000
DEFINE DALOCMAXSZ =	06408100
\$ SET OMIT = NOT(AUXMEM)	06408199
111#; %DALOC SIZE MUST = 7 INITIALLY.	06408200
\$ POP OMIT	06408201
\$ SET OMIT = AUXMEM	06408299
127#; %DALOC SIZE MUST = 7 INITIALLY.	06408300
\$ POP OMIT	06408301
P(0, 0, 0, 0);	06409000
IF (T+DALOC[MIX,0],[CF]+1)=DALOCMAXSZ THEN BEGIN	06410000
TERMINATE(MIX & 37[CTF]);	06411010
GO TO EXIT; END;	06411030
IF T=DALOCROW[MIX],[8:10] THEN%	06412000
BEGIN IF(J+T+P(DUP)=1)=97 THEN J+DALOCMAXSZ;	06413000
WHILE (I := GETSPACE(J, 0, 3)+2)=2 DO	06414000
SLEEP([CLOCK], NOT CLOCK);	06415000
MOVE(T, DALOCROW[MIX], 1);	06416000
FORGETSPACE(DALOCROW[MIX]);	06417000
DALOCROW[MIX] := (*P(DUP)) & I[CTC] & J[8:38:10];	06417500
END AIT TYPE ACTION;%	06419000
IF (I + GETUSERDISK(500 OR MEMORY))=0 THEN GO TO EXIT;%	06420000
DALOC[MIX,0] + (*P(DUP))&(T+1)[CTC];%	06421000
DALOC[MIX,T] + 1;%	06422000
DALOC[MIX,T+1] + 0;%	06423000
EXIT; OLAYMASK + TWO(MIX) OR OLAYMASK;%	06424000
IOCOUNT[MIX] + *P(DUP)-1;%	06425000
KILL([MIX] INX NOT 1);	06426000
END GET MORE OVERLAY DISK FOR A GIVEN JOB;%	06427000
REAL PROCEDURE SECURITYCHECK(MID,FID,USERID,HEADER);	06460000
VALUE MID,FID,USERID;	06460100
REAL MID,FID,USERID,HEADER;	06460200
%  MID      MULTI FILE ID OF FILE TO BE CHECKED	06460300
%  FID      FILE ID OF FILE TO BE CHECKED	06460400
%  USERID   USER IDENTIFICATION	06460500

%	HEADER	06460600
%	>512 CORE ADDRESS OF HEADER IN 33:15, JUST CHECK IT,	06460700
%	>0, <512 VALUE FOR DIRECTORYSEARCH, FIND THE FILE AND PASS	06460800
%	BACK THE HEADER IN ADDITION TO SECURITY INFO,	06460900
%		06461100
%	RESULT FROM SECURITYCHECK	06461200
%	=0 NO LEGITIMATE USER FOUND	06461300
%	=2 TERTIARY USER ( INPUT ONLY)	06461400
%	=3 SECONDARY USER (INPUT/OUTPUT)	06461500
%	=7 PRIMARY USER (INPUT/OUTPUT/LIB MAINT.)	06461600
	BEGIN	06462000
	REAL T2,DKSGROW,CODES,ROWS,ROW,DKADR,ROWSZ,C,USER,TYPE,SH;	06462100
	REAL I=DKSGROW, FPBSIZE=CODES;	06462105
	ARRAY FH[*],FPB=ROW[*];	06462110
	LABEL FOUND;	06462120
	LABEL EXYT,NOTFOUND,LOOK,WHY,FORGET;	06462200
	REAL SUBROUTINE DIRSRH;	06463000
	BEGIN	06463100
LOOK:	IF (T2=DIRECTORYSEARCH(MID,FID,HEADER)) LSS 64 THEN	06463200
WHY:	BEGIN	06463210
	IF T2=0 THEN FILEMESS(IF LOGLINE,[33:7]=0 THEN "#NO FIL"	06463220
	ELSE ="NO FILE","ON DISK",MID,FID,0,0,0)	06463225
	ELSE IF T2=1 THEN BEGIN P(DEL); TYPE=-1; GO EXYT; END	06463230
	ELSE IF T2=2 THEN FILEMESS("#SYSFIL","ERROR ",	06463240
	MID,FID,0,0,0);	06463250
	REPLY[P1MIX]:=-(SH:=VWY&VOK[36:42:6]&VIL[30:42:6]);	06463280
	IF P(0,RDS)>FENCE THEN SWAP(WAITSWAP,1) ELSE	06463300
	COMPLEXSLEEP(REPLY[P1MIX]>0 OR TERMSET(P1MIX));	06463320
	IF TERMSET(P1MIX) THEN GO TO INITIATE;	06463340
	IF NOT WHYSLEEP(SH) THEN GO TO WHY;	06463360
	IF (SH:=T2:=REPLY[P1MIX],[FF]) GTR 32 THEN % IL	06463380
	BEGIN STREAM(T2:);	06463400
	BEGIN SI:=T2;	06463420
	LL: SI:=SI+1; IF SC="L" THEN GO TO LL;	06463440
	SI:=SI+1; T2:=SI;	06463460
	END;	06463480
	T2:=P;	06463500
	FPBSIZE:=(FPB:=PRT[P1MIX,3]),[8:10];	06463520
	FOR I:=0 STEP ETRLNG UNTIL FPBSIZE DO	06463540
	IF (FPB[I] EQV MID)=NOT 0 THEN	06463560
	IF (FPB[I+1] EQV ABS(FID))=NOT 0 THEN GO FOUND;	06463580
FOUND:	NAMEID(C,T2); MID:=C; NAMEID(C,T2);	06463600
	NAMEID(C,T2); FID:=C&FID[1:1:1];	06463620
	IF I LSS 1020 THEN	06463640
	BEGIN FPB[I]=MID;	06463660
	FPB[I+1]=C;	06463680
	END;	06463700
	FORGETSPACE(SH=1);	06463720
	END;	06463740
	REPLY[P1MIX]:=0;	06463760
	GO TO LOOK;	06463780
	END;	06463800
	DIRSRH I= T2;	06463810
END DIRSRH;		06463820
	FH:=IOQUE&(IF HEADER GTR 512 THEN HEADER ELSE DIRSRH)(CTC);	06463900
	IF(FH[2] EQV 0)=NOT 0 OR (ABS(USERID) EQV ABS(FH[2]))=NOT 0	06463910
	OR (USERID EQV MCP)=NOT 0 THEN TYPE=7 ELSE%	06463920
	IF (FH[5] EQV @14)=NOT 0 THEN%	06463930
	IF (FH[6] EQV @14)=NOT 0 THEN TYPE=2 ELSE TYPE=3;%	06463940
	IF TYPE # 0 THEN GO TO EXYT;	06463950



```

%
IF FH[5],[1:1] THEN
BEGIN IF (SH=DIRECTORYSEARCH(ABS(FH[5]),FH[6],19))=0
THEN BEGIN TYPE:=0; GO TO EXYT END;
M[SH+4],[11:11]:=1;
STREAM(DATE,J:=5); BEGIN SI:=LOC DATE; DS:=8OCT; END;
M[SH+3],[12:18]:=JUNK;
DISKWAIT(SH,[CF],30,SH,[FF]);
$ SET OMIT = SHAREDISK
UNLOCKDIRECTORY;
$ POP OMIT
DKSGROW:=M[SH INX 8];
CODES:=GETSPACE(30,0,0)+2; ROWS:=(M[SH INX 9]AND 31)-1;
FOR ROW:=0 STEP 1 UNTIL ROWS DO
BEGIN IF (DKADR:=M[SH INX 10+ROW])=0 THEN
NOTFOUND: BEGIN TYPE := 0;
FORGET: FORGETSPACE(CODES); FORGETSPACE(SH); GO TO EXYT;
END;
ROWSZ := DKADR + DKSGROW;
WHILE DKADR < ROWSZ DO
BEGIN DISKWAIT(=CODES,30,DKADR);
FOR C:=0 STEP 1 UNTIL 29 DO
BEGIN IF((USER:=NFLAG(M[CODES INX C]))EQV @114)=
NOT 0 THEN GO TO NOTFOUND;
IF (USER EQV @14)≠ NOT 0 THEN
IF USER,[3:3]=0 THEN
BEGIN
IF (USERID EQV ABS(USER))=NOT 0 THEN
BEGIN TYPE :=
IF USER < 0 THEN 2 ELSE 3;
GO TO FORGET;
END;
END ELSE
BEGIN
IF P1MIX ≠ 0 THEN
BEGIN
IF LOGLINE,[33:7] NEQ 0 THEN
USER:=" "&USER[6:12:36]&USER[3:3:3];
IF(ABS(JAR[P1MIX,0])EQV
USER,[6:42])= NOT 0 THEN
IF((IF JAR[P1MIX,0]<0 THEN "DISK "ELSE JAR[P1MIX,1])EQV
M[CODES INX C+1],[6:42])= NOT 0
THEN
BEGIN
TYPE := USER,[3:3];
GO TO FORGET;
END; C:=C+1;
END; % P1MIX NEQ 0
END;
END; % 30 USERS
DKADR := DKADR + 1;
END; % ROW
END; % ROWS
GO TO NOTFOUND;
END; % NO SECURITY BLOCK FILE
TYPE :=0;
EXYT:
IF HEADER LSS 512 THEN HEADER:=FH;
SECURITYCHECK :=TYPE;

```

```

06463955
06463960
06463970
06463980
06463982
06463984
06463986
06463988
06463990
06463992
06463994
06463996
06464000
06464100
06464200
06464300
06464400
06464500
06464600
06464700
06464800
06465000
06465100
06465200
06465210
06465220
06465230
06465300
06465400
06465500
06465600
06465700
06465800
06465805
06465810
06465812
06465814
06465816
06465820
06465830
06465840
06465850
06465860
06465870
06465880
06465890
06465900
06465910
06465912
06465920
06466000
06466100
06466200
06466300
06466310
06466400
06466500
06466600
06466620
06466700

```

END SECURITYCHECK;	06466800
BOOLEAN PROCEDURE OUTWAIT(B); BOOLEAN B;	06467000
BEGIN REPLY[P1MIX]←VOK&VWY[36:42:6]&1[2:47:1];	06468000
IF P(O,RDS)>FENCE THEN	06469000
SWAP(WAITSWAP,1) ELSE	06470000
COMPLEXSLEEP((B OR REPLY[P1MIX]>0 OR TERMSET(P1MIX)));	06471000
	06472000
IF TERMSET(P1MIX) THEN	06473000
IF (JAR[P1MIX,0] EQV "LIBMAIN")≠NOT 0 OR	06473100
(JAR[P1MIX,1] EQV "DISK ")≠NOT 0 THEN GO TO INITIATE;	06473200
IF REPLY[P1MIX]=VWY THEN OUTWAIT←NOT WHYSLEEP(VOK&VWY[36:42:6]);	06474000
REPLY[P1MIX]←0;	06475000
END OUTWAIT;	06476000
REAL PROCEDURE OUTRAN980 (ADR,NUM,TYPE,LCC,B);	%10906500000
VALUE ADR,NUM,TYPE,LCC;	%10906501000
REAL ADR,NUM,TYPE,LCC,B;	%10906502000
BEGIN REAL C,T,W,LAS;	06503000
REAL SPARES=ADR,DONE=TYPE,LSIZ=LCC;	06503100
LABEL AGAIN;	06503200
B←SPACE(30)+1;	06504000
LAS:=TYPE,[2:1];	06504500
IF TYPE,[1:1] THEN	06505000
BEGIN T←B; C←1 END ELSE T←(C+B)+@677777;	06506000
STREAM(W←0;C←0);	06508000
BEGIN W←CI; GO TO X;	06509000
W←IF SC="0" THEN BEGIN DS←LIT"<"; GO TO L END;	06510000
IF SC="1" THEN BEGIN DS←LIT">"; GO TO L END;	06511000
IF SC="=" THEN BEGIN DS←LIT"≤"; GO TO L END;	06512000
IF SC="\$" THEN BEGIN DS←LIT"≥"; GO TO L END;	06513000
IF SC="*" THEN BEGIN DS←LIT"≠"; GO TO L END;	06514000
DS←LIT MARK;	06515000
L1 SI←SI+1; JUMP OUT TO Y);	06516000
IF SC=">" THEN GO TO E;	06517000
IF SC="<" THEN GO TO E;	06518000
IF SC="≥" THEN GO TO E;	06519000
IF SC="≤" THEN GO TO E;	06520000
IF SC="+" THEN GO TO E;	06520500
IF SC="*" THEN	06521000
E: BEGIN DS←LIT MARK; SI←SI+1; CI←C END;	06522000
DS←CHR;	06523000
Y: CI←C;	06523100
X:END;	06524000
W←P;	06525000
DO BEGIN;	06525100
IF TYPE,[1:1] THEN	06526000
STREAM(Q←0,C←ADR,T;LAS,G←C,W←NI←IF NUM GTR 63 THEN	06527000
63 ELSE NUM);	06528000
BEGIN SI←ADR; DI←T;	06529000
N←IF SC=ALPHA THEN IF SC<"0" THEN	06530000
BEGIN DS←CHR; GO TO L END;	06531000
IF SC="*" THEN	06532000
BEGIN G←TALLY←0; JUMP OUT TO L1); TALLY←1;	06533000
L1: G←TALLY; SI←SI+1;	06534000
IF SC="*" THEN DS←LIT MARK;	06535000
SI←SI+2; GO TO L1;	06536000
END;	06537000
CI←CI+LAS; GO TO L2;	06537500
IF SC="+" THEN	06538000
BEGIN TALLY←1; Q←TALLY; DS←CHR;	06539000
DI←DI+1; JUMP OUT END;	06539100

```

L2:  NI=CI; CI=W; 06540000
L:  );) 06541000
    ADR=SI; T=DI; TALLY=G; C=TALLY; 06542000
END ELSE 06543000
STREAM(Q:=0,C,ADR,T;LAS,G:=0,W,N:=IF NUM GTR 63 THEN 06544000
      63 ELSE NUM); 06545000
BEGIN SI=ADR; DI=C; 06546000
N(IF SC=" " THEN DS=CHR ELSE 06547000
  BEGIN T=DI; 06548000
    IF SC=ALPHA THEN DS=CHR ELSE 06549000
    BEGIN 06550000
      CI:=CI+LAS; GO TO L2; 06550500
      IF SC="*" THEN 06551000
      BEGIN DI=DI-1; T=DI; 06552000
        TALLY+1; Q=TALLY; 06553000
        JUMP OUT; 06554000
      END; 06555000
    L2:  NI=CI; CI=W; 06556000
      END END); 06557000
    ADR=SI; C=DI; 06558000
  END; 06559000
  T=P; ADR=P; C=P; 06560000
END UNTIL (NUM+NUM=63)≤0 OR P; 06561000
NUM=P(T=B+1,DUP),[30:3]&P(XCH)[30:33:15]=/; 06562000
IF TYPE,[1:1] THEN 06563000
BEGIN STREAM(T); 06563200
  DS=LIT LEFTARROW; 06563400
END ELSE 06563600
BEGIN IF LCC THEN LCC:=1+(NUM+8),[40:4]; 06564000
  IF TYPE,[CF] GTR 14 THEN TYPE,[CF]:=14; 06564250
  STREAM(T; XC=LCC, CR=TYPE,[FF]=1, LF=TYPE); 06564500
  BEGIN DI=T; DI:=DI+1; %10906565000
    CR(DS=LIT"≤"); 06566000
    LF(DS=LIT"≠"); 06568000
    XC(DS=LIT"<"); % LIT IS A LSS CHR, %10906568100
    DS=LIT"="; 06569000
    T=DI; 06570000
  END; 06571000
  TI=P; 06571500
  NUM:=(C:=NUM)+TYPE,[CF]=TYPE,[32:1]+LCC+2; 06572000
  IF LCC=0 AND C GTR 72 THEN % FOLD THE LINE 06572500
  BEGIN SPARES:=IF NUM>184 THEN 186-C ELSE 107; 06573000
    DONE:=LSIZ:=70; 06573500
    IF (W:=IF C=LSIZ GTR 38 THEN 24 ELSE LSIZ+60-C) 06574000
      GEQ SPARES THEN W=SPARES; 06574500
    STREAM(NI=(NUM+7),[39:6]=(DONE+7),[39:6]; 06575000
      NI:=(8-DONE,[45:3]),[45:3], W, T); 06575500
    BEGIN SI=T; SI:=SI-8; DI:=DI+8; 06576000
      N(DS=WDS; SI:=SI-16; DI:=DI-16); 06576500
      DS=WDS; SI:=SI-NI; DI:=DI-NI; 06577000
      W(IF SC=" " THEN 06577500
        BEGIN DS=CHR; DI:=DI-17; 06578000
          JUMP OUT TO L; 06578500
        END; 06579000
        TALLY:=TALLY+1; DS=CHR; 06579500
        SI:=SI-2; DI:=DI-2); 06580000
        SI:=SI+17; DI:=DI-15; DS=W CHR; 06580500
        TALLY:=0; DI:=DI-1; 06581000
      L:  NI=TALLY; DS=16 LIT"xxS# xx"; 06581500
    END; 06582000

```

AGAIN!

```

NUM:=NUM+16;
IF (C:=(LAS:=P)+C=LSIZ) GTR 60 THEN
BEGIN DONE:=DONE+74-LAS;
SPARES:=SPARES-LAS;
T:=T+2;
LSIZ:=58;
GO AGAIN;
END END END;
OUTRAN980+NUM;
END;
$ SET OMIT = TWXONLY
REAL PROCEDURE OUTRANBIDS(ADR,NUM,TYPE,B,T);
VALUE ADR,NUM,TYPE;
REAL ADR,NUM,TYPE,B,T;
BEGIN REAL D,X,Y,Z,N1,N,Q;
LABEL ZIT;
B:=SPACE(30)+1;
IF TYPE,[1:1] THEN
BEGIN;
STREAM(Y:=Q:=0,ADR:=N:=ADR,D:=B);
BEGIN SI:=ADR; DI:=D;
IF SC=ALPHA THEN IF SC LSS 0 THEN GO TO B;
A: IF SC="*" THEN
BEGIN DS:=CHR;
IF SC!="*" THEN
B: BEGIN TALLY:=1; Y:=TALLY; GO TO YIT END;
DS:=CHR; GO TO A;
END;
YIT: DI:=DI; ADI:=SI;
END STREAM;
DI:=P; ADR:=P; X:=Y:=P;
NUM:=NUM - ADR.[30:3]&(ADR=N)[30:33:15];
END ELSE
BEGIN STREAM(D:=Q:=B,X:=(NUM#X);LAS:=(NOT TYPE),[2:1],ADR);
BEGIN DI:=D;
X(SI:=ADR;
IF SC=LEFTARROW THEN
BEGIN CI:=CI+LAS; GO TO L;
TALLY:=0; X:=TALLY;
END ELSE
L: DS:=LIT"*");
DI:=DI;
END;
X:=Y:=P; DI:=P;
END;
DO
BEGIN NI:=IF (N1:=NUM GTR 63) THEN 63 ELSE NUM;
IF TYPE,[1:1] THEN
BEGIN;
STREAM(DISC:=8,ADR,D,Z,X,Y:=Q;LAS:=(NOT TYPE),[2:1],N,N1,
A:=@1660140206557501,B:=@2072350304000000,
C:=0,C1:=0,C2:=0,C3:=0);
BEGIN SI:=ADR; DI:=D;
C1:=C1; GO TO NEXT;
Y(SI:=SI+1;
IF SC="*" THEN SI+SI=1 ELSE
BEGIN TALLY:=0; Y:=TALLY;
TALLY:=1; JUMP OUT TO RETURN;
END);
X(TALLY:=0; JUMP OUT TO L1);

```

```

06582500
06583000
06583500
06584000
06584500
06585000
06585500
06586000
06587000
06588000
06588990
06590000
06590100
06590200
06590300
06590400
06590500
06590600
06590700
06590800
06590900
06591300
06591400
06591500
06591600
06591700
06591800
06591900
06592400
06592500
06592600
06592700
06593100
06593200
06593300
06593320
06593340
06593360
06593380
06593400
06593420
06593440
06593500
06593600
06593700
06593800
06593900
06594000
06594100
06594200
06594250
06594300
06594400
06594500
06594600
06594700
06594800
06594900
06595000
06595100

```

	TALLY:=1;	06595200
L1:	DS:=CHR;	06595300
RETURN:	X:=TALLY; CI:=C3;	06595400
		06595500
NEXT:	C2:=C1; GO TO START;	06595600
	IF SC=" " THEN	06595700
	BEGIN DS:=3 LIT"1 "; DS:=CHR;	06595800
	IF SC#"*" THEN DS:=3 LIT "3 "; GO BACK END;	06595850
	IF SC="*" THEN	06595900
	BEGIN SI:=SI-1;	06596000
	IF SC=" " THEN BEGIN SI:=SI+2; GO BACK END;	06596100
	SI:=SI+1; DS:=3 LIT"1 "; DS:=CHR;	06596200
	GO BACK;	06596300
	END;	06596400
	CI:=DI; DI:=LOC A;	06596500
	7(IF SC=DC THEN	06596600
	BEGIN DI:=C; SI:=SI-1;	06596700
	DS:=CHR; JUMP OUT TO BACK	06596800
	END ELSE SI:=SI-1);	06596900
	6(IF SC=DC THEN	06597000
	BEGIN DI:=C; SI:=SI-1; DS:=CHR;	06597100
	IF SC=" " THEN	06597200
	BEGIN SI:=SI+1;	06597300
	IF SC NEQ " " THEN DS:=LIT" ";	06597400
	SI:=SI-1;	06597500
	END ELSE DS:=2 LIT" ";	06597600
	JUMP OUT TO BACK;	06597700
	END ELSE SI:=SI-1);	06597800
	SI:=SI+1; DI:=C;	06597900
BACK:	CI:=C3;	06598000
		06598100
START:	N(IF SC="*" THEN	06598200
	BEGIN C3:=C1; CI:=C1; GO TO L END;	06598300
	IF SC="*" THEN	06598400
	BEGIN CI:=CI+LAS; JUMP OUT TO LA;	06598420
	X(DS:=LIT MARK; SI:=SI+1; JUMP OUT TO L);	06598440
	GO TO CM;	06598460
	END;	06598480
	X(DS:=CHR; JUMP OUT TO L);	06598500
	IF SC=ALPHA THEN IF SC LSS "0" THEN	06598600
	BEGIN Y(JUMP OUT TO AOK);	06598700
	DS=LIT" "; TALLY+1; Y←TALLY;	06598800
AOK:	DS:=CHR; GO TO L;	06598900
	END;	06599000
CM:	Y(DS:=LIT" "; TALLY:=0; Y:=TALLY);	06599100
	IF SC="S" THEN BEGIN TALLY:=4; JUMP OUT TO LAR END;	06599200
	IF SC="*" THEN BEGIN TALLY:=1; JUMP OUT TO LAR END;	06599300
	C3:=C1; CI:=C2;	06599400
L:	);	06599500
	N1(JUMP OUT TO YIT);	06599600
LAR:	SI:=SI-1;	06599700
	X(IF SC#"*" THEN DS=LIT"*" ELSE	06599800
	BEGIN DI:=DI-1; TALLY:=0; X:=TALLY END;	06599850
	JUMP OUT TO LL);	06599900
	Y(DS=LIT" ");	06600000
LL:	TALLY:=0;	06600100
LAR:	DS:=LIT LEFTARROW; DISC:=TALLY;	06600200
YIT:	DI:=DI; ADR:=SI; Z := DI;	06600300
	END;	06600400
	Q:=P; X:=P;	06600500

END	06600600
ELSE	06600700
BEGIN;	06600750
STREAM(DISC:=B,ADR,D,Z,Y;LAS:=TYPE,[2:1],N,N1,X,	06600800
CR:=TYPE,[FF]-1,LF:=TYPE,[CF]-(P(DUP)≠0),	06600850
RLF:=(TYPE=0),Q);	06600900
BEGIN SI:=ADR; DI:=D; TALLY:=0;	06601000
N(IF SC = LEFTARROW THEN	06601100
BEGIN	06601200
LAS(Z:=DI; DISC:=TALLY; JUMP OUT 2 TO L3);	06601300
DS:=LIT MARK; SI:=SI+1;	06601400
GO SETZ;	06601450
END ELSE	06601500
BEGIN IF SC="≠" THEN DS:=LIT"≠";	06601600
IF SC=" " THEN DS:=CHR ELSE	06601650
BEGIN DS:=CHR;	06601700
SETZ:  Z:=DI;	06601750
END;	06601800
END);	06601850
N1(JUMP OUT TO YIT);	06601900
L3:  DISC(SI:=LOC Z; SI:=SI+5; DI:=LOC Z;	06602000
IF 3 SC=DC THEN	06602020
BEGIN DI:=Q; X(Y:=TALLY);	06602040
JUMP OUT TO L2	06602060
END ELSE JUMP OUT);	06602080
DI:=Z; X(DS:=LIT"≠");	06602100
L2:  CR(DS:=4 LIT"1 ");	06602200
LF(DS:=4 LIT"1 ");	06602300
RLF(DS:=3 LIT"3 ");	06602350
DS:=LIT LEFTARROW;	06602400
Z:=DI;	06602500
YIT:  DI:=DI; ADR:=SI;	06602600
END;	06602700
Y:=P;	06602730
END;	06602760
Z:=P; D:=P; ADR:=P;	06602800
END	06602900
UNTIL P([T],SND) NEQ 8 OR (NUM:=NUM-63) LEQ 0;	06603000
OUTRANBIDS:=Z,[30:3]&(Z-B)[30:33:15];	06603100
T:=T&Y[2:47:1]&(IF TYPE,[1:1] THEN (X OR Q) ELSE NOT (TYPE,[FF]	06603200
= 0 OR TYPE,[CF] ≠ 0))[1:47:1];	06603300
ZIT; END OUTRAN BIDS;	06603400
REAL PROCEDURE OUTRANTC(ADR,NUM,TYPE,B,C);	06610000
VALUE ADR, NUM, TYPE;	06610100
REAL ADR, NUM, TYPE, B, C;	06610200
BEGIN REAL D, X, Y, Z, N1, N, Q;	06610300
B := SPACE(30)+1;	06610400
IF TYPE,[1:1] THEN	06610500
BEGIN;	06610600
STREAM(Y:=0,ADR:=N:=ADR,DI:=B);	06610700
BEGIN SI:=ADR;  DI:=D;	06610800
A:  IF SC = "≠" THEN	06611200
BEGIN DS:=CHR;	06611300
IF SC NEQ "≠" THEN	06611400
BEGIN TALLY:=1; Y:=TALLY; GO TO YIT END;	06611500
DS:=CHR; GO TO A;	06611600
END;	06611700
YIT:  DI:=DI;  ADR:=SI;	06611800
END STREAM;	06611900
DI:=P;  ADR:=P;  X:=Y:=P;	06612000

NUM:=NUM = ADR,[30:3]&(ADR=N)[30:33:15];	06612100
END ELSE %BCL SCAN	06612200
BEGIN;	06612300
STREAM(D:=Q:=B,X:=(NUM#0);LAS:=(NOT TYPE),[2:1],ADR);	06612400
BEGIN DII:=D;	06612500
X(SI:=ADR)	06612520
IF SC=LEFTARROW THEN	06612540
BEGIN CI:=CI+LAS; GO TO L;	06612560
TALLY:=0; XI=TALLY;	06612580
END ELSE	06612600
L: DS:=LIT"#");	06612620
D:=DI;	06612640
END STREAM;	06612700
XI=YI=P; DI=P;	06612800
END;	06612900
DO	06613000
BEGIN NI:=IF(NI=NUM GTR 63) THEN 63 ELSE NUM;	06613100
IF TYPE,[1:1] THEN %MORE SICASCII	06613200
BEGIN;	06613300
STREAM(DISC:=B,ADR,D,Z,X;LAS:=(NOT TYPE),[2:1],N,N1,C1:=0,C3:=0);	06613400
BEGIN SII:=ADR; DII:=D;	06613500
C1:=C1; GO TO START;	06613600
X(TALLY:=0; JUMP OUT TO L1);	06613700
TALLY:=1;	06613800
L1: DS:=CHR; XI=TALLY;	06613900
C1:=C3;	06614000
START: NCIF SC = "#" THEN	06614100
BEGIN C3:=C1; C1:=C1; GO TO L END;	06614200
IF SC = LEFTARROW THEN	06614300
BEGIN CI:=CI+LAS; JUMP OUT TO LA;	06614320
X(DS:=LIT MARK); GO TO LO;	06614340
END;	06614360
X(DS:=CHR; JUMP OUT TO L);	06614400
IF SC = ALPHA THEN	06614500
BEGIN IF SC="5" THEN GO TO LO;	06614600
IF SC="7" THEN GO TO LO;	06614700
END ELSE	06614800
BEGIN IF SC="&" THEN GO TO LO;	06614900
IF SC="≤" THEN GO TO LO;	06615000
IF SC="%" THEN GO TO LO;	06615100
IF SC="'" THEN GO TO LO;	06615200
IF SC="≥" THEN GO TO LO;	06615300
IF SC="#" THEN BEGIN TALLY:=1; JUMP OUT TO LAR END;	06615400
IF SC="\$" THEN BEGIN TALLY:=4; JUMP OUT TO LAR END;	06615500
END;	06615600
DS:=CHR; GO TO L;	06615700
LO: SII:=SI+1;	06615750
LI: ));	06615800
N1(JUMP OUT TO YIT);	06615900
LA: SII:=SI-1;	06616000
X(IF SC NEQ "#" THEN DS:=LIT "#"	06616100
ELSE BEGIN DII:=DI-1; TALLY:=0; XI=TALLY END);	06616200
TALLY:=0;	06616300
LAR: DS:=LIT LEFTARROW; DISC:=TALLY;	06616400
YIT: DI:=DI; ADR:=SI; Z:=DI;	06616500
END SICASCII STREAM;	06616600
XI=P;	06616700
END	06616800
ELSE %SCAN SOME MORE BCL	06616900
BEGIN;	06616950

```

STREAM(DISC:=8,ADR,D,Z,Y;LAS:=TYPE,[2:1],N,N1,X,CR:=TYPE,[FF]=1,06617000
  LF:=TYPE,Q); 06617100
BEGIN SI:=ADR; DI:=D; TALLY:=0; 06617200
  NCIF SC = LEFTARROW THEN 06617300
  BEGIN 06617400
    LAS(ZI:=DI; DISC:=TALLY; JUMP OUT 2 TO L3); 06617500
    SI:=SI+1; DS:=LIT MARK; 06617550
    GO TO SETZ; 06617600
  END ELSE 06617650
  BEGIN IF SC="*" THEN DS:=LIT"*"; 06617700
    IF SC=" " THEN DS:=CHR ELSE 06617750
    BEGIN DS:=CHR; 06617800
    SETZ; ZI:=DI; 06617850
    END; 06617900
  END); 06617950
  N1(JUMP OUT TO YIT); 06618000
L3: DISC(SI:=LOC Z; SI:=SI+5; DI:=LOC Z); 06618100
  IF 3 SC=DC THEN 06618120
  BEGIN DI:=Q; X(Y:=TALLY); 06618140
  JUMP OUT TO L2 06618160
  END ELSE JUMP OUT); 06618180
  DI:=Z; X(DS:=LIT "*"); 06618200
L2: CR(DS:=LIT "="); 06618300
  LF(DS:=LIT "*"); 06618400
  DS:= LIT LEFTARROW; 06618500
  ZI:=DI; 06618600
  YIT; DI:=DI; ADR:=SI; 06618700
END BCL STREAM; 06618800
YI=P; 06618830
END; 06618860
ZI=P; DI=P; ADR:=P; 06618900
END 06619000
UNTIL P([C],SND) NEQ 8 OR (NUM:=NUM-63) LEQ 0; 06619100
OUTRANTC:=Z,[30:3]&(Z=B)[30:33:15]; 06619200
C:=C&Y[2:47:1]&(IF TYPE,[1:1] THEN X ELSE 06619300
  NOT(TYPE,[FF]=0 OR TYPE,[CF] NEQ 0))[1:47:1]; 06619400
END OUTRANTC; 06619500
$ POP OMIT 06619501
% * * * * * %JS06845900
DEFINE KLUMP=@174#; % @173 IS RESERVED FOR THE DISK ADDRESS 07000000
COMMENT LASTCDNUM, FIRSTDECK, AND LASTDECK ARE STORED IN THE 07000010
FIRST THREE WORDS OF THE DISK SEGMENT LOCATED AT DIRECTORYTOP 07000015
+3. IN A NON SHARED DISK SYSTEM, THEY ARE WRITTEN OUT EACH 07000020
TIME ONE OF THEM IS CHANGED SO THAT THEY WILL BE PRESERVED 07000025
IF A HALT/LOAD OCCURS. IN A SHARED DISK SYSTEM, THEY ARE 07000030
READ INTO THE PRT WITH A READ=LOCK COMMAND EACH TIME THEY ARE 07000035
USED, THIS PROVIDES CONTROL DECK INTERLOCKING BETWEEN SYSTEMS 07000040
IN ADDITION TO PASSING THE INFORMATION BETWEEN SYSTEMS, 07000045
END COMMENT; 07000050
INTEGER LASTCDNUM=@174; 07000100
REAL FIRSTDECK=@175; 07000200
REAL LASTDECK=@176; 07000300
DEFINE LOCKCONTROLDECKS=BEGIN SLEEP([TOGGLE],CDMASK); LOCKTOG(CDMASK); 07001000
$ SET OMIT = NOT(SHAREDISK) 07001099
  DISKWAIT(=KLUMP,-3,DIRECTORYTOP+3); 07001100
$ POP OMIT 07001101
END#; 07001200
UNLOCKCONTROLDECKS=BEGIN 07001300
$ SET OMIT = NOT(SHAREDISK) 07001399
  DISKWAIT(KLUMP,-3,DIRECTORYTOP+3); 07001400

```



```

$ POP OMIT
      UNLOCKTOG(CDMASK) END#;
REAL PROCEDURE NEXTCDNUM(UPDATE); VALUE UPDATE; BOOLEAN UPDATE;
BEGIN
  LOCKCONTROLDECKS;
  LASTCDNUM := (LASTCDNUM MOD 9999) + 1;
  STREAM(CDNUM:=0; LNUM:=LASTCDNUM);
  BEGIN
    SI:=LOC LNUM; DI:=LOC CDNUM; DS:=8 DEC;
  END;
  NEXTCDNUM := P;
  IF UPDATE THEN
  BEGIN
    DISKWAIT(KLUMP,-3,DIRECTORYTOP+3);
    UNLOCKTOG(CDMASK);
  END;
END;
PROCEDURE STARTADECK(N); VALUE N; REAL N; FORWARD;
PROCEDURE ENTERCONTROLDECK(H); VALUE H; ARRAY H[*]; FORWARD;
REAL RUNNUMBER;%
PROCEDURE COM23;%
  BEGIN%
    REAL INBUFF,% ADDRESS OF THE INPUT BUFFER,
    OUTBUFF,% " " " OUTPUT BUFFER,
    FIRSTCARD,% " " " CARD IMAGE OF THE FIRST CARD
    OUTBUFFOLD,% " " " LAST OUTPUT BUFFER,
    RESERVE,% " " 30 WDS OF CORE USED TO BUILD THE
    T,T1,T2,% TEMPORARY VARIABLES,
    R,L,N,% " " USED TO COUNT CARD IMAG
    Q,% USUALLY INDICATES COL 1 HAS A QUESTION MARK
    IU,% UNIT NUMBER OF THE INPUT UNIT,
    OU,% " " " " OUTPUT UNIT,
    FIRST,% TRUE IF THE FIRST CARD OF A DECK,
    S,% USED AS A TEMPORARY VARIABLE IN SUBROUTINE
    % AND TO HAND THE UNIT NUMBER TO SUBROUTINE S
    D;% USED AS A MASK TO SLEEP UNTIL DISK OPERATIO
    % ARE COMPLETED,
$ SET OMIT = NOT(PACKETS)
    REAL VERYFIRST, %TRUE IF THE FIRST CARD OF THE FIRST DECK
    %IN SINCE LOAD CONTROL WAS EXECUTED,
    FIRSTORSEC, %TRUE IF THE FIRST OR SEC. CARD OF NEW DECK
    PTYPE,% CONTAINS THE RESULT OF REAL SUBROUTINE
    % PACKETCARD,SAVING SOME NEEDLESS EXTRA
    % CALLS ON IT TO CHECK THE TYPE OF A CARD,
    % THE VALUE OF PTYPE IS AS FOLLOWS:
    %
    % 0 = NOT A PACKET CONTROL CARD
    % 1 = "PACKEND" CARD, (USED BY THE
    % OPERATORS TO END A GROUP
    % OF PACKETS BEING LOADED TO
    % DISK)
    % 3 = "PACKET" CARD,(FIRST CARD
    % OF A PACKET)
    % 5 = "END PACKETS" CARD, (USED BY
    % THE OPERATORS TO BOTH END
    % A GROUP OF PACKETS AND
    % SIMULTANEOUSLY DISCONTINUE
    % LOAD CONTROL),
    PLUGGED;% TRUE IF THE LAST "PACKET" CARD(I.E.,
    % PTYPE=3), WAS BOTH THE START OF A NEW
    % PACKET AND WAS USED TO "PLUG" THE END

```

```

07001401
07001500
07001600
07001620
07001640
07001660
07001680
07001700
07001720
07001740
07001760
07001780
07001800
07001820
07001840
07001860
07001880
07002000
07002100
07003000
07004000
07005000
07006000
07006010
07006020
07006030
07006040
07006050
07006060
07006070
07006080
07006090
07006100
07006140
07006150
07006160
07006161
07006169
07006172
07006174
07006176
07006180
07006190
07006200
07006210
07006220
07006230
07006240
07006250
07006260
07006270
07006280
07006290
07006300
07006310
07006320
07006330
07006340
07006350
07006360

```

%	OF THE LAST PACKET WITH AN ARTIFICIAL	07006370
%	"-QUESTION MARK" PACKET," CARD;	07007000
\$ POP OMIT		07007001
BOOLEAN CDONLY;		07007100
INTEGER A,I;%		07008000
\$ SET OMIT = NOT(PACKETS)		07008199
REAL CONTINUE,DISKCHAIN,ADECK; LABEL DK;		07008200
LABEL INPUTL;		07008300
\$ POP OMIT		07008301
LABEL AGAIN,INL,ERROR,SUPER,BOMB,SKIPIT,EXIT;		07009000
ARRAY FPB[*],H[*];%		07010000
SUBROUTINE STOP;%		07011000
BEGIN IF S ≠ 18 THEN%		07012000
BEGIN READY ← NOT(Q ← TWO(S)) AND READY;%		07013000
RRRMECH ← NOT Q AND RRRMECH OR Q AND SAVEWORD;%		07014000
LABELTABLE[S] ← @114;%		07015000
RDCTABLE[S] ← MULTITABLE[S] ← 0%		07016000
END;%		07017000
FPBIT+1] ← *P(DUP)+CLOCK+P(RTR);%		07018000
FPBIT],[24:12] ← TINU[S],[18:12];%		07019000
TINU[S],[18:12]:=0;		07020000
END;%		07021000
\$ SET OMIT = PACKETS		07021999
SUBROUTINE FORGETIT;%		07022000
\$ POP OMIT		07022001
\$ SET OMIT = NOT(PACKETS)		07022099
SUBROUTINE FORGETONE;		07022100
\$ POP OMIT		07022101
BEGIN T1 ← H[9]+9;%		07023000
FOR T2 ← 10 STEP 1 UNTIL T1 DO%		07024000
FORGETUSERDISK(H[T2],-H[8]);		07025000
END;%		07026000
\$ SET OMIT = NOT(PACKETS)		07026099
SUBROUTINE FORGETIT;		07026100
BEGIN FORGETONE;		07026200
WHILE DISKCHAIN NEQ 0 DO		07026300
BEGIN DISKWAIT(=(H INX 0),30,DISKCHAIN);		07026400
DISKCHAIN:=H[6],[FF];		07026500
FORGETONE;		07026600
END;		07026700
END FORGETIT;		07026800
\$ POP OMIT		07026801
SUBROUTINE BOMBTIME;%		07027000
BEGIN WHILE PRTRROW[P1MIX],[PSF]>1 DO		07028000
STOPM;		07028100
IF TERMSET(P1MIX) THEN GO BOMB;		07028800
END;		07028900
\$ SET OMIT = NOT(PACKETS)		07028999
REAL SUBROUTINE PACKETCARD;% THIS USED TO BE "ENDCARD"		07029000
BEGIN IF Q THEN%		07030000
BEGIN;%		07031000
STREAM(X:="PACKETS"Y:="CONTINU",Z:="END,"INBUFF);		07032000
BEGIN SI ← INBUFF;%		07033000
LI SI ← SI+1; IF SC = " " THEN GO TO L;%		07034000
DI←LOC X; DI←DI+1; POINT TO "PACKETS"		07035000
IF 4SC=DC THEN% A "PACKET" OR "PACKEND" CARD		07036000
IF 2SC=DC THEN TALLY←3% A "PACKET" CARD		07036100
ELSE TALLY←1% A "PACKEND" CARD		07036150
ELSE BEGIN DI←DI+4; POINT TO "PACKETS"		07036200
IF 7 SC=DC THEN TALLY:=5 %"END PACKET"		07036210

```

ELSE BEGIN SI:=SI-11; DI:=LOC Y; 07036220
      DI:=DI+1; IF 7SC=DC THEN TALLY:=6 07036230
      ELSE BEGIN DI:=LOC Z; DI:=DI+1; 07036240
      SI:=SI-7; 07036250
      IF 3SC=DC THEN TALLY:=7; 07036260
      END; 07036270
    END; 07036400
  END; 07037000
  X ← TALLY; 07038000
  END; 07039000
END ELSE P(0); 07039100
PTYPE:=P; 07039200
IF PTYPE=6 THEN BEGIN PTYPE:=3; CONTINUE:=TRUE END 07039300
ELSE CONTINUE:=FALSE; 07039410
IF PTYPE=7 THEN PTYPE:=ADECK; 07040000
PACKETCARD:=PTYPE; 07041000
END; 07041001
$ POP OMIT 07041099
$ SET OMIT = PACKETS 07041100
  BOOLEAN SUBROUTINE ENDCARD; 07041110
  BEGIN IF Q THEN 07041120
    BEGIN M[INBUFF+9]:=0&"",[1:43:5]; 07041130
    STREAM(X:="END";INBUFF); 07041140
    BEGIN SI:=INBUFF; 07041150
    L: SI:=SI+1; IF SC=" " THEN GO TO L; 07041160
    DI:=LOC X; DI:=DI+5; 07041170
    IF 3 SC=DC THEN TALLY:=1; 07041180
    X:=TALLY; 07041190
    END; 07041200
  END ELSE P(0); 07041210
  ENDCARD:=P; 07041220
  END; 07041221
$ POP OMIT 07042000
  REAL SUBROUTINE ADR; 07043000
  BEGIN IF (T2 + H[(T1 + R DIV 200)+10]) = 0 THEN 07044000
    BEGIN H[9] ← T1+1; 07045000
    H[T1+10] ← T2 + GETUSERDISK(200); 07046000
    END; 07047000
  ADR ← R MOD 200+T2 07048000
  END; 07049000
  SUBROUTINE INPUT; 07050000
  BEGIN 07051000
    IF IU < 16 THEN 07051099
      $ SET OMIT = NOT(PACKETS) 07051100
      BEGIN 07051110
        INPUTL: T ← WAIT10 (@120540000000 + INBUFF, 07051120
          @6000040, IU); 07051130
        IF T=@40 THEN GO TO INPUTL; 07051140
        IF T#0 THEN 07051150
          BEGIN 07051160
            P(DEL); 07051170
            GO TO ERROR 07051180
          END; 07051181
        END; 07051999
      END; 07052000
    END; 07053000
  END; 07054000
  T ← WAIT10(@140000000 + INBUFF,% 07055000
    @40,IU); 07056000
  T ← WAIT10(@12054000000 + INBUFF,% 07056001
    @2000000,IU);
$ POP OMIT
$ SET OMIT = PACKETS
  BEGIN IF FIRST THEN
    T ← WAIT10(@140000000 + INBUFF,%
      @40,IU);
    T ← WAIT10(@12054000000 + INBUFF,%
      @2000000,IU);
$ POP OMIT

```

```

Q ← M(INBUFF-1)≡9;% 07057000
END% 07058000
ELSE BEGIN WHILE(Q←WAITIO(@40000000+INBUFF,FIRST×4+ 07059000
@4000000,IU)),[45:1] DO 07059100
IF FIRST AND CONLY THEN GO EXIT ELSE 07059110
BEGIN SLEEP([TOGGLE],STATUSMASK); 07059200
RRRMECH←RRRMECH AND NOT Q←TWO(IU); 07059300
READY←READY AND NOT Q; 07059400
DO BEGIN CLICK←CLOCK+P(RTR)+64; 07059500
SLEEP([READY],Q); 07059600
BOMBTIME; 07059700
END UNTIL (READY AND Q)≠0; 07059800
END; 07060000
IF Q ← Q ≠ 0 THEN 07061000
UNIT[IU],[5:13]←0; 07066000
T ← 0;% 07067000
END;% 07068000
END;% 07069000
% SET UP INPUT VARIABLES% 07071000
$ SET OMIT = NOT(PACKETS) 07071899
OU←PSEUDOMIX[P1MIX]; 07071900
PSEUDOMIX[P1MIX]←0; 07071910
$ POP OMIT 07071911
IF CONLY:=(PRT[P1MIX,@25]>22) THEN 07072000
BEGIN IU:=PRT[P1MIX,@25]; 07072100
PRT[P1MIX,@25]:=0; % DISK 07072200
END ELSE 07072300
BEGIN IF (IU:=FINDINPUT("CONTROL","DECK ",0,0,0,0,0,0, 07072400
0,0)) LSS 0 THEN GO INITIATE; % BEEN DS=ED 07072500
IF IU GEQ 32 THEN P(XIT); % EOJ IF PSEUDODCK 07072600
END; 07072700
$ SET OMIT = NOT(PACKETS) 07072899
PSEUDOMIX[P1MIX]←OU; 07072900
$ POP OMIT 07072901
STARTIMING(0,IU); 07073000
FPB:=PRT[P1MIX,3]; 07073500
IF NOT(JAR[P1MIX,9],[2:1]) THEN % DONT SUPPRESS MESSAGE 07074090
FILEMESSAGE(" IN "&TINU[IU][6:30:18],0, 07074100
"CONTROL","DECK ",0,0,0,0,OPNMESS OR OPENK); 07074200
RDCTABLE[IU],[8:6] ← P1MIX;% 07075000
IF IU LSS 16 THEN 07076000
BEGIN FPB[3],[23:1]:=1; % SET INPUT FLAG FOR LOG 07076010
T:=WAITIO(@540000005,0,IU); 07077000
END 07077010
ELSE IF IU=23 AND READERA NEQ 0 THEN 07078000
BEGIN FORGETSPACE(READERA=2);% 07079000
READERA ← 0;% 07080000
END% 07081000
ELSE IF IU=24 AND READERB NEQ 0 THEN 07082000
BEGIN FORGETSPACE(READERB=2);% 07083000
READERB ← 0;% 07084000
END;% 07085000
INBUFF ← GETSPACE(11,0,1) + 2; 07086000
FIRSTCARD ← GETSPACE(10,0,1)+2;% 07087000
% SET UP OUTPUT VARIABLES% 07088000
IF PRT[P1MIX,@25] THEN% 07089000
BEGIN OU ← LABELASCRATCH(T +% 07090000
TAPELABEL("CONTROL","DECK ",1,1,100));% 07091000
FORGETSPACE(T);% 07093000
FPB[3],[23:1]:=0; % SET OUTPUT FLAG FOR LOG 07093010

```

END%	07094000
ELSE BEGIN OUTBUFFOLD + OUTBUFF + GETSPACE(60,0,1)+2;%	07095000
RESERVE + GETSPACE(30,0,1)+2;%	07096000
H + [M[GETSPACE(30,0,1)+2]]&30[8:38:10];%	07097000
OU + 18;%	07098000
END;%	07103000
STARTIMING(5,OU);	07104000
FPB:=PRT[P1MIX,3];	07104500
% STARTIMING MAY HAVE MOVED IT.	07105499
\$ SET OMIT = NOT(PACKETS)	07105500
VERYFIRST+1;%	07105501
\$ POP OMIT	07106000
% BEGIN ONE DECK%	07107000
AGAIN!	07108000
OUTBUFF + OUTBUFFOLD;%	07108099
L + N + 0;%	07108100
\$ SET OMIT = NOT(PACKETS)	07108101
FIRST+FIRSTORSEC+D+1; ADECK+0;	07108999
\$ POP OMIT	07109000
\$ SET OMIT = PACKETS	07109001
FIRST + D + 1;	07109001
\$ POP OMIT	07110000
IF OU = 18 THEN%	07111000
BEGIN H[ 9] + 0;%	07112000
MOVE(20,[H[9]], [H[10]]);	07112100
H[8]+200;	07113000
END;%	07114000
% BEGIN ONE CARD%	07115000
INL!	07115099
\$ SET OMIT = NOT(PACKETS)	07115100
IF PTYPE NEQ 3 OR VERYFIRST THEN	07115101
\$ POP OMIT	07115200
INPUT;	07115499
\$ SET OMIT = NOT(PACKETS)	07115500
IF FIRSTORSEC THEN%	07115501
\$ POP OMIT	07116000
IF FIRST THEN%	07117000
BEGIN	07117099
\$ SET OMIT = NOT(PACKETS)	07117100
PLUGGED:=VERYFIRST;	07117101
\$ POP OMIT	07117199
\$ SET OMIT = PACKETS	07117200
FIRST:=FALSE;	07117201
\$ POP OMIT	07118000
MOVE(10,INBUFF,FIRSTCARD);%	07118099
\$ SET OMIT = NOT(PACKETS)	07118100
IF PACKETCARD LSS 3 AND PKTONLY	07118200
THEN BEGIN VERYFIRST+2;%	07118300
GO TO ERROR%	07118400
END ELSE % INV PKT CARD	07118500
IF PTYPE=5 THEN	07118510
IF OU<16 THEN FIRST:=VERYFIRST:=0 ELSE	07118520
GO TO EXIT ELSE	07118550
IF PTYPE#3 OR CONTINUE THEN	07118600
BEGIN	07118690
ADECK:=1; GO DK;	07118700
END;	07119000
END ELSE% THIS MUST BE THE SECOND CARD IN	07119001
\$ POP OMIT	07119009
\$ SET OMIT = PACKETS	07119010
END;	07119011
\$ POP OMIT	

\$ SET OMIT = NOT(PACKETS)	07119099
DK: IF Q THEN FIRSTORSEC:=0 ELSE%BAD SEC./FIRST	07119100
BEGIN VERYFIRST+4; % CARD	07119200
GO TO ERROR;%	07119300
END;% INV DECK SET=UP	07119400
\$ POP OMIT	07119401
IF T NEQ 0 THEN	07120000
\$ SET OMIT = NOT(PACKETS)	07120009
IF PTYPE NEQ 3 OR VERYFIRST THEN	07120010
\$ POP OMIT	07120011
GO TO ERROR;	07120020
BOMBTIME;%	07121000
IF OU < 16 THEN%	07122000
BEGIN	07122010
\$ SET OMIT = NOT(PACKETS)	07122999
PLUGGED + VERYFIRST OR (PACKETCARD#3)	07123500
OR FIRST;	07124000
IF PLUGGED THEN	07124500
\$ POP OMIT	07124501
T=WAITIO(INBUFF#@5000[18:33:15]	07125000
&(10-Q)[8:38:10],0,OU);	07125500
\$ SET OMIT = PACKETS	07125599
IF NOT ENDCARD THEN GO TO INL;	07125600
\$ POP OMIT	07125601
\$ SET OMIT = NOT(PACKETS)	07125999
IF VERYFIRST THEN VERYFIRST+PTYPE+0;	07126000
IF FIRST THEN FIRST+PTYPE+0;	07126500
IF PTYPE=0 THEN GO TO INL;	07127000
\$ POP OMIT	07127001
M[INBUFF=1] + @17370000000000000;	07127500
T + WAITIO(INBUFF=1,0,OU);	07128000
SUPER;	07129000
\$ SET OMIT = NOT(PACKETS)	07129099
IF PTYPE=5 THEN GO TO EXIT;	07129100
IF PTYPE=1 THEN VERYFIRST:=TRUE;	07129200
GO TO AGAIN;	07129300
\$ POP OMIT	07129301
\$ SET OMIT = PACKETS	07129999
STREAM(X:="CONTROL";INBUFF);	07130000
BEGIN SI:=INBUFF;	07130100
E: IF SC NEQ "E" THEN	07130200
BEGIN SI + SI+1; GO TO E END;%	07131000
SI + SI+3;%	07132000
L: IF SC = " " THEN%	07133000
BEGIN SI + SI+1; GO TO L END;%	07134000
DI + LOC X; DI + DI+1;%	07135000
IF 7 SC = DC THEN X + TALLY;%	07136000
END;%	07137000
IF P # 0 THEN GO TO AGAIN ELSE GO TO EXIT;%	07138000
\$ POP OMIT	07138001
END;%	07139000
IF D = 0 THEN SLEEP([D],NOT 0);	07139500
\$ SET OMIT = NOT(PACKETS)	07139509
IF PACKETCARD NEQ 0 AND NOT(ADECK AND PTYPE=1) THEN	07139510
BEGIN IF NOT(PLUGGED OR FIRST) THEN%	07139511
BEGIN STREAM(D+OUTBUFF); BEGIN DS+27 LIT	07139512
"CC END...IN CASE YOU FORGOT";DS+45LIT" " END;	07139513
IF NOT PKTONLY THEN IF PTYPE=3 AND NOT CONTINUE AND NOT ADECK THEN	07139520
BEGIN STREAM(FIRSTCARD,T+T+SPACE(13));	07139530
BEGIN DS+24LIT"#NO PACKEND CARD, PKT = "; SI+FIRSTCARD;	07139540

DS=9 WDS; DS=LIT"+";	07139550
END;	07139560
SPOUT(T);	07139565
END;	07139570
END ELSE MOVE(10,INBUFF,OUTBUFF);%	07139575
END ELSE%	07139590
\$ POP OMIT	07139591
MOVE(10,INBUFF,OUTBUFF);%	07140000
\$ SET OMIT = NOT(PACKETS)	07140099
IF VERYFIRST THEN PLUGGED=0;%	07140100
\$ POP OMIT	07140101
IF Q THEN%	07141000
BEGIN IF L DIV 6 ≠ N DIV 6 THEN%	07142000
BEGIN R = L DIV 3;%	07143000
A = ADR;%	07144000
DISKWAIT(-RESERVE,30,A);	07145000
M[I=L MOD 3×10+9+RESERVE] = N;%	07147000
DISKWAIT(RESERVE,30,A);	07148000
END%	07150000
ELSE M[I = (L-N)×10+9+OUTBUFF] = N;%	07151000
L = M[OUTBUFF+9] = N;%	07152000
END;%	07153000
IF N = 12000 THEN%	07154000
BEGIN T := SPACE(14);	07155000
STREAM(FIRSTCARD,T);	07156000
BEGIN DS = 32 LIT%	07157000
\$ SET OMIT = NOT(PACKETS)	07157099
"#MORE THAN 12000 CARDS IN PKT = ";	07157100
\$ POP OMIT	07157101
\$ SET OMIT = PACKETS	07157999
"#MORE THAN 12000 CARDS IN	07158000
\$ POP OMIT	07158001
SI=FIRSTCARD;DS=9WDS;DS=LIT "+";	07159000
END;%	07160000
GO TO SKIPIT;	07161000
END;%	07162000
IF (N = N+1) MOD 6 = 0 THEN%	07163000
BEGIN R = N DIV 3-2;%	07164000
A = ADR;%	07165000
OUTBUFF = OUTBUFFOLD;%	07166000
DISKIO(D,OUTBUFF=1,60,A);	07167000
END ELSE OUTBUFF = OUTBUFF+10;%	07168000
\$ SET OMIT = NOT(PACKETS)	07169000
IF FIRST THEN FIRST=PTYPE=0;%	07169099
IF VERYFIRST THEN VERYFIRST:=PTYPE:=0;	07169100
\$ POP OMIT	07169110
\$ SET OMIT = NOT(PACKETS)	07169201
IF PTYPE=0 THEN GO INL;	07169499
\$ POP OMIT	07169500
\$ SET OMIT = PACKETS	07169501
IF NOT ENDCARD THEN GO TO INL;%	07169999
\$ POP OMIT	07170000
IF D = 0 THEN SLEEP([D],NOT 0);	07170001
OUTBUFF = OUTBUFFOLD;%	07171000
R = N DIV 6×2;%	07172000
A = ADR;%	07173000
IF N MOD 6 ≠ 0 THEN	07174000
BEGIN	07175000
	07175100
	07175200

DISKWAIT(OUTBUFF,60,A);	07176000
END;%	07178000
IF R+2 < 200 THEN	07178100
BEGIN H[8] ← R+2;	07178200
FORGETUSERDISK(A+2,R=19B);	07178300
END;	07178400
H[7]←N=1;	07179000
H[4]←H[6]←0;	07179050
H[5]←= 0;	07179100
\$ SET OMIT = NOT(PACKETS)	07179199
H[6]←0&DISKCHAIN[CTF]&(IF IU<23 THEN 2 ELSE IU=23)	07179200
[2:42:6];	07179202
IF CONTINUE THEN	07179205
BEGIN	07179210
H[2]←NEXTCDNUM(1);	07179220
DISKCHAIN←GETESPDISK;	07179230
DISKWAIT(H INX 0,30,DISKCHAIN);	07179250
STREAM(A←H[2],B←FIRSTCARD,INBUFF);	07179260
BEGIN SI←B; DSI←8 CHR;	07179270
DS←15 LIT "CONTINUES PKT#";	07179280
SI←LOC A; SI←SI+4; DS←4 CHR; DS←LIT " ";	07179290
END;	07179300
END ELSE	07179310
BEGIN DISKCHAIN←0;	07179320
\$ POP OMIT	07179321
ENTERCONTROLDECK(H);	07180000
\$ SET OMIT = NOT(PACKETS)	07180009
END;	07180010
\$ POP OMIT	07180011
GO TO SUPER;	07181000
ERROR:	07214000
T ← SPACE(12);	07214099
\$ SET OMIT = NOT(PACKETS)	07214100
STREAM(FIRSTCARD,X←VERYFIRST,T);%	07214110
BEGIN SI←LOC X; SI←SI+7; IF SC="2" THEN	07214120
DS←16 LIT "#INV PKT CARD = %"	07214130
ELSE IF SC="4" THEN%	07214140
DS←16 LIT "#INV DECK,PKT = %"	07214150
ELSE DS←16 LIT "#READ ERR,PKT = %";	07214151
\$ POP OMIT	07214999
\$ SET OMIT = PACKETS	07215000
STREAM(FIRSTCARD,T);%	07216000
BEGIN DS ← 16 LIT "#READ ERROR FOR %";	07216001
\$ POP OMIT	07217000
SI ← FIRSTCARD; DS ← 9 WDS; DS ← LIT " ";%	07218000
END;%	07219000
SKIPIT: SPOUT(T);	07220000
DO BEGIN INPUT;%	07221000
BOMBTIME;%	07221999
\$ SET OMIT = PACKETS	07222000
END UNTIL ENDCARD;%	07222001
\$ POP OMIT	07222099
\$ SET OMIT = NOT(PACKETS)	07222100
END UNTIL PACKETCARD NEQ 0;	07222101
\$ POP OMIT	07223000
IF OU < 16 THEN%	07224000
BEGIN DO BEGIN T ← WAITIO(@340000005,@60,OU);%	07225000
BOMBTIME;%	07226000
END UNTIL T,[42:1];%	07227000
T ← WAITIO(@140000005,@60,OU);%	07228000
END%	



	ELSE FORGETIT;%	07229000
	GO TO SUPER;%	07230000
BOMB;%	FORGETIT;%	07231000
EXIT;	SLEEP([TOGGLE],STATUSMASK);	07232000
	IF IU GEQ 23 THEN UNITCODE[IU=23]:=(-0);	07232500
	S ← IU; T ← 3; STOP;%	07233000
	S ← OU; T ← 8; STOP;%	07234000
	FORGETSPACE(INBUFF);%	07235000
	FORGETSPACE(FIRSTCARD);%	07236000
	IF OU > 16 THEN%	07237000
	BEGIN FORGETSPACE(H);%	07238000
	FORGETSPACE(OUTBUFFOLD);%	07239000
	FORGETSPACE(RESERVE);%	07240000
	END;%	07241000
	END COM23;%	07242000
	PROCEDURE STARTLOADN(KTR); VALUE KTR; REAL KTR;%	07243000
	BEGIN REAL I,HDR,SEGO,F,C,T; ARRAY SHEAT[*];	07244000
	LABEL TRYAGAIN,LDCNTRL,DISK;	07244100
	STREAM(K←0;KTR);%	07245000
	BEGIN SI ← KTR;%	07246000
	LI IF SC = " " THEN%	07247000
	BEGIN SI ← SI+1; GO TO L END;%	07248000
	DI ← LOC K; DI ← DI+6; DS ← 2 CHR;%	07249000
	END;%	07250000
	C ← P;%	07251000
	T ← KTR,[15:15]=1;%	07252000
	IF (C NEQ "MT" AND C NEQ "DK") OR	07253000
	(C = "DK" AND CDONLY ) THEN	07253100
	SPOUT(T)	07254000
	ELSE BEGIN C ← C = "MT";%	07255000
TRYAGAIN;	IF (HDR:=DIRECTORYSEARCH(P(LDCNTRL),P(DISK),3)) ≠ 0 THEN	07255100
	BEGIN	07256000
	SHEAT := [M[F:=GETSPACE(31,64,0)+2]] & 30[8:38:10];	07256200
	STREAM(S:=F-1, D:=F); % ZERO OUT THE SHEAT ENTRY	07256400
	BEGIN	07256600
	SI:=S; DS:=30 WDS;	07256800
	END;	07257000
	SEGO := GETSPACE(30,64,0)+2;	07257200
	DISKWAIT(=SEGO, 30, M[HDR INX 10]);	07257400
	F,[FF] := HDR; % CORE ADDRESS OF HEADER IN [FF] OF PARAM,	07257600
	SHEAT[7] := SEGO; % CORE ADRS,OF SEGMENT ZERO IN SHEAT[7]	07257800
	SHEAT[0] := SHEAT[14] := P(LDCNTRL);	07258000
	SHEAT[1] := P(DISK);	07258200
	SHEAT[2] := 0 & 5[8:38:10];	07258400
	% [4:1] IN SHEET[2] MEANS SUPRESS BOJ/EOJ MESSAGES	07258600
	SHEAT[16] := SHEAT[17] := @377777777777; % TIME LIMITS	07258800
	SHEAT[19] := C; % COMMON VALUE	07259000
	SHEAT[20] := 4; % CORE ESTIMATE	07259200
	SHEAT[21] := 150; % STACK SIZE	07259400
	STREAM(A:=0 ; S := P(,SCHEDULEIDS));	07259600
	BEGIN	07259800
	SI:=S;	07260000
	47(SKIP SB; SKIP DB; TALLY:=TALLY+1;	07260200
	IF SB THEN ELSE JUMP OUT);	07260400
	DSI:=SET; AI:=TALLY;	07260600
	END STREAM STATEMENT;	07260800
		07261000
		07261200
		07261400
	I := P;	07261600

SHEAT[3],[8:10] := 1;	% SCHEDULE NUMBER	07261800
SHEAT[23] := (CLOCK + P(RTR)) DIV 60;		07262000
SHEAT[24] := MCP;		07262100
SHEAT[25] := HDR,[FF];	% DISK ADDRESS OF FILE HEADER	07262200
SHEAT[26] := -31;	% LOGLINE	07262400
STREAM(C, T);		07262600
BEGIN		07262800
DSI:=27LIT"CC RUN LDCNTRL/DISK;COMMON=";		07263000
SII:=LOC C; DSI:=8DEC;		07263200
DSI:=6LIT";END.+";		07263400
END STREAM STATEMENT;		07263600
SHEAT[6] := GETESPDISK;		07263800
DISKWAIT(T, 10, SHEAT[6]);		07264000
FORGETSPACE(T);		07264200
FORK(P(,SELECTRUN), F, 0, 160, 0);		07264400
END ELSE % IF IN DIRECTORY		07265000
BEGIN		07265100
ENTERSYSFILE(2);		07265200
GO TRYAGAIN;		07265300
LDCNTRL::: "LDCNTRL";		07265400
DISK::: "DISK ";		07265500
END;		07265600
END;%		07266000
END;%		07267000
PROCEDURE TABLEOFCONTENTS(B,COUNT);%		07268000
VALUE B,COUNT; REAL B,COUNT;%		07268100
BEGIN REAL I,T,N,A,TUSTA;		07269000
% SET OMIT = NOT(PACKETS)		07269099
REAL FIRST,START,FINAL,PKTCT;%		07269100
% POP OMIT		07269101
LABEL L,EXIT,G;%		07270000
% SET OMIT = NOT(SHAREDISK)		07270099
REAL SYS;		07270100
IF (SYS:=UNITIN(TINU,B)-37)>3 OR SYS LSS 0 THEN		07270200
BEGIN		07270300
STREAM(AI="ALL";B);		07270400
BEGIN SII:=B;		07270500
63(IF SC#" " THEN JUMP OUT;SII:=SII+1);		07270600
DII:=LOC A;DII:=DII+5;IF 3 SC=DC THEN		07270700
TALLY:=5;AI=TALLY;		07270800
END;		07270900
IF NOT (SYS:=P) THEN SYS:=SYSNO;		07271000
END;		07271100
% POP OMIT		07271101
AI:=B,[15:15]=1;		07271900
TUSTA:=M[A=1];		07272000
LOCKCONTROLDECKS;		07272500
AI:=FIRSTDECK;		07273000
% SET OMIT = NOT(PACKETS)		07273099
FIRST+1;%		07273100
% POP OMIT		07273101
LI :=SPACE(13);		07274000
GI :=IF A = 0 THEN GO TO EXIT;%		07275000
DISKWAIT(=I,12,A);		07276000
AI:=M[I+6],[CF];		07278000
% SET OMIT = NOT(SHAREDISK)		07279000
IF SYS LSS 5 THEN		07279100
IF SYS#M[I+4],[4:2] THEN GO TO G;		07279200
TI:=M[I+4],[4:2]+1;		07280000
% POP OMIT		07280001

N ← M[I+2];%	07281000
\$ SET OMIT = NOT(PACKETS)	07281099
IF NOT COUNT THEN%	07281100
BEGIN%	07281200
\$ POP OMIT	07281201
DISKWAIT(-I-3,9,M[I+10]);	07282000
STREAM(N,T,I);	07284000
BEGIN SI ← LOC N; SI ← SI+1;%	07285000
\$ SET OMIT = NOT(PACKETS)	07285099
DS:=8 LIT " PACKET ";DS:=5 CHR;	07285100
\$ POP OMIT	07285111
\$ SET OMIT = PACKETS	07285999
DS ← 6 LIT " DECK "; DS ← 5 CHR;%	07286000
\$ POP OMIT	07286001
\$ SET OMIT = NOT(SHAREDISK)	07286100
DS:=6 LIT" (SYS ";SI:=SI+9;DS:=CHR;	07286200
DI:=DI-1;SKIP DB;DS:=SET;DS:=LIT")";	07286300
\$ POP OMIT	07286301
\$ SET OMIT = SHAREDISK	07286400
DS:=8 LIT" ";	07286500
\$ POP OMIT	07286501
\$ SET OMIT = NOT(PACKETS)	07286509
DS:=3 LIT " =";	07286510
\$ POP OMIT	07286511
\$ SET OMIT = PACKETS	07286999
DS ← 5 LIT " IS: ";%	07287000
DI ← DI+40; DI ← DI+32; DS ← LIT "+";%	07288000
\$ POP OMIT	07288001
\$ SET OMIT = NOT(PACKETS)	07288099
DI:=DI+40;DI:=DI+19;DS:=LIT"+";	07288100
\$ POP OMIT	07288101
END;%	07289000
SPOUT(I);%	07290000
\$ SET OMIT = NOT(PACKETS)	07290099
END ELSE%	07290100
BEGIN% OPERATOR WANTS A COUNT	07290200
IF FIRST THEN BEGIN% STORE FIRST DECK #.	07290300
FIRST←0; START←N;%	07290400
END;%	07290500
PKTCT←PKTCT+1; FINAL←N;%	07290600
FORGETSPACE(I);%	07290650
END;%	07290700
\$ POP OMIT	07290701
GO TO L;%	07291000
EXIT;IF N≠0 THEN	07292000
BEGIN STREAM(I);	07293000
\$ SET OMIT = NOT(PACKETS)	07293099
DS:=20 LIT " NO PACKETS ON DISK+";	07293100
\$ POP OMIT	07293101
\$ SET OMIT = PACKETS	07293199
DS:=18 LIT " NO DECKS ON DISK+";	07293200
\$ POP OMIT	07293201
SPOUT(I);%	07294000
\$ SET OMIT = PACKETS	07294899
END ELSE FORGETSPACE(I);	07294900
\$ POP OMIT	07294901
\$ SET OMIT = NOT(PACKETS)	07294999
END ELSE% CHECK FOR COUNT REQUEST,	07295000
IF COUNT THEN%	07295010
BEGIN;STREAM(C+PKTCT,S+START,%;	07295020

```

F*FINAL,T1*0,T2*0,I)I;%
BEGIN DS=LIT " "; T2=D1I;%
SI*LOC C; DI*LOC T1I;%
DS*2 DEC; SI*LOC T1; DI*T2;
DS*2 CHR; T2=D1; DI=D1-2I;%
DS*FILL; DI*T2I;%
DS*7 LIT " PACKET";%
SI*LOC C; SI*SI+7I;%
IF SC="1" THEN% ONLY 1 DECK
BEGIN DS*2LIT " ";%
SI*LOC F; SI*SI+1I;%
DS*5 CHR;%
END ELSE% MORE THAN 1
BEGIN DS*3 LIT "S, ";%
SI*LOC S; SI*SI+1I;%
DS*5 CHR;%
DS*6 LIT " THRU ";%
SI*SI+4; DS*4 CHR;%
END;%
DS=LIT "+";%
END;%
SPOUT(I)I;%
END ELSE FORGETSPACE(I)I;%
$ POP OMIT
UNLOCKCONTROLDECKS;
END;%
PROCEDURE REMOVEDECK(N,U); VALUE N,U; REAL N,U;
BEGIN REAL I,T,A,L1,J=I,L2,V;%
$ SET OMIT = NOT(PACKETS)
REAL L3;
LABEL FAIL,CONTINUE;
$ POP OMIT
LABEL L,EXIT,REMOVE;%
LOCKCONTROLDECKS;
IF (I ← DIRECTORYSEARCH("DECK "N,5)) = 0 THEN%
$ SET OMIT = NOT(PACKETS)
FAIL;
$ POP OMIT
BEGIN I := SPACE(5);
STREAM(N,I)I;%
$ SET OMIT = NOT(PACKETS)
BEGIN DS:=5 LIT " PKT ";
$ POP OMIT
$ SET OMIT = PACKETS
BEGIN DS ← 6 LIT " DECK ";%
$ POP OMIT
SI ← LOC N; SI ← SI+1; DS ← 5 CHR;%
DS ← 13 LIT " NOT ON DISK";%
END;%
GO TO EXIT;%
END;%
$ SET OMIT = NOT(SHAREDISK)
IF (T:=M[I+4]),[2:1] THEN
IF T,[4:2]≠SYSNO THEN
BEGIN
STREAM(N,A+T,[4:2]+1,I←I INX 0);
BEGIN DS:=6 LIT" DECK ";SI:=LOC N;SI:=SI+1;
DS:=5 CHR;DS:=18 LIT" IN USE BY SYSTEM ";
SI:=SI+9;DS:=CHR;DI:=DI-1;SKIP DB;DS:=SET;
DS:=LIT"+";

```

```

07295030
07295040
07295050
07295060
07295070
07295080
07295090
07295100
07295110
07295120
07295130
07295140
07295150
07295160
07295170
07295180
07295190
07295200
07295210
07295220
07295230
07295235
07295240
07295241
07296000
07297000
07298000
07299000
07299499
07299500
07299600
07299601
07300000
07301000
07303000
07303499
07303500
07303501
07304000
07305000
07305099
07305100
07305101
07305999
07306000
07306001
07307000
07308000
07309000
07310000
07311000
07311199
07311200
07311210
07311220
07311230
07311240
07311250
07311260
07311270

```

END;	07311280
GO TO EXIT;	07311300
END;	07311310
\$ POP OMIT	07311311
\$ SET OMIT = NOT(PACKETS)	07311499
L3:=M[I+6],[FF];	07311500
\$ POP OMIT	07311501
L2:=M[I+6],[CF];	07312000
IF (A1=FIRSTDECK)=(L1:=1,[FF]) THEN	07313000
BEGIN	07314000
\$ SET OMIT = PACKETS	07314099
IF (FIRSTDECK:=L2)=0 THEN LASTDECK:=0;	07314100
\$ POP OMIT	07314101
\$ SET OMIT = NOT(PACKETS)	07314109
FIRSTDECK:=IF L3 NEQ 0 THEN L3 ELSE L2;	07314110
IF L2=0 THEN LASTDECK+IF L3 NEQ 0 THEN L3 ELSE 0;	07314120
\$ POP OMIT	07314121
DISKWAIT(KLUMP,3,DIRECTORYTOP+3);	07314200
\$ SET OMIT = NOT(PACKETS)	07314289
IF L3 NEQ 0 THEN GO TO CONTINUE ELSE	07314290
\$ POP OMIT	07314291
GO TO REMOVE;	07314300
END;	07314400
J + I,[33:15];%	07315000
L:	07316000
DISKWAIT(-J,30,A);	07317000
IF (V:=M[J+6],[CF])=0 THEN	07318000
\$ SET OMIT = NOT(PACKETS)	07318009
IF A=L1 THEN GO REMOVE ELSE BEGIN FORGETSPACE(I); GO FAIL	07318010
END;	07318012
\$ POP OMIT	07318013
\$ SET OMIT = PACKETS	07318019
GO TO REMOVE;	07318020
\$ POP OMIT	07318021
IF V ≠ L1 THEN%	07319000
BEGIN A ← V; GO TO L END;%	07320000
\$ SET OMIT = PACKETS	07320999
M[J+6],[CF]:=L2;	07321000
\$ POP OMIT	07321001
\$ SET OMIT = NOT(PACKETS)	07321099
M[J+6],[CF]+IF L3≠0 THEN L3 ELSE L2;	07321100
\$ POP OMIT	07321101
DISKWAIT(J,30,A);	07322000
IF L2 = 0 THEN%	07324000
BEGIN	07325000
\$ SET OMIT = PACKETS	07325999
LASTDECK:=A;	07326000
\$ POP OMIT	07326001
\$ SET OMIT = NOT(PACKETS)	07326099
LASTDECK:=IF L3 NEQ 0 THEN L3 ELSE A;	07326100
\$ POP OMIT	07326101
DISKWAIT(KLUMP,3,DIRECTORYTOP+3);	07327000
\$ SET OMIT = PACKETS	07327999
END;	07328000
\$ POP OMIT	07328001
\$ SET OMIT = NOT(PACKETS)	07329999
END ELSE IF L3=0 THEN ELSE	07330000
CONTINUE;	07330050
BEGIN J+I INX 0;	07330100
DISKWAIT(-J,30,L3);	07330200

```

M[J+6],[CF]+L2;                                07330300
DISKWAIT(J,30,L3);                              07330400
END;                                              07330500
$ POP OMIT                                       07330501
  REMOVE;                                        07331000
    FORGETSPACE(I);                             07332000
    I:=DIRECTORYSEARCH("DECK  ",N,8),[CF];      07333000
    T ← M[I+9];%                                07343000
    FOR V ← 1 STEP 1 UNTIL T DO%                07344000
      IF M[I+V+9]≠0 THEN FORGETUSERDISK(M[I+V+9],M[I+8]); 07345000
    STREAM(N,I);%                                07346000
$ SET OMIT = NOT(PACKETS)                       07346099
  BEGIN DS:=5 LIT " PKT ";                      07346100
$ POP OMIT                                       07346101
$ SET OMIT = PACKETS                            07346999
  BEGIN DS ← 6 LIT " DECK ";%                  07347000
$ POP OMIT                                       07347001
  SI ← LOC N; SI ← SI+1; DS ← 5 CHR;%          07348000
  DS ← 9 LIT " REMOVED";%                      07349000
  END;%                                         07350000
EXIT;                                           07351000
  SPOUTER(I,U,LIBMSG);                          07352000
  UNLOCKCONTROLDECKS;                          07353000
  END;%                                         07354000
PROCEDURE DECKREMOVER(B); VALUE B; REAL B;%    07355000
  BEGIN REAL K,N,F;%                             07355100
    INTEGER U; LABEL ON,ERR;                   07355200
    REAL D;                                     07356000
    LABEL L,TRYIT,GIVEUP;                     07357000
    K ← B,[15:15]=1;%                          07358000
  LI  STREAM(X+12,B:A+0);%                     07359000
    BEGIN SI ← B;%                             07360000
      U:  IF SC = " " THEN BEGIN SI←SI+1; GO TO U END;% 07360100
          IF SC="=" THEN BEGIN DI←LOC X; DI←DI+6; DS←CHR; 07360200
              SI←SI-1; B←SI; GO TO E END;%07361000
          IF SC = "#" THEN SI:=SI+1;           %033=07361500
      BL: IF SC=" " THEN BEGIN SI:=SI+1;GO TO BL; END; 07362000
          DI:=LOC X; DI:=DI+1; DS:=5 LIT "#0000"; %033=07363000
          4(IF SC < "0" THEN JUMP OUT TO EN; %033=07364000
          IF SC > "9" THEN JUMP OUT TO EN; %033=07365000
          SI:=SI+1; TALLY:=TALLY+1); %033=07365500
      EN: A:=TALLY; SI:=SI-A; DI:=DI-A; DS:=A CHR; %033=07366000
      NI  IF SC = " " THEN BEGIN SI←SI+1; GO TO N END;% 07367000
          DS ← CHR; B ← SI;%                   07368000
      EI END;%                                  07369000
  P(B,+,+,N,+);%                               07370000
  F←N,[36:6]; N,[36:6]←" ";                    07371000
  IF F="+" OR F="," OR F="=" THEN              07371100
    BEGIN IF F="=" THEN                        07371200
      BEGIN IF D=0 THEN D := SPACE(30);      07371300
          LOCKCONTROLDECKS;                  07371400
          IF (NI=FIRSTDECK)=0 THEN          07371450
            BEGIN FI=" ";                    07371500
                UNLOCKCONTROLDECKS;        07371600
                GO ON;                     07371700
            END;                             07371750
          DISKWAIT("D,30,N);                07371800
          IF M[D+4],[4:2]≠SYSNO THEN        07371809
            IF (NI=M[D+6],[CF])≠0 THEN GO TRYIT ELSE 07371820

```

```

                                GO GIVEUP;
$ POP OMIT
                                Ni=M[D+2];
                                UNLOCKCONTROLDECKS;
                                END;
                                FOR U=0 STEP 1 UNTIL 3 DO
                                IF CIDROW[U]#0 THEN
                                IF(CIDTABLE[U,2] EQV N)=NOT 0 THEN
                                IF LABELTABLE[U+32]#0
$ SET OMIT = NOT(PACKETS)
                                AND LABELTABLE[U+32]#0214 AND
                                PACKETACT[U]=0
$ POP OMIT
                                THEN
                                BEGIN
                                ENDOFDECK(U);
                                GO ON;
                                END ELSE GO TO ERR;
                                REMOVEDECK(N,0);
                                ON;
                                IF F#"" THEN GO TO L;
                                FORGETSPACE(K);%
                                END ELSE
                                SPOUT(K);
                                ERR;
                                IF D#0 THEN FORGETSPACE(D);
                                END;%
                                BOOLEAN PROCEDURE READFROMDISK(H,IB);%
                                VALUE H,IB; ARRAY H[*],IB[*];%
                                BEGIN%
                                % H[0] = ADDRESS OF BU+1 (B)%
                                % H[1] = ADDRESS OF B2+1%
                                % H[2] = DECK NAME%
                                % H[3] = RECORDCOUNT (N)%
                                % H[4] = NEXT CONTROL CARD (L)%
                                % H[5] = RECORDS USED IN THIS BLOCK * 10 (R)%
                                % H[7] = H[30] ARE FILE HEADER%
                                REAL A,B;%
                                DEFINE N=H[3]#,L=H[4]#,R=H[5]#;%
                                INTEGER I=A;%
                                B = H[0];%
                                IF R = 0 THEN%
                                IF (M[B=2] AND IOMASK) = 0 THEN%
                                SLEEP([M[B=2]],IOMASK);%
                                STREAM(B+B+R,IB);%
                                BEGIN SI = B; DS = 10 WDS END;%
                                M[IB INX NOT 0] = 10;
                                IF (READFROMDISK + N=L) THEN%
                                L = IB[9];%
                                IF (A + N + *P(DUP)+1) > H[7] THEN%
                                BEGIN READFROMDISK:=1;
                                STREAM(1B);
$ SET OMIT = NOT(PACKETS)
                                BEGIN
$ POP OMIT
                                DS:=7 LIT "CC END,";
$ SET OMIT = NOT(PACKETS)
                                DS=LIT" "; 8(DS+8LIT" "); END;
$ POP OMIT
                                END
                                ELSE BEGIN IF (R + *P(DUP)+10) = 30 THEN%
                                BEGIN IB = [M[B=2]];%

```

```

07371830
07371851
07371900
07371950
07372000
07372090
07372100
07372200
07372300
07372309
07372310
07372320
07372321
07372330
%030=07372400
07372500
07372600
07372700
07372800
07372900
07373000
07374000
07374100
07374200
07375000
07376000
07377000
07378000
07379000
07380000
07381000
07382000
07383000
07384000
07385000
07386000
07387000
07388000
07389000
07390000
07391000
07392000
07393000
07394000
07394500
07395000
07396000
07397000
07398000
07398100
07398109
07398110
07398111
07398200
07398209
07398210
07398211
07398300
07399000
07400000

```

```

R ← 0;
A ← A DIV 3+1;
I ← H[A DIV H[8]+10]+A MOD H[8];
DISKIO(18,1-B,30,1);%
H[0] ← H[1];%
H[1] ← B;%
07400400
07400500
07401000
07402000
07403000
07404000
07405000
07405100
07405110
07406000
07407000
07408000
07408100
07408199
07408200
07408201
07408500
07408600
07409000
07409100
07409200
07409300
07410000
07410010
07411000
07412000
07413000
07414000
07415000
07415100
07416000
07417000
07417099
07417100
07417200
07417300
07417400
07417500
07417600
07417700
07417800
07417900
07418100
07418200
07418300
07418500
07418600
07418700
07418800
07418900
07419000
07419100
07419200
07419300
07419400
07419500
07419600
07419700
07419701
07419800

END; END; END;%
BOOLEAN PROCEDURE PRINTORPUNCHWAIT(Q,PNCH);VALUE Q,PNCH;REAL Q,PNCH;
FORWARD;
PROCEDURE ENDOFDECK(R); VALUE R; REAL R;
BEGIN ARRAY H[*];%
REAL B,1;%
BOOLEAN TOG;
$ SET OMIT = NOT(PACKETS)
REAL DISKAD,PBREC,T;
$ POP OMIT
LABEL EXIT;
TOG ← R,[1:1]; R ← ABS(R);
IF (H[1=CIDROW[R]])=0 THEN GO TO EXIT;
LABELTABLE[R+32] ← @114;
MULTITABLE[R+32] ← RDCTABLE[R+32] ← 0;
UNITCODE[R+9] := 0;
IF NOT TOG THEN REMOVEDECK(H[2],R+32) ELSE
P(DIRECTORYSEARCH("DECK ",H[2],14),DEL);
FOR I ← 0 STEP 1 UNTIL 1 DO%
BEGIN B ← H[I];%
IF (M[B=2] AND IOMASK) = 0 THEN
SLEEP([M[B=2]],IOMASK);%
END;%
IF CIDROW[R]=0 THEN GO TO EXIT; % FIXES TIMING PROB,
IF H,[18:15] ≠ 0 THEN
FORGETSPACE(H,[18:15]-2);
$ SET OMIT = NOT(PACKETS)
IF PACKETPBD[R] GEQ 11 THEN
BEGIN
PBCOUNT := PBCOUNT+1;
I := 001 & CIDTABLE[R,6][6:6:24];
IF (PBREC := DIRECTORYSEARCH("PBD ",I,5))≠0 THEN
BEGIN
IF PACKETPAGE[R]>1 THEN
BEGIN
PBREC := PBREC,[CF];
DISKAD := M[PBREC+10]+2;
DISKWAIT(-PBREC,30,DISKAD);
IF (M[PBREC+12] EQV ("ABORTED"))≠NOT 0 THEN
STREAM(B:=PBREC+11);
BEGIN
DS:=8LIT":x0x0000"; DS:=8LIT"OPACKET ";
END;
DISKWAIT(PBREC,30,DISKAD);
END;
P(DIRECTORYSEARCH("PBD ",I,14),DEL);
IF AUTOPRINT THEN
P(PRINTORPUNCHWAIT(I,0),DEL);
FORGETSPACE(PBREC);
END;
END;
PSEUDO[R] :=
$ POP OMIT
CIDROW[R] := 0;

```



IF(RUNUMBER+RUNUMBER+1)>0 THEN	07420000
STARTADECK(IF TOG THEN -H[2] ELSE 0);	07420010
FORGETSPACE(H);	07420050
EXIT;	07420100
END;*	07421000
PROCEDURE STARTADECK(N); VALUE N; REAL N;	07422000
BEGIN LABEL EXIT,L,POSSIBLE,NEXT;*	07423000
REAL I,R,T,A,S;	07424000
ARRAY H[*];*	07425000
REAL SDED;	07425100
LABEL AGAIN,START;	07425500
IF N,[1:1] THEN BEGIN SDED+ABS(N); N+0 END;	07425700
LOCKCONTROLDECKS;	07426000
IF RUNUMBER LEQ 0 AND N=0 THEN GO TO EXIT;	07426100
AGAIN;	07427500
FOR R + 0 STEP 1 UNTIL 3 DO*	07428000
IF CIDROW[R] = 0 THEN GO TO POSSIBLE;*	07429000
STREAM(S+S+SPACE(4));	07429100
DS:=27 LIT" ALL PSEUDO-READERS IN USE";	07429200
SPOUT(S);	07429300
GO TO EXIT;*	07430000
POSSIBLE;*	07431000
IF (A:=FIRSTDECK)=0 THEN GO TO EXIT;	07432000
H + CIDROW[R] + [M[S+GETSPACE(94,64,1)+2]]&94[8:38:10];	07433000
H[2] + 0;	07434000
L: DISKWAIT(-S,30,A);	07435000
IF N#0 THEN	07436000
BEGIN	07436100
IF H[2],[12:24]#N THEN GO TO NEXT;	07436200
IF H[4],[2:1] THEN	07436300
BEGIN	07436400
STREAM(A:=H[2]);	07436500
\$ SET OMIT = NOT(SHAREDISK)	07436509
N+H[4],[4:2]+17,	07436510
\$ POP OMIT	07436511
S);	07436520
\$ SET OMIT = PACKETS	07436599
BEGIN SI:=A;SI:=SI+1;DS:=6 LIT" DECK ";	07436600
\$ POP OMIT	07436601
\$ SET OMIT = NOT(PACKETS)	07436609
BEGIN SI:=A;SI:=SI+1;DS:=5 LIT" PKT ";	07436610
\$ POP OMIT	07436611
DS:=5 CHR;DS:=7LIT" IN USE";	07436700
\$ SET OMIT = NOT(SHAREDISK)	07436799
DS:=11 LIT" BY SYSTEM ";SI:=LOC N;	07436800
SI+SI+7; DS+CHR;	07436900
\$ POP OMIT	07436901
DS:=LIT"-";	07437000
END;	07437100
SPOUT(S);	07437200
CIDROW[R]=0;	07437300
GO TO EXIT;	07437400
END;	07437500
END ELSE	07437600
IF H[4],[2:1] OR (SDED#0 AND H[2]=SDED)	07437800
\$ SET OMIT = NOT(SHAREDISK)	07437899
OR H[4],[4:2]#SYSNO AND NOT RNALL	07437900
OR (BATCHTOG AND H[4],[7:1])	07437910
\$ POP OMIT	07437911
THEN GO TO NEXT;	07438000

H[4]:=(*P(DUP))&2[2:46:2]&SYSNO[4:46:2];	07438100
DISKWAIT(S,30,A);	07438200
H[0] ← S+32;%	07441000
H[1] ← S+64;%	07442000
T ← [H[30]]; DISKIO(T,1=H[0],30,H[10]);%	07443000
IF H[7] LSS 3 THEN H[62]:=IOMASK ELSE	07444000
BEGIN T1=[H[62]]; IF H[8]=1 THEN	07445000
DISKIO(T,1=H[1],30,H[11]) ELSE	07445100
DISKIO(T,1=H[1],30,H[10]+1);	07445200
END;	07445300
T1=GETSPACE(13,64,5)+4;	07445400
M[T INX 10] := H[5];	07446000
\$ SET OMIT = NOT(PACKETS)	07446100
T,[24:6]←H[6],[2:6];	07446149
\$ POP OMIT	07446150
H[3] := H[4] := H[5] := H[6] := 0;	07446151
LABELTABLE[R+32]:= "@14; %LET IT BE MOVED	07446200
I:=READFROMDISK(H,[M[T]]&10[8:38:10]);	07447000
FREECARD(T&R[3:43:5]);	07448000
IF (RUNUMBER←RUNUMBER-1) LEQ 0 OR N#0 THEN GO TO EXIT;	07448500
GO TO AGAIN;	07449000
NEXT: IF (A:=H[6],[CF])#0 THEN GO TO L;	07450000
IF N#0 THEN	07450200
BEGIN	07451000
STREAM(N,S);	07452000
\$ SET OMIT = PACKETS	07452100
BEGIN SI:=LOC N; SI:=SI+4; DS:=7 LIT" DECK #";	07452200
\$ POP OMIT	07452299
\$ SET OMIT = NOT(PACKETS)	07452300
BEGIN SI←LOC N; SI←SI+4; DS←6 LIT" PKT #";	07452301
\$ POP OMIT	07452309
DS:=4 CHR; DS:=13 LIT" NOT ON DISK #";	07452310
END;	07452311
SPOUT(S);	07452400
END ELSE FORGETSPACE(S);	07452500
CIDROW[R] ← 0;%	07452600
EXIT: UNLOCKCONTROLDECKS;	07452700
END;%	07453000
PROCEDURE RUNTHEDECK(B); VALUE B; REAL B;%	07455000
BEGIN REAL I,J;%	07456000
STREAM(SI:=0;B,AI=[I]);	07457000
BEGIN SI ← B;	07458000
L1 IF SC = " " THEN	07461000
BEGIN SI ← SI+1; GO TO L END;	07461100
IF SC="# " THEN	07461200
BEGIN L1: SI:=SI+1; IF SC=" " THEN GO TO L1;	07461300
DS:=4 LIT"0000"; DS:=4 CHR; TALLY:=1; GO TO EX;	07461310
END;	07461320
DI ← A; DI ← DI+7; DS ← CHR;	07461330
EX: SI:=TALLY;	07461340
END;	07461400
J:=P;	07461450
R:=B.[15:15]-1;	07461500
IF J THEN	07461560
BEGIN	07461570
FORGETSPACE(B);	07461600
STARTADECK(I);	07461700
END ELSE	07461800
	07461900
	07462000

```

BEGIN
    IF I GTR 4 THEN I:=NABS(RUNUMBER) ELSE
    BEGIN
        RUNUMBER:=I;
        FOR J:=0 STEP 1 UNTIL 3 DO
            RUNUMBER:=RUNUMBER-(CIDROW[J]#0);
        END;
        STREAM(X1:=1-I,[1:1],X2:=RUNUMBER,[1:1],I:=ABS(I),B);
        BEGIN CI:=CI+X1; GO L1; DS:=10LIT" WILL USE ";GO L2;
            L1: CI:=CI+X2; GO L2; DS:=LIT"-"; L2:
            SI:=LOC I;DS:=2 DEC;
            DS ← 13 LIT " PSEUDO-RDRS-";
        END;
        SPOUT(B);
        IF RUNUMBER GTR 0 THEN STARTADECK(0);
    END;
END;
PROCEDURE EXTERNALEND(B); VALUE B; REAL B;
    BEGIN REAL U; LABEL EXIT;
        U ← UNITIN(TINU,B);
        B ← B,[15:15]=1;
        IF U≤35 AND U≥32 THEN
            IF LABELTABLE[U] ≥ 0 THEN
                $ SET OMIT = NOT(PACKETS)
                IF LABELTABLE[U] NEQ @214 AND PACKETACT[U=32]=0 THEN
                    $ POP OMIT
                    IF CIDROW[U=32] # 0 THEN
                        BEGIN
                            ENDOFDECK(U=32);
                            FORGETSPACE(B);
                            GO TO EXIT;
                        END;
                        SPOUT(B);
                    EXIT;END;
PROCEDURE CHANGE PRIORITY(BUFF,MIX); VALUE BUFF,MIX; REAL BUFF,MIX;
    BEGIN INTEGER PRIORITY; REAL B;
        ARRAY LINKR = NT1[*];
        $ SET OMIT = NOT(PACKETS)
        DEFINE UNITNO = PSEUDOMIX[MIX]#;
        $ POP OMIT
        BUFF ← ((B+BUFF),[15:15]=1)&M[P(DUP)=1][9:9];
        STREAM(PRIORITY;B);
        BEGIN SI←B;
            N: IF SC="+" THEN GO TO X;
                IF SC<"0" THEN BEGIN SI←SI+1; GO TO N; END; B←SI;
            K: IF SC≥"0" THEN IF SC≤"9" THEN
                    BEGIN TALLY←TALLY+1; SI←SI+1; GO TO K END;
                SI←B; DI←LOC PRIORITY; B←TALLY; DS←B OCT; GO TO Z;
            X: DI←LOC PRIORITY; SKIP DB; DS←11 SET;
            Z:
        END STREAM;
        IF (PRIORITY+P) ≥ 0 THEN
            IF [MEM[MIX,0]],[CF]≥FENCE THEN
                IF JAR[MIX,*]#0 THEN
                    IF PUTORTAKE(MIX,[PRYOR[MIX]],1,0) ≥ 0 THEN
                        BEGIN NT1←PUTORTAKE(MIX,[PRYOR[MIX]],0,PRIORITY);
                            B ← PUTORTAKE(MIX,[JAR[MIX,2]],1,0);
                            B,[CF] ← IF PRIORITY>32767 THEN 32767 ELSE PRIORITY;
                            NT1 ← PUTORTAKE(MIX,[JAR[MIX,2]],0,B);
                            STREAM(J←PUTORTAKE(MIX,[JAR[MIX,0]],1,0),

```

\*030-07479000

07462100  
07462200  
07462250  
07462300  
07462400  
07462500  
07462600  
07463000  
07464000  
07464100  
07465000  
07466000  
07467000  
07468100  
07469000  
07472500  
07473000  
07473100  
07474000  
07475000  
07476000  
07477000  
07478000  
07478099  
07478100  
07478101  
07478500  
07479100  
07480000  
07481000  
07482000  
07483000  
07484000  
07485000  
07486000  
07486100  
07486499  
07486500  
07486501  
07487000  
07488000  
07489000  
07490000  
07491000  
07492000  
07493000  
07494000  
07495000  
07496000  
07497000  
07498000  
07500000  
07501000  
07502000  
07503000  
07503100  
07503200  
07503500  
07504000

```

I←PUTORTAKE(MIX,[JAR[MIX,1]],1,0),
MIX,PRIORITY,BUFF);
BEGIN DS←10 LIT " PRIORITY=";
SI←LOC PRIORITY; BUFF←DI; DS←6 DEC; DI←DI-6;
DS←5 FILL; DI←BUFF; DI←DI+6; DS←LIT"!";
SI←LOC J; 2(SI←SI+1; DS←7CHR; DS←LIT"/"); DI←DI-1;
DS←LIT"="; SI←LOC MIX; DS←2DEC; DS←LIT"+";
DI←DI-3; DS←FILL;
END END;
SPOUTER(BUFF,UNITNO,1);
IF STASUS[MIX] = RUNNING THEN
IF MIX=P2MIX THEN PRYOR[P2MIX] ← PRIORITY ELSE
BEGIN LINKR ← BED1;
DO IF NOT (B ← LINKR[2]=MIX) THEN LINKR ← LINKR[1]
UNTIL B;
LINKR[0] ← (*P(DUP)) & (PRIORITY+64)[CTF];
END;
END CHANGE PRIORITY;
PROCEDURE ENTERCONTROLDECK(H); VALUE H; ARRAY H[*];
BEGIN REAL R,S,T,T1,T2;
INTEGER I;
$ SET OMIT = NOT(PACKETS)
LABEL MORE;
$ POP OMIT
*
T←"DECK "&H[4][1:47:1]; % FOR SCRATCHDIR DELETE
S:=NEXTCDNUM(0);
DISKWAIT(KLUMP,3,DIRECTORYTOP+3); % CHANGE LASTCDNUM ON DISK
$ SET OMIT = NOT(PACKETS)
MORE;
$ POP OMIT
H[0]:=001200036000301;
$ SET OMIT = NOT(PACKETS)
T2←H[6],[FF]; H[6],[FF]←T1;
$ POP OMIT
STREAM(DATE,B←[H[3]]);
BEGIN SI←LOC DATE;DS←8 OCT;DI←DI-8;DS←2 LIT"+7";END;
H[4] := 0&
$ SET OMIT = NOT SHAREDISK
SYSNO[4:46:2]&
$ POP OMIT
15[12:44:4];
H[1]←(XCLOCK+P(RTR))&H[3][6:30:18];
H[2]←S:=@14&@12[6:42:6]&S[12:24:24]&@37[36:42:6];
T1:=EUF(T,S,H,[CF]=1);
$ SET OMIT = NOT(PACKETS)
IF T2 NEQ 0 THEN
BEGIN DISKWAIT(-(H INX 0), 30, T2);
FORGETESPDISK(T2);
S←H[2]; GO TO MORE;
END;
$ POP OMIT
H[2]←LASTCDNUM;
IF FIRSTDECK=0 THEN FIRSTDECK:=T1 ELSE
BEGIN
$ SET OMIT = SHAREDISK
LOCKDIRECTORY;
$ POP OMIT
DISKWAIT(-(I:=SPACE(30)),-30, LASTDECK);
M[I+6],[CF]:=T1;

```

```

07504100
07504200
07505000
07506000
07507000
07508000
07509000
07509500
07510000
07511000
07511100
07511200
07511300
07511400
07511500
07511600
07511700
07512000
07541000
07542000
07543000
07543099
07543100
07543101
07544000
07545000
07547000
07547100
07547499
07547500
07547501
07548000
07548099
07548100
07548101
07549000
07549100
07550000
07550003
07550005
07550007
07550010
07550100
07557000
07559000
07559099
07559100
07559110
07559120
07559180
07559190
07559191
07559500
07560000
07561000
07561990
07562000
07562010
07564000
07565000

```

DISKWAIT(I,-30, LASTDECK);	07566000
FORGETSPACE(I);	07567000
\$ SET OMIT = SHAREDISK	07567990
UNLOCKDIRECTORY;	07568000
\$ POP OMIT	07568010
END;	07569000
LASTDECK:=T1;	07570000
DISKWAIT(KLUMP,-3, DIRECTORYTOP+3);	07571000
UNLOCKTOG(CDMASK);	07572000
IF RUNUMBER GTR 0 THEN STARTADECK(0);	07573000
END ENTERCONTROLDECK;	07575000
BOOLEAN PROCEDURE MTXIN(I,U,BUFF);%	08000000
REAL U,BUFF; INTEGER I;%	08001000
BEGIN LABEL EXIT,X;%	08002000
U ← UNITIN(TINU,BUFF);	08003000
BUFF ← BUFF,[15:15]=1;	08004000
IF U > 15 THEN%	08005000
BEGIN;STREAM(BUFF); DS ← 8 LIT "INV KBD ";%	08006000
GO TO EXIT;%	08007000
END ELSE I ← TWO(U);	08008000
STREAM(A←TINU[U];BUFF);%	08009000
BEGIN SI←LOC A; SI←SI+5; DS←LIT " "; DS←3 CHR;%	08010000
DS ← LIT " "; A ← DI;%	08011000
END;%	08012000
P([BUFF],+);%	08013000
IF LABELTABLE[U] = @114 OR LABELTABLE[U] = @214 THEN%	08014000
BEGIN	08015000
STREAM(SAV:=((I AND SAVEWORD) NEQ 0), BUFF);	08015100
BEGIN	08015200
DS:=10LIT"NOT READY←";	08015300
SAV(DI:=DI-1; DS:=8LIT"(SAVED)←");	08015400
END;	08015500
GO TO EXIT;	08016000
END;%	08017000
IF LABELTABLE[U] < 0 THEN%	08018000
BEGIN;STREAM(BUFF); DS ← 7 LIT "IN USE←";%	08019000
END%	08020000
ELSE GO TO X;%	08021000
EXIT;MTXIN ← TRUE;%	08022000
XIEND;%	08023000
PROCEDURE TAPEPURGE(BUFF); VALUE BUFF; REAL BUFF;%	08024000
BEGIN LABEL EXIT;%	08025000
REAL I,U;%	08026000
REAL R,T;	08027000
BOOLEAN TEST;	08027100
REAL WHAT = BUFF;%	08028000
IF MTXIN(I,U,WHAT) THEN GO TO EXIT;%	08029000
STREAM(B:=BUFF,T+[T]);	08029015
BEGIN SI:=B; SI:=SI+6;	08029020
IF SC="*" THEN	08029025
BEGIN SI:=SI+1;	08029030
5(IF SC="*" THEN JUMP OUT;	08029035
TALLY:=TALLY+1;SI:=SI+1);	08029040
B:=TALLY; SI:=SI-B; DS:=B OCT;	08029045
END;	08029050
END;	08029055
LABELTABLE[U] ← -@14;	08029100
IF (R←WAITIO(@5000000000,@177,U))≠0 THEN	08030000
IF R≠@120 THEN %ERROR OTHER THAN WRITE LOCK	08030100
BEGIN;STREAM(U←TINU[U],BUFF);	08030200

BEGIN DS←14 LIT "#CANNOT PURGE ";	08030300
SI←LOC U; SI←SI+5; DS←3CHR;	08030310
DS←LIT"+";	08030320
END;	08030330
LABELTABLE[U]←@214;	08030400
GO TO EXIT;	08030500
END ELSE %NO WRITE RING	08030600
BEGIN; STREAM(BUFF); DS ← 11 LIT "WRITE LOCK+";%	08031000
LABELTABLE[U] ← @114;	08031100
GO TO EXIT;%	08032000
END;%	08033000
IF T=0 THEN	08033980
BEGIN T:=PRNTABLE[U],[30:18]; TEST:= TRUE END;	08033990
STREAM(A:=T,BUFF);	08034000
BEGIN DI ← DI + 3; DS ← 8 LIT " LABEL ";	08035000
8(DS←2 LIT "OX");	08035100
24(DS←2 LIT "0"); DS←2 LIT "≥+";	08036000
DI ← DI-21; SI ← LOC A; DS ← 5 DEC;%	08037000
END;%	08038000
RRRMECH ← I OR RRRMECH;%	08039000
	08040000
MULTITABLE[U] ← 0;%	08041000
P(WAITIO(@4200000000,0,U),DEL);%	08042000
R ← WAITIO(BUFF INX @120500000001,@2000000,U) OR%	08043000
WAITIO(BUFF INX 10,@2000000,U);%	08044000
IF MOD3IOS THEN	%A108044500
DO UNTIL P(WAITIO(BUFF INX @340000012,@50,U))=e10	%A108044600
ELSE	%A108044700
P(WAITIO(@4200000000,0,U),DEL);%	08045000
SLEEP([TOGGLE],STATUSMASK);	08046000
RRRMECH ← RRRMECH AND NOT I;%	08047000
READY ← READY AND NOT I;%	08048000
IF R = 0 THEN BEGIN%	08049000
LABELTABLE[U] ← @114;%	08050000
IF TEST THEN BEGIN STREAM(BUFF);DS:=6LIT"PG=ED+";END	08051000
ELSE BEGIN STREAM(A:=T,B:=PRNTABLE[U],[30:18],BUFF);	08051010
BEGIN DS:=10LIT"PG=ED(PRN=";	08051020
SI:=LOC A; DS:=5 DEC;	08051030
DS:=5LIT".WAS ";	08051040
SI:=LOC B; DS:=5 DEC;DS:=2LIT"+";	08051050
END;	08051060
END;	08051070
EXIT;SPOUT(BUFF);	08052000
END ELSE BEGIN%	08053000
LABELTABLE[U] ← @214;%	08054000
FORGETSPACE(BUFF);%	08055000
END;%	08056000
END;%	08057000
PROCEDURE MIXPRINT(BUFF); VALUE BUFF; REAL BUFF;%	08058000
BEGIN REAL C,I,T;	08059000
FOR I ← 1 STEP 1 UNTIL MIXMAX DO%	08060000
IF JAR[I,*] ≠ 0 THEN%	08061000
BEGIN	08062000
TABCNT[I]:=TABCNT[I]+1;	08062025
IF T THEN BUFF:=BUFF&SPACE(5)[CTC];	08062050
IF (C:=PUTORTAKE(I,[PRYOR[I]],1&1[2:47:1],0)) ≠	08063000
NOT 0 THEN STREAM(C,	08063050
J:=PUTORTAKE(I,[JAR[I,0]],1,0),	08063100
K:=PUTORTAKE(I,[JAR[I,1]],1,0),	08063200
U:=PUTORTAKE(I,[USERCODE[I]],1,0),	08063300



```

IF INFOTYPE="DATE " THEN %RH 08099200
BEGIN %RH 08099210
    INFOTEMP:=HDR[3],[30:18]&1[1:43:5]; %RH 08099220
    GO GETADATE; %RH 08099230
END ELSE %RH 08099240
IF INFOTYPE="LAST " THEN %RH 08099300
BEGIN %RH 08099310
    INFOTEMP:=HDR[3],[12:18]&2[1:43:5]; %RH 08099320
    GO GETADATE; %RH 08099330
END ELSE %RH 08099340
IF INFOTYPE="RECS " THEN %RH 08099400
BEGIN %RH 08099410
IF (A EQV "PBD ") = NOT 0 OR %RH 08099412
(A EQV "PUD ") = NOT 0 THEN %RH 08099414
    INFO := (HDR[7]*5)&3[1:43:5] ELSE %RH 08099416
    INFO:=(HDR[7]+1)&3[1:43:5]; %RH 08099420
    GO TO INFOUT; %RH 08099430
END ELSE %RH 08099440
IF INFOTYPE="SIZE " THEN %RH 08099500
BEGIN PG:=HDR[9] AND @37; %RH 08099510
    INFO:=0; WHILE (INFO:=INFO+1) LEQ PG DO IF %RH 08099520
    HDR[INFO+9]#0 THEN INFOTEMP:=INFOTEMP+1; %RH 08099530
    INFO:=(INFOTEMP*HDR[8])&4[1:43:5]; INFOTEMP:=0; %RH 08099540
    GO TO INFOUT; %RH 08099550
END; %RH 08099560
GO TO INFOUT; %RH 08099570
GETADATE; %RH 08099700
    STREAM(A:=INFOTEMP,[30:18],I:=[INFO]); %RH 08099710
    BEGIN SI:=LOC A;DS:=8 DEC END; %RH 08099720
    GIMEDATE([INFO],[CF],-INFO); %RH 08099730
    INFO,[1:5]:=INFOTEMP,[1:5]; %RH 08099740
INFOUT; %RH 08099750
    P(DEL); %RH 08099760
    GO TO IN; %RH 08099770
END; %RH 08099800
CODE := BUFF,[9:16]; %RDS%08099900
BUFF←(C←BUFF),[15:15]-1; %RH 08100000
NB:=IM[SPACE(60)]&60[8:38:10]; %RH 08101300
NAMEID(A,C); %RH 08102000
NAMEID(B,C); IF B="/" THEN NAMEID(B,C); %RH 08102100
NAMEID(INFOTYPE,C); %RH 08102200
CI:=0; %RH 08103000
IF ((A EQV B)=NOT 0) AND ((A EQV "← ")=NOT 0) THEN GO XOUT; %RH 08103100
IF (A EQV "←← ")=NOT 0 THEN A←"← "; %RH 08103200
IF (B EQV "←← ")=NOT 0 THEN B←"← "; %RH 08103300
J1:=J3:=0; K1:=K2:=MODULUS-1; %10408103400
IF CODE#3 THEN %RH 08103450
BEGIN %RH 08103460
    IF A,[6:6] NEQ LFT THEN J1:=K1:=(A,[6:18]+A,[24:24]) MOD MODULUS; %RH 08103500
    IF B,[6:6] NEQ LFT THEN J3:=K2:=(B,[6:18]+B,[24:24]) MOD MODULUS; %RH 08103600
END; %RH 08103650
IF ((INFOTYPE EQV "← ")# NOT 0 AND CODE=0) THEN CODE:=5; %RH 08103700
IF CODE#0 THEN HDR:=IOQUE&SPACE(30)[CTC]; %RH 08103900
FOR J1:=J1 STEP 1 UNTIL K1 DO %10408104000
FOR J2:=J3 STEP 1 UNTIL K2 DO %10408104100
BEGIN J:=SCRAMBLE(J1,J2); %10408104200
DO BEGIN %10408104300
    BEGIN DISKWAIT(=NB,[CF],60,J); %10408105000
    FOR I:=0 STEP 3 UNTIL 57 DO %10408107000
    IF (T:=NB[I]) NEQ @14 THEN %10408109000

```



```

IF CODE=3 THEN COMMENT UD; GO GETHDR ELSE                                08110000
IF (T EQV A)≠ NOT 0 OR (A EQV "←")≠ NOT 0 THEN                          08111000
IF (NB[I+1]EQV B)≠ NOT 0 OR (B EQV "←")≠ NOT 0 THEN                      08112000
GETHDR: IF CODE=0 AND USERID=0 THEN GO TO IN ELSE                          08113000
        BEGIN T:=HDR,[CF]&NB[I+2][CTF]; %10408114000
          DISKWAIT(-T,[CF],30,T,[FF]); 08115000
          IF CODE=5 THEN FINDWHATINFO; %RH 08115060
          IF CODE=0 THEN IF SECURITYCHECK(NB[I],NB[I+1],USERID,T) 08115100
          ≠0 THEN GO TO IN ELSE GO TO AROUND; 08115200
          IF CODE=1 THEN COMMENT EX; 08116000
          BEGIN; 08116700
            STREAM(A+CALCULATEPURGE(-HDR[3],[2:10]),B+[PG]);08116800
            BEGIN SI+LOC A;DS+8 OCT END; 08116900
            IF PG>HDR[3],[12:18] 08117000
              THEN GO IN ELSE GO AROUND; 08117100
          END; 08117105
          IF ((NB[I]EQV "DECK")≠ NOT 0) AND 08117110
            (((NB[I+1]AND @77000000007777) EQV 08117120
              @12000000003714)≠ NOT 0) OR 08117130
            FALSE THEN GO AROUND; % 08117140
          IF CODE=4 THEN COMMENT SB; 08117200
          IF HDR[5]=12 THEN GO AROUND; N+HDR[6]; 08117300
          IF CODE≥2 THEN COMMENT UD,CR,SB; 08117400
            IF (NOT(CR:=HDR[IF CODE=4 THEN 5 ELSE 2],[6:42])) 08117500
              = NOT 0 THEN IF CODE≠2 THEN GO AROUND; 08117600
            IF CODE=2 THEN IF (PBDTOG+(P(NB[I],DUP) EQV 08117610
              "PBD")≠ NOT 0 OR (P(XCH) EQV "PUD") 08117620
              = NOT 0) THEN 08117630
          $ SET OMIT = PACKETS 08117631
            IF NB[I+1],[30:18] NEQ "001" THEN GO AROUND ELSE 08117632
          $ POP OMIT 08117633
          $ SET OMIT = NOT(PACKETS) 08117634
            IF NB[I+1],[42:6] NEQ "1" THEN GO AROUND ELSE 08117635
          $ POP OMIT 08117636
          BEGIN 08117640
            IF LABELREC=0 THEN LABELREC+SPACE(30); 08117650
            DISKWAIT(-LABELREC,30,HDR[10]+2); 08117660
            INFO,[1:5]+0; GO IN; 08117670
            END ELSE IF NOT CR=NOT 0 THEN GO AROUND; 08117680
          IF CODE=3 THEN COMMENT UD; 08117700
            IF (CR EQV A)≠ NOT 0 AND (A EQV "←")≠ NOT 0 08117800
              THEN GO TO AROUND; 08117900
          IN: BEGIN C+SPACE(12); 08118000
            STREAM(F:=[D[I]],CR,N,X:=(CODE≥2 AND CODE<5), 08119000
              Y:=CODE=4,INF1:=CODE=5,INFO,INFNO:= %RH 08119100
              INFO,[1:5],PBDTOG,LR+LABELREC INX 12,C); 08119200
            BEGIN SI:=F; DS:=LIT" "; SI:=SI+1; DS:=7 CHR; 08120000
              SI:=SI+1; DS:=LIT"/"; DS:=7 CHR; 08121000
              Y(DS:=8 LIT" SECURED"); 08122000
              X(DS:=4 LIT" BY "; SI:=LOC CR; SI:=SI+1; 08123000
                DS:=7 CHR); 08124000
              Y(DS:=LIT"/"; SI:=SI+1; DS:=7 CHR); 08125000
              PBDTOG(DS:=4 LIT" IS"; SI:=LR; 08125010
                2(SI:=SI+1; DS:=7 CHR; DS:=LIT"/"); 08125020
                DI:=DI-1; DS:=4 LIT" OF "; 08125030
                2(SI:=SI+1; DS:=7 CHR; DS:=LIT"/"); 08125040
                DI:=DI-1); 08125050
              INF(CI:=CI+INFNO;GO ERR;GO INF1;GO INF2; 08125100
                GO INF3;GO INF4; %RH 08125110
                INF1; %RH 08125120

```

```

DS:=10LIT" CREATED: "; %RH 08125130
DATE: SII=LOC INFO; %RH 08125140
SI:=SI+2;DS:=2 CHR;2(DS:=LIT"/"); %RH 08125150
DS:=2 CHR);GO INFEND; %RH 08125160
INF2; %RH 08125200
DS:=11LIT" ACCESSED: ";GO DATE; %RH 08125210
INF3; %RH 08125300
DS:=10LIT" RECORDS: "; %RH 08125310
DECM; %RH 08125320
SI:=LOC INFO;DS:=6 DEC;X:=DI; %RH 08125330
DI:=DI-6;DS:=5 FILL;DI:=X; %RH 08125340
GO INFEND; %RH 08125350
INF4; %RH 08125400
DS:=11LIT" SEGMENTS: ";GO DECM; %RH 08125410
ERR: INFEND;); %RH 08125420
DS:=LIT"+"; 08126000
END; 08127000
SPOUT(C); 08128000
AROUND: END; END; END; 08129000
END UNTIL (J:=NB[2],[FF])=0; %10408129100
END; %10408129200
OUT: IF CODE#0 THEN FORGETSPACE(HDR); 08130000
XOUT: FORGETSPACE(NB); 08130100
IF LABELREC#0 THEN FORGETSPACE(LABELREC); 08130200
IF C = 0 THEN 08131000
BEGIN M[BUFF]:=FLAG(NABS("NULL ")); 08132000
SPOUT(BUFF); 08133000
END ELSE FORGETSPACE(BUFF); 08134000
END PRINTDIRECTORY; 08135000
PROCEDURE PBIO(A,P); VALUE A; REAL A,P; FORWARD; %P 08170100
PROCEDURE CONTINUITYBIT;% 08171000
BEGIN REAL T,IOD,LINK,U;% 08172000
ARRAY A[*]; 08172500
REAL RCW=+0;% 08173000
ARRAY R=-4[*]; DEFINE FIB=A#; %P 08173100
CHECKSTACKSPACE;% %WF 08173200
U=(LINK + NFLAG(M[(IOD + NFLAG(MET+PRT[P1MIX,9])) INX% 08174000
P(0,LNG,XCH)) INX NOT 0)),[12:6]);% 08175000
IF U ≥ 32 THEN 08175100
BEGIN A + M[T]; 08175200
IF READFROMDISK(CIDROW[U-32],A) THEN 08175300
M[T] + A&1[27:47:1]&0[2:47:1] ELSE 08175400
M[T] + R; GO TO RETURN; 08175500
END; 08175600
M[IOD INX NOT 1]+FLAG(LINK); FIB+M[T-3]; %P 08176000
M[FIB[14]INX 17]+[M[FIB[5],[FF]]]&IOD[3:3:30]&0[20:20:1]; 08177000
;FIB[5]+P(DUP,LOD,0,1,CFX,ADD); %P 08177100
IF FIB[14],[FF]≤FIB[14],[CF] THEN %% BUFFER FULL %P 08177200
PBIO(T,FIB[14]) %P 08178000
ELSE %P 08179000
BEGIN; STREAM(A+FIB[14],[CF], B+FIB[14],[FF]); %P 08179600
BEGIN SI+A; DS+18 WDS END; %P 08179700
FIB[14],[FF]+FIB[14],[FF]=18; %P 08179800
END; %P 08179900
GO RETURN %P 08180000
END CONTINUITYBIT; %P 08181000
BOOLEAN PROCEDURE PRINTORPUNCHWAIT(Q,PNCH);VALUE Q,PNCH;REAL Q,PNCH; 08255000
% 08255050
% THIS PROCEDURE IS RESPONSIBLE FOR STARTING PRNPBT/DISK. IT CHECKS 08255055
% FOR I/O UNITS AS REQUIRED AND, IF AVAILABLE, GRABS THEM, THE 08255060

```

```

% PARAMETERS ARE:
% Q ≤-16 LOGICAL UNIT NUMBER FOR OUTPUT, TAPES AND DISK ARE 08255065
% SEARCHED TO FIND A FILE TO PRINT, THIS IS USED ONLY 08255070
% WHEN AUTOPRINT IS SET OR FOR RJE, 08255075
% >=16, ≤0 LOGICAL UNIT NUMBER OF A BACK-UP TAPE, CHECK FOR AN 08255080
% AVAILABLE OUTPUT UNIT, 08255085
% >0 FID OF A DISK FILE, CHECK FOR OUTPUT UNIT, 08255090
% PNCH,[47:1] ON FOR PUNCH BACK-UP, 08255095
% [39:8] NUMBER OF COPIES FROM PB MESSAGE, 08255100
% [31:8] IF TAPE, NUMBER OF FILE TO PRINT (FROM PB), 08255105
% IF DISK, =0 IF ENTIRE PACKET SHOULD BE PRINTED, =1 IF 08255110
% NOT, 08255115
% [30:1] ON IF =0 WAS USED IN PB MSG, 08255120
% [2:1] ON IF CALLED FROM PRINTBACKUP, I.E. A PB MESSAGE, 08255122
% [1:1] ON IF CALLED FROM PRNPBT/DISK, 08255130
% 08255135
% 08255140
% 08255140
BEGIN INTEGER U,V,I,J,J1,J2,S; 08255200
REAL A,HDR,SEGO=S,F=J; 08255400
REAL PBT,PUD,PBD; 08255500
ARRAY D[*],SHEAT=D[*]; 08255600
LABEL TRYAGAIN,PRNPBT,DISK; 08255700
LABEL PBTAPE,FOUND,FIREITUP,QUIT; 08255800
DEFINE MFID = (IF V=22 THEN PUD ELSE PBD)*; 08255900
$ SET OMIT = SHAREDISK 08256190
DEFINE SIXTY = 60*; 08256200
$ SET OMIT = NOT SHAREDISK 08256210
INTEGER SIXTY; 08256220
$ POP OMIT OMIT 08256230
PBT := "PBT "; PUD := "PUD "; PBD := "PBD "; 08256500
IF Q≥(-15) THEN %%% PB GIVEN: LOOK FOR LP, 08257000
BEGIN IF PNCH THEN IF LABELTABLE[V+22]≠0 THEN V←0 ELSE ELSE 08257500
IF LABELTABLE[V+20]≠0 THEN 08257600
IF LABELTABLE[V+21]≠0 THEN V←0; 08258000
IF V≠0 THEN % WE HAVE AN OUTPUT UNIT 08258200
IF Q>0 THEN % BACK-UP DISK 08258400
BEGIN U:=18; 08258600
IF AUTOPRINT THEN % CHECK IF A PRNPBT WAS STARTED, 08258800
% IF SO, START THIS ONLY FOR PB, 08258990
IF (A:=DIRECTORYSEARCH(MFID,Q,19))=0 08259225
THEN IF PNCH,[2:1] THEN ELSE GO QUIT 08259250
ELSE BEGIN IF M[A+4],[6:1] 08259275
THEN 08259300
$ SET OMIT = NOT SHAREDISK 08259325
UNLOCK(A,[FF]) 08259350
$ POP OMIT 08259375
ELSE BEGIN 08259400
M[A+4],[6:1] := 1; 08259425
DISKWAIT(A,[CF],-30,A,[FF]); 08259450
END; 08259500
FORGETSPACE(A); 08259525
$ SET OMIT = SHAREDISK 08259550
UNLOCKDIRECTORY; 08259575
$ POP OMIT 08259600
END; 08259625
END 08259800
ELSE GO TO PBTAPE; 08260000
END ELSE 08260250
BEGIN V:=ABS(Q); % LP (OR PUNCH) GIVEN, LOOK FOR FILE, 08260500
BEGIN IF V=22 THEN % CHECK FOR TAPE 08261000
BEGIN A:="PUTMCP "; 08261250

```

	PNCH:=PNCH OR 1;	08261500
	END ELSE	08261750
	A:="PBTMCP ";	08262000
	FOR Q:=0 STEP 1 UNTIL 15 DO	08262250
	IF (MULTITABLE[Q] EQV A)=NOT 0 THEN	08262500
	IF (LABELTABLE[Q] EQV @122212342546447)=NOT 0 THEN	08262750
	IF RDCTABLE[Q],[14:10]=1 THEN	08263000
	BEGIN RRRMECH:=TWO(Q) OR RRRMECH;	08263250
PBTAPE:	LABELTABLE[U:=ABS(Q)] :=	08263500
	PBT&TINU[V][6:30:18]&@21[1:43:5];	08263750
	MULTITABLE[V] := PBT;	08264000
	LABELTABLE[V] :=	08264500
	PBT&TINU[U][6:30:18]&@21[1:43:5];	08265000
	GO FIREITUP;	08265500
	END;	08266000
	END SEARCHING FOR TAPE;	08266500
	IF PBCOUNT#0 THEN % TRY FOR DISK	08267000
	BEGIN D:=[M[SPACE(90)]]&90[8:38:10];	08267500
\$ SET OMIT = SHAREDISK	LOCKDIRECTORY;	08267990
\$ POP OMIT		08268000
	A:=MFID;	08268010
	J1:=(A,[6:18] + A,[24:24]) MOD MODULUS;	08268500
	FOR J2:=0 STEP 1 UNTIL (MODULUS-1) DO	08268600
	BEGIN	08268700
\$ SET OMIT = NOT SHAREDISK	SIXTY:=60;	08268750
	S:=	08268790
\$ POP OMIT		08268800
	J:=SCRAMBLE(J1,J2);	08268850
	DO BEGIN DISKWAIT(=(D INX 30),SIXTY,J);	08268860
	FOR I:=30 STEP 3 UNTIL 87 DO	08268900
	IF (D[I] EQV A) = NOT 0 THEN	08268950
	IF D[I+1],[CF]=1 THEN	08269000
	BEGIN DISKWAIT(=D,[CF],-30,D[I+2],[CF]);	08269100
	IF D[4],[1:3] # 0 OR D[4],[6:1]	08269200
\$ SET OMIT = NOT(PACKETS)	OR LABELTABLE[IF V=20 THEN 21 ELSE	08269300
	IF V=21 THEN 20 ELSE 22],[6:24]	08269400
	=D[I+1],[6:24]	08269509
\$ POP OMIT		08269510
	OR (D[4],[16:20] OR D[9],[1:28])#0	08269520
\$ SET OMIT = NOT SHAREDISK	THEN	08269530
\$ POP OMIT		08269531
	UNLOCK(D[I+2],[CF])	08269600
		08269650
	ELSE	08269690
	BEGIN D[4],[6:1]I=1;	08269700
	PBCOUNT:=PBCOUNT+1;	08269710
	DISKWAIT(D,[CF],-30,D[I+2],[CF]);	08269750
\$ SET OMIT = NOT SHAREDISK		08269800
\$ POP OMIT		08269900
	UNLOCK(S);	08270000
		08270040
	UI=18;	08270050
	Q:=D[I+1];	08270060
	GO FOUND;	08270100
		08270200
		08270300
\$ SET OMIT = NOT SHAREDISK	END END;	08270350
\$ POP OMIT	SIXTY:=60;	08270390
		08270400
		08270410

```

                                END UNTIL (J:=D[32],[FF])=0;
$ SET OMIT = NOT SHAREDISK
                                UNLOCK(S);
$ POP OMIT
                                END;
                                FOUND;
                                FORGETSPACE(D);
$ SET OMIT = SHAREDISK
                                UNLOCK(DIRECTORY);
$ POP OMIT
                                END SEARCHING FOR DISK;
                                END;
%                                IF WE HAVE BOTH AN INPUT FILE AND AN OUTPUT UNIT,
%                                FIRE UP PRNPBT/DISK.
                                IF U#0 AND V#0 THEN
                                BEGIN
                                BEGIN LABELTABLE[V]:=Q&@21[1:43:5];
                                MULTITABLE[V]:=PBD;
                                END;
                                FIREITUP:
                                A:=V&U[38:43:5]&PNCH[21:30:17];
                                IF PNCH,[1:1] THEN P(A) ELSE
                                BEGIN
                                TRYAGAIN:
                                IF (HDR:=DIRECTORYSEARCH(P(PRNPBT),P(DISK),3)) # 0 THEN
                                BEGIN
                                IF P1MIX NEQ 0 THEN
                                BEGIN
                                F:=GETSPACE(30,64,0)+2;
                                MOVE(30, HDR INX 0, F);
                                FORGETSPACE(HDR);
                                HDR,[CF]:=F;
                                END;
                                SHEAT := [M[F:=GETSPACE(31,64,0)+2]] & 30[8:38:10];
                                MOVE(30,F-1,F);
                                SEGO := GETSPACE(30,64,0)+2;
                                DISKWAIT(-SEGO, 30, M[HDR INX 10]);
                                F,[FF] := HDR; % CORE ADDRESS OF HEADER
                                SHEAT[7] := SEGO; % CORE ADDRESS OF SEGMENT ZERO
                                SHEAT[0] := SHEAT[14] := P(PRNPBT);
                                SHEAT[1] := P(DISK);
                                SHEAT[2] := 0 & 5[8:38:10] & % PRIORITY#0,RUN CODE
                                (PNCH,[2:1]=0)[4:47:1]; % SET NOT "PB"
                                SHEAT[16] := SHEAT[17] := @377777777777; % TIME LIMITS
                                SHEAT[19] := A; % COMMON VALUE
                                SHEAT[20] := 4; % CORE ESTIMATE
                                SHEAT[21] := 150; % STACK SIZE

                                STREAM(A:=0 : S := P(,SCHEDULEIDS));
                                BEGIN
                                SI:=S;
                                47(SKIP SB; SKIP DB; TALLY:=TALLY+1;
                                IF SB THEN ELSE JUMP OUT);
                                DS:=SET; A:=TALLY;
                                END STREAM STATEMENT;

                                I := P;
                                SHEAT[3],[8:10] := I; % SCHEDULE NUMBER
                                SHEAT[23] := (CLOCK + P(RTR)) DIV 60;
                                SHEAT[24] := MCP;
                                SHEAT[25] := HDR,[FF]; % DISK ADDRESS OF FILE HEADER

```

```

08270450
08270490
08270500
08270510
08270550
08270600
08270640
08270650
08270660
08270700
08270725
08270740
08270745
08270750
08270800
08271000
08271250
08271500
08271750
08272000
08273250
08273500
08273600
08273750
08274000
08274010
08274020
08274030
08274040
08274050
08274060
08274070
08274250
08274500
08275500
08275750
08276000
08276050
08276100
08276150
08276200
08276205
08276210
08276220
08276230
08276240
08276250
08276260
08276270
08276280
08276290
08276300
08276310
08276320
08276330
08276340
08276350
08276360
08276365
08276370

```



```

      END;
      END;
      COPY:=(COPY:=P)&(NOT COPY = NOT 0)[31:47:1];
%
%
%
BACK UP TAPE. CHECK THE LABEL THEN CALL PRINTORPUNCHWAIT.
      IF (U:=P) < 0 THEN
      BEGIN COPY:=COPY&(P(XCH)=1)[32:40:8];
        IF NOT MTXIN(I,U,B) THEN
        IF (I:=MULTITABLE[U]#"PBTMCP ") AND
        MULTITABLE[U]#"PUTMCP " THEN
        BEGIN STREAM(BUFF); DS:=19 LIT" NOT A BACKUP TAPE*";
          GO TO SPIT;
        END
        ELSE
        IF PRINTORPUNCHWAIT(-U, I&COPY[30:31:17] OR M) THEN
          GO TO OK ELSE BEGIN MS:=1; GO TO BAD END
        ELSE GO TO SPIT;
      END;
%
%
%
BACK UP DISK. SET FIRST REEL NUMBER, IF COPIES OR REEL NUMBER
      GIVEN, DIAL IN "P" BIT, ELSE LEAVE IT OFF TO PRINT ENTIRE
      THING. CHECK FOR THE FILE, THEN CALL PRINTORPUNCHWAIT.
      STREAM(I:=P; U:=U);
      BEGIN SI:=LOC I; DI:=DI+5;
        DS:=3 DEC;
      END;
      I:=P-1;
      IF (COPY OR I),[CF]=0 THEN P(DEL) ELSE
        COPY:=COPY&P(XCH)[39:47:1];
      BUFF:=BUFF,[15:15]-1;
      IF (I:=DIRECTORYSEARCH("PBD " ,U,5))=0 THEN
      IF (I:=DIRECTORYSEARCH("PUD " ,U,5))=0 THEN GO TO BAD
        ELSE MS:=1;
      P(M[I+4]);
      FORGETSPACE(I);
      IF P.[2:1] THEN BEGIN MS:=2; GO TO BAD END;
      IF PBCOUNT LSS 1 THEN PBCOUNT:=1;
      IF PRINTORPUNCHWAIT(U, MS&COPY[30:31:17] OR M) THEN
      FORGETSPACE(BUFF)
      ELSE
      BEGIN MS:=1;
%
%
%
BAD:
      STREAM(MS, XI:=MS<0, U:=IF P(DUP) THEN TINU[U] ELSE U,
        BUFF:=BUFF,[CF]);
      BEGIN DS:=8 LIT" NULL PB";
        SI:=LOC U; CI:=CI+X; GO TO DK;
        SI:=SI+5; DS:=3 CHR; GO TO LL;
      DK: SI:=SI+1; DS:=4 CHR;
        BUFF:=DI; DI:=DI-4; DS:=3 FILL; DI:=BUFF;
      LL: DS:=2 LIT"(";
        CI:=CI+MS; GO TO LO; GO TO L1;
        DS:= 6 LIT"IN USE"; GO TO L;
      L1: DS:=14 LIT"NO OUTPUT UNIT"; GO TO L;
      LO: DS:=11 LIT"NOT ON DISK";
      L : DS:= 2 LIT")*";
      END;
%
%
%
SPIT:
      SPOUT(BUFF);
      END;
      END OF PB KEYBOARD MESSAGE HANDLER;

```

```

08291425
08291450
08291460
08291470
08291475
08291480
08291500
08291750
08292000
08292500
08293000
08293500
08294000
08294500
08295000
08295200
08295600
08295800
08296000
08296160
08296170
08296180
08296190
08296200
08296225
08296250
08296275
08296300
08296325
08296350
08296375
08296400
08296600
08296800
08297000
08297200
08297300
08297400
08297600
08298000
08298200
08298400
08298600
08298800
08299000
08299200
08299400
08299600
08299800
08300000
08300200
08300400
08300600
08300800
08301200
08301400
08301600
08301800
08302000
08302500

```

```

PROCEDURE TIMEOUT (B); VALUE B; REAL B;
  BEGIN INTEGER M,H,C;
    C ← XCLOCK/3600;
    M ← C MOD 60;
    H ← C DIV 60;
    STREAM(H,M,B);
    BEGIN DS ← 9 LIT " TIME IS ";
      SI ← LOC H; DS ← 2 DEC; DS ← 2 DEC;
      DS ← LIT "+";
    END;
    IF SWAPEND≠0 THEN CHANGEDATE(0);
    SPOUTIT(B,TIMEK);
  END;
PROCEDURE GIMEDATE(B,DT); VALUE B,DT; REAL B,DT;
  %% PARAMETER USE IS:
  %% B=OUTPUT AREA FOR MESSAGE OR DATE
  %% DT=0 RECONVERT ACTDATE,WEEKDAY THEN SPOUT TIME MSG
  %% DT>0 SPOUT TIME MSG ONLY
  %% DT<0 CONVERT MMDDYY USING DT (ACTDATE,WEEKDAY NOT CHANGED)
  BEGIN REAL M,D,Y,NCV,NMG;
    REAL SUBROUTINE DAY;
    BEGIN;STREAM(M,X+0,Y+0,Z+0);
      BEGIN DI←LOC X; DS←24 LIT"000+0%1.1Y2G2V3D3T4A4 5>";
        SI←LOC X; SI←SI+M; SI←SI+M;
        DI←LOC M; DI←DI+6; DS←2 CHR;
      END;
      DAY←P;
    END DAY;
    LABEL DAYS;
    LABEL ON;
    IF NOT (NCV←(DT>0)) THEN % NOT PRINT ONLY
    BEGIN
    STREAM(DATE:=IF (NMG:=DT,[1;1]) THEN DT ELSE DATE,R:=[Y]);
      BEGIN SI ← LOC DATE; SI ← SI+3;
        DS←2 OCT; DI←DI-16; DS←3 OCT;
      END;
      IF Y MOD 4 = 0 AND Y ≠ 0 THEN%
        BEGIN IF D = 60 THEN%
          BEGIN M←2; GO ON END;
          IF D > 60 THEN D ←D-1;%
        END;
        FOR M←1 STEP 1 UNTIL 11 DO
          IF DAY≥D THEN GO ON;
        ON: M←M+1;
          D←D-DAY;
          IF M<2 THEN P(Y=1,M+11) ELSE P(Y,M=1);
          P(26,x,2,=,10, IDV,D,+,XCH,P(DUP),[36;10],+,+,7,RDV,5,ISN);
          ; P(DAYS,+,LOD);
          M←M+1;
        END ELSE P(WEEKDAY);
        STREAM(M+[M],NMG,NCV,MDY+[ACTDATE],B,UW+[WEEKDAY]);
        BEGIN NMG(JUMP OUT TO NOMSG);
          SI←LOC M; SI←SI-16;
          NCV(SI+SI+2; JUMP OUT TO NOCNV);
          DS←WDS; SI←SI-6;
        NOCNV: DI←B; DS←9 LIT" DATE IS "; DS←6 CHR;
          DS←5 LIT"DAY, "; B←DI; NCV(JUMP OUT TO NULCV);
        NOMSG: SI←M; NMG(DI←B; JUMP OUT TO NULMS);
          DI←MDY;
          NULMS: DS←4 DEC; DS←2 DEC; DS←2 DEC;

```

```

08305000
08306000
08307000
08308000
08309000
08310000
08311000
08312000
08313000
08314000
08315000
08315100
08316000
08317000
08317100
08317200
08317300
08317400
08317500
08318000
08318100
08318200
08318300
08318400
08318500
08318600
08318700
08318800
08318900
08319000
08319700
08319900
08320000
08321000
08322000
08323000
08324000
08325000
08326000
08327000
08328000
08329000
08330000
08331000
08332000
08332100
08332200
08332300
08332400
08332500
08333000
08334000
08334100
08334300
08334500
08334700
08334900
08335000
08335200
08335400

```



```

NULCV:          NMG(JUMP OUT TO OXIT); DI←B;          08335600
                SI←MDY; SI←SI+2;                    08335800
                DS←2 CHR; 2(DS←LIT "/"; DS←2 CHR);  08336000
                DS ← LIT "←"; SI ←B;                08337000
                3(DI ← B; DS ← FILL; SI ← SI+3; B ←SI);% 08338000
OXIT:           END;                                08339000
                IF NOT NMG THEN SPOUTIT(B,DATEK);    08339100
                P(OXIT);                             08340000
                P(OXIT);                             08340100
DAYS:          " MON", " TUES", "WEDNES", " THURS", " FRI", " SATUR", 08340200
                " SUN";                             08340300
                END;%                                08341000
PROCEDURE DISKLOG(MID,FID,H); VALUE MID,FID,H;REAL MID,FID;ARRAY H[*]; 08341100
BEGIN                                                 08341120
    INTEGER HRS,MIN,SEC,SIX; % DO NOT CHANGE ORDER  08341140
    REAL SEG,T,DT;                                     08341160
    DT:=H[9],[43;5]+9;                                08341180
    FOR T:=10 STEP 1 UNTIL DT DO                       08341200
    IF H[T]≠0 THEN SEG←SEG+H[8];                      08341220
    SIX←(T+H[1],[25;23]) MOD 60;                     08341280
    SEC←(T DIV 60) MOD 60;                            08341300
    MIN←(T DIV 3600) MOD 60;                          08341320
    HRS←T DIV 216000;                                 08341340
    T:=SPACE(10);                                     08341360
    STREAM(A+H[1],[6;18],D+[DT]);                     08341380
    BEGIN SI←LOC A; DS←8 DEC END;                     08341400
    GIMEDATE([DT],[CF],-DT);                          08341420
    STREAM(DT,TIME←[HRS],FN←[MID],USER←[H[2]],SEG←[SEG],T); 08341440
    BEGIN DS←LIT " "; SI←FN; SI←SI+1;                08341460
        DS←7 CHR; DS←LIT "/"; SI←SI+1; DS←7 CHR; DS←LIT "/"; 08341480
        SI←USER; SI←SI+1; DS←7 CHR; DS←LIT "="; SI←SEG; 08341500
        DS←8 DEC; SEG←DI; DI←DI-8; DS←7 FILL; DI←SEG; 08341520
        DS←15 LIT " SEGS←-CREATED "; SI←LOC DT; 08341530
        SI←SI+2; DS←2 CHR; 2(DS←LIT "/"; DS←2 CHR); 08341540
        DS←4 LIT " AT "; SI←TIME; DS←2 DEC; DS←LIT " "; 08341560
        DS←2 DEC; DS←LIT " "; DS←2 DEC; DS←LIT " "; DS←2 DEC; DS←LIT "←"; 08341580
    END;                                               08341600
    IF MID,[6;12] GTR 9 AND % NOT 01SLLL, MAYBE 1SLLL =  %11208341610
        (MID,[6;6] GTR 9 OR MID,[6;6] =0) THEN% NAME OR ONAME, %11208341612
        SPOUTIT(T,DISKLOGGER+DKLOG) ELSE FORGETSPACE(T); %11208341620
    H[1]:=XCLOCK+P(RTR); %11208341622
    STREAM(D:=[DATE],H:=[H[1]],T:=0); %11208341624
    BEGIN SI:=D; DI:=LOC T; DS:=8 OCT; SI:=LOC T; %11208341626
        SI:=SI+5; DI:=H; DI:=DI+1; DS:=3 CHR; %11208341628
    END STREAM; %11208341630
END;                                                  08341640
DEFINE DATEOUT(DATEOUT1)=GIMEDATE(DATEOUT1,0); %CHANGE DATE & SPOUT IT 08342000
PROCEDURE SETDATE(BUFF); VALUE BUFF; REAL BUFF;% 08343000
    BEGIN REAL DY,MN,YR; INTEGER D=DY; REAL B,T=MN; 08344000
        REAL SUBROUTINE C; 08344100
        BEGIN;STREAM(C←0;B+[B]); 08344200
            BEGIN% 08345000
                SI ← B; SI ← SI+5; SI ← SC;% 08346000
                L: IF SC < "0" THEN% 08347000
                    BEGIN IF SC = "←" THEN GO TO X;% 08348000
                        SI ← SI+1; GO TO L;% 08349000
                    END;% 08350000
                K: IF SC ≥ "0" THEN% 08351000
                    BEGIN TALLY ← TALLY+1;% 08352000
                        SI ← SI+1; GO TO K END;% 08353000
            END%
    END

```

DI ← B; B ← SI; SI ← LOC B; DS ← WDS; %	08354000
SI ← B; B ← TALLY; DI ← LOC C; %	08355000
SI ← SI-B; DS ← B OCT; %	08356000
X; END; %	08357000
C ← P;	08358000
END C;	08358100
B ← BUFF; %	08359000
MN ← C; DY ← C; YR ← C;	08360000
BUFF ← BUFF, [15:15] = 1; %	08361000
IF MN > 0 AND MN ≤ 12 AND %	08362000
DY > 0 AND DY ≤ 31 AND %	08363000
YR > 0 THEN %	08364000
BEGIN; STREAM(M ← MN - 1; X ← 0, Y ← 0, Z ← 0);	08365000
BEGIN DI ← LOC X; DS ← 24 LIT "000 + 0%1.1Y2G2V3D3T4A4 5 >";	08365100
SI ← LOC X; SI ← SI + M; SI ← SI + M;	08365200
DI ← LOC M; DI ← DI + 6; DS ← 2 CHR;	08365300
END;	08365400
DY ← P + DY;	08366000
IF YR MOD 4 = 0 AND MN > 2 AND (YR MOD 100 ≠ 0 OR %	08367000
YR MOD 400 = 0) THEN %	08368000
DY ← DY + 1; %	08369000
D ← YR MOD 100 × 1000 + DY; %	08370000
STREAM(D, A ← [DATE]); %	08371000
BEGIN SI ← LOC D; DS ← 8 DEC END; %	08372000
CHANGEDATE(BUFF); %	08373000
END ELSE SPOUT(BUFF);	08374000
END; %	08375000
PROCEDURE CHANGEDATE(BUFF); VALUE BUFF; REAL BUFF; %	08376000
BEGIN REAL B, C, D; %	08377000
SLEEP([TOGGLE], HOLDMASK);	08378000
LOCKTOG(HOLDMASK);	08379000
B := SPACE(30);	08380000
DISKWAIT(-B, -30, DIRECTORYTOP = SYSNO);	08381000
DI := M[B+1]; %	08381100
M[B] ← OPTION;	08382000
M[B+1] ← DATE; %	08383000
M[B+18] := XCLOCK;	08383100
M[B+20], [18:30] := LOGARRAY[33];	08383200
M[B+21] := SCHEDWRD;	08383210
DISKWAIT(B, -30, DIRECTORYTOP = SYSNO);	08384000
IF BUFF ≠ 0 THEN	08384100
BEGIN %	08384200
DATEOUT (BUFF); %	08385000
C := GETSPACE(5, 9, 5) + 2;	08385100
M[C ] := M[C+2] := 0; %	08385200
M[C+3] := D; %	08385300
STREAM(DATE, A := C+1); BEGIN SI := LOC DATE; DS := 8 OCT; END; %	08385400
LINKUP(17, C); %	08385500
END; %	08385600
FORGETSPACE(B); %	08387000
UNLOCKTOG(HOLDMASK);	08388000
END; %	08389000
PROCEDURE SETIME(BUFF); VALUE BUFF; REAL BUFF; %	08390000
BEGIN REAL B = BUFF, T; %	08391000
REAL I, R; %	08392000
LABEL EXIT; %	08393000
REAL CLOCK = XCLOCK; %	08394000
INTEGER CLCK = CLOCK; %	08395000
T ← -1; %	08396000
STREAM(B, T ← [T]); %	08397000

BEGIN SI ← B;%	08398000
L: IF SC = " " THEN%	08399000
BEGIN SI ← SI+1; GO TO L END;%	08400000
IF SC < "0" THEN GO TO X;%	08401000
K: IF SC ≥ "0" THEN%	08402000
BEGIN SI ← SI+1; TALLY ← TALLY+1;%	08403000
GO TO K END;%	08404000
B ← TALLY; SI ← SI-B; DS ← B OCT;%	08405000
X:%	08406000
END;%	08407000
BUFF ← BUFF.[15:15]=1;%	08408000
IF T ≥ 0 AND T DIV 100 < 24 AND T MOD 100 < 60 THEN%	08409000
BEGIN R:=GETSPACE(5,9,5)+2;	08410000
M[R+2]:=XCLOCK;%	08410100
CLCK:= (T DIV 100 × 60 + T MOD 100)×3600;%	08410200
CLOCK ← (CLOCK OR @77)+1;%	08411000
TIMEOUT (BUFF);%	08412000
M[R]:= M[R+3];= 0;%	08412100
STREAM(DATE,A:=R+1);%	08412200
BEGIN SI:=LOC DATE; DS:=8 OCT; END;%	08412300
LINKUP(17,R);%	08412400
GO TO EXIT;%	08413000
END;%	08414000
SPOUT(BUFF);	08415000
EXIT;%	08416000
END;%	08417000
REAL PROCEDURE FORMESS(BUFF,H1); VALUE BUFF,H1; REAL BUFF,H1;	08418000
BEGIN REAL B,H,U;	08418500
INTEGER I;	08418700
LABEL AGAIN,EXIT,AWAY;	08419000
IF H1=0 THEN	08419100
BEGIN STREAM(U:=0,B:=BUFF);	08419150
BEGIN SI:=B;	08419200
L: IF SC=" " THEN	08419250
BEGIN SI:=SI+1; GO TO L END;	08419300
B:=SI;	08419320
IF SC GEQ "0" THEN	08419350
IF SC LEQ "9" THEN TALLY:=1;	08419400
U:=TALLY;	08419450
END;	08419500
BUFF:=P;	08419520
IF P THEN	08419550
BEGIN SCHEDLOOK(BUFF,2);	08419600
GO AWAY;	08419650
END;	08419700
END;	08419750
AGAIN: U:=FORMESS:=UNITIN(TINU,BUFF);	08420000
IF US31 THEN BEGIN SLEEP([TOGGLE],STATUSMASK);	08421000
IF LABELTABLE[U] < 0 THEN%	08422000
BEGIN STREAM(A:=TINU[U],B:=B:=SPACE(5));	08424000
BEGIN SI ← LOC A; SI ← SI + 5; DS ← 3 CHR;%	08425000
DS:=24LIT" IN USE(TO BE READIED)+";	08426000
END;%	08427000
SAVEWORD := SAVEWORD AND NOT TWO(U);	08427100
SPOUT(B);	08428000
IF H1 THEN GO AWAY ELSE GO TO EXIT;	08429000
END;	08429500
LABELTABLE[U]:=@114&H1[1:47:1];	08430000
MULTITABLE[U] ← 0;%	08431000
I ← TWO(U);%	08432000

```

IF H1 THEN B:=NOT 0 ELSE                                08433000
BEGIN B:=NOT 1; H:=I:=0;                                08434000
  IF U=23 THEN H:=P(.READER);                          %R7608434100
  IF U=24 THEN H:=P(.READERB);                         %R7608434200
  IF H NEQ 0 THEN                                       %R7608434300
  BEGIN UNITCODE[U=23]:=0;                             %R7608434310
    IF (*H).[CF] NEQ 0 THEN                             08434400
    BEGIN                                               %R7608434500
      FORGETSPACE(*H=2);                               %R7608434600
      M[H]:=0;                                         %R7608434700
    END;                                               %R7608434800
  END;                                               %R7608434900
END;
READY ← READY AND B OR I;%                             08434910
RRRMECH ← RRRMECH AND B OR I;%                         08435000
SAVEWORD ← SAVEWORD AND B OR I;%                       08436000
END;                                                    08437000
EXIT: IF NOT H1 THEN                                    08437050
BEGIN IF U GTR 31 THEN                                  08437100
  BEGIN STREAM(BUFF,B:=B:=SPACE(5));                   08437150
    BEGIN DS:=10 LIT"INV KBD RY";                       08437200
      SI:=BUFF; DS:=3 CHR;                              08437250
      DS:=LIT"*";                                       08437300
    END;                                                08437350
    SPOUT(B);                                           08437400
  END;                                                 08437450
  STREAM(OK:=0,BUFF);                                   08437500
  BEGIN SI:=BUFF;                                       08437550
    3(IF SC=" " THEN JUMP OUT;                          08437600
      IF SC="," THEN JUMP OUT;                         08437650
      IF SC="*" THEN JUMP OUT TO L3;                   08437700
      SI:=SI+1);                                       08437750
    L1: IF SC=" " THEN                                  08437800
    L2: BEGIN SI:=SI+1; GO TO L1 END;                   08437850
      IF SC="," THEN GO TO L2;                         08437900
      BUFF:=SI;                                        08437950
      IF SC≠"*" THEN TALLY:=1;                        08438000
    L3: OK:=TALLY;                                     08438050
    END;                                               08438100
    BUFF:=P;                                           08438150
    IF P THEN GO AGAIN;                                08438200
    FORMESS:=1;                                        08438250
  END;                                                 08438300
  AWAY: END;                                           08438350
  END;                                                 08438500
PROCEDURE SUSTATUS(A,DDD,B); VALUE A,DDD,B; REAL A,B;  08438900
  FORWARD;                                             08438910
PROCEDURE OUTPUTLABEL(B); VALUE B; REAL B;%          08439000
  BEGIN REAL BU=B,U,T,A;%                              08440000
    REAL G,Q;%                                         08441000
      REAL TEMP;                                       08441050
    BOOLEAN SCRTOG;                                    08441100
    REAL BLURB,MIX;                                    08441200
    LABEL EXIT;%                                       08442000
    SUBROUTINE DOIT;%                                   08443000
      BEGIN; STREAM(A+TINU[U];B);%                     08444000
        BEGIN SI ← LOC A; SI ← SI+5; DS ← LIT " ";%    08445000
          DS ← 3 CHR; DS ← LIT " "; A ← DI END;%      08446000
        A ← P; T ← LABELTABLE[U];%                    08447000
        IF U LSS 16 THEN TEMP:=PRNTABLE[U],[30:18];%  08447100
        IF T=0 THEN                                     08448000

```

```

STREAM(BI=TEMP,VI=(U LSS 16),A); 08448050
BEGIN SI:=LOC V;SI:=SI+7; 08448100
IF SC NEQ "0" THEN BEGIN SI:=LOC B;DS:=5DEC END; 08448110
          DS:=9LIT" SCRATCH;" END 08448150
ELSE IF T = @114 OR T = @214 THEN% 08449000
  BEGIN 08450000
  STREAM(SAV:=(TWO(U) AND SAVEWORD) NEQ 0),A); 08450100
  BEGIN 08450200
  DS:=10LIT"NOT READY;" 08450300
  SAV(DI:=DI-1; DS:=8LIT"(SAVED)"); 08450400
  END 08450500
  END 08450600
ELSE IF ABS(T)=@314 THEN 08451000
STREAM(BI=TEMP,VI=(U LSS 16),A); 08451100
BEGIN SI:=LOC V; SI:=SI+7; 08451200
IF SC NEQ "0" THEN BEGIN SI:=LOC B;DS:=5DEC END; 08451210
          DS:=11 LIT " UNLABELED;" 08452000
END 08452100
ELSE BEGIN;% 08453000
  STREAM(K:=T<0; TEMP, VI:=U<16, A); 08454000
  BEGIN V(SI:=LOC TEMP; DS:=5 DEC; DS:=LIT" "); 08454500
  CI:=CI+K; GO TO LAB; 08455000
  DS:=6 LIT"IN USE"; GO TO L; 08455500
  LAB: DS:=7 LIT"LABELED"; 08456000
  LI DS:=LIT" "; K:=DI; 08457000
  END; 08458000
A ← P;% 08459000
IF (MIX ← RDCTABLE[U],[8:6]) ≠ 0 THEN 08460000
IF JARROW[MIX] ≠ 0 THEN 08461000
  BEGIN TABCNT[MIX]:=TABCNT[MIX]+1; 08461250
  BLURB:=PUTORTAKE(MIX,[JAR[MIX,0]],10,0); 08461500
  TABCNT[MIX]:=TABCNT[MIX]-1; 08461750
  STREAM(J:=BLURB; MIX, A); 08462000
  BEGIN DS ← 3 LIT "BY "; SI ← J;% 08463000
  SI ← SI+1; DS ← 7 CHR;% 08464000
  DS ← LIT "/"; SI ← SI+1;% 08465000
  DS ← 7 CHR; DS ← LIT "=";% 08466000
  SI ← LOC MIX; DS ← 2 DEC; 08467000
  DS ← LIT "i"; J ← DI;% 08468000
  DI←DI-3; DS←FILL; 08468500
  END;% 08469000
  A ← P;% 08470000
  FORGETSPACE(BLURB); 08470100
  END ELSE ELSE 08471000
IF T<0 AND (U=23 OR U=24) THEN 08471010
  BEGIN 08471020
  STREAM(SI=0 ; A); 08471030
  BEGIN 08471040
  DS:=22LIT"BY AUTO LOAD CONTROL; "; 08471050
  SI:=DI; 08471060
  END; 08471070
  A:=P; 08471080
  END ELSE 08471090
IF U GEQ 20 AND U LEQ 22 THEN 08471100
IF LABELTABLE[U],[1:5]=@21 THEN 08471105
  BEGIN STREAM(SI=0;A); 08471110
  BEGIN 08471120
  DS:=13 LIT "BY SCHEDULED "; 08471130
  DS:=13 LIT "PRNPBT/DISK; "; 08471140
  SI:=DI; 08471150

```

END;	08471160
A:=P;	08471170
END)	08471180
STREAM(S+0;K+MULTITABLE[U],T,	08472000
R+RDCTABLE[U],[14:10],D+RDCTABLE[U],[24:17],	08473000
C+RDCTABLE[U],[41:7],A);	08473500
BEGIN SI ← LOC K;	08474000
2(SI ← SI+1; DS ← 7 CHR; DS ← LIT " ");%	08475000
DS ← 3 DEC; DS ← LIT " ";%	08476000
DS ← 5 DEC; DS ← LIT " ";%	08477000
DS ← 2 DEC;	08478000
DS ← LIT"+";	08478300
S←DI;	08478500
END;	08478600
A+P;	08478700
IF U≥32 THEN IF CIDROW[U -32]≠0 THEN	08478800
STREAM(DK+CIDTABLE[U -32,2],A);	08478900
BEGIN	08479000
DI←DI-1;	08479100
\$ SET OMIT = NOT(PACKETS)	08479109
DS:=5 LIT ",PKT ";	08479110
\$ POP OMIT	08479111
\$ SET OMIT = PACKETS	08479199
DS←6 LIT ",DECK ";	08479200
\$ POP OMIT	08479201
SI←LOC DK; SI←SI+1;	08479300
DS←7 CHR;	08479400
END;	08479500
END;	08479600
SPOUT(B);	08480000
B ← 0;%	08481000
END;%	08482000
IF (U ← UNITIN(TINU,BU)) ≤ 35 THEN	08482050
BEGIN B ← B,[15:15]-1;%	08483000
IF (U OR 1)=19 THEN SUSTATUS(B,0,U) ELSE	08484000
DOIT;%	08484500
GO TO EXIT;%	08485000
END;%	08486000
\$ SET OMIT = SHAREDISK	08487000
SCRTOG ← U=36;	08487099
\$ POP OMIT	08487100
\$ SET OMIT = NOT(SHAREDISK)	08487101
SCRTOG:=(U=40);	08487199
\$ POP OMIT	08487200
STREAM(A+0;B);%	08487201
BEGIN SI ← B;%	08488000
L: IF SC = " " THEN%	08489000
BEGIN SI ← SI+1; GO TO L END;%	08490000
DI ← LOC A; DI ← DI+6; DS ← 2 CHR;%	08491000
END;%	08492000
Q ← P; B ← B,[15:15]-1;%	08493000
FOR U ← 0 STEP 1 UNTIL 35 DO	08494000
IF TINU[U],[30:12] = Q THEN%	08495000
IF (G ← LABELTABLE[U])≠0 AND G≠@114 AND G≠@214	08496000
AND NOT SCRTOG OR G=0 AND SCRTOG THEN	08497000
BEGIN IF B=0 THEN B := SPACE(10);	08497100
DOIT;%	08498000
END;%	08499000
IF B ≠ 0 THEN%	08500000
	08501000

```

BEGIN;STREAM(Q,B);%                                08502000
    BEGIN DS ← 6 LIT " NULL ";%                    08503000
        SI ← LOC Q; SI ← SI+6; DS ← 2 CHR;%        08504000
        DS ← 7 LIT " TABLE";%                    08505000
    END;%                                            08506000
    SPOUT(B);                                        08507000
END;%                                              08508000
EXIT; END;%                                        08509000
PROCEDURE TIMEUSED(B,X); VALUE B,X; REAL B,X;%    08525000
    BEGIN INTEGER H,M,S,Q,T;                        08526000
    $ SET OMIT = NOT(PACKETS)                       08526499
    DEFINE UNITNO = PSEUDOMIX[X];%                 08526500
    $ POP OMIT                                       08526501
    T ← PUTORTAKE(X,[JAR[X,3]],1,0)                 08527000
      + PUTORTAKE(X,[PROCTIME[X]],1,0);%           08527100
    IF X=P2MIX THEN T ← T+CLOCK+P(RTR);%           %WF 08528000
    FORMTIME([M],T);                                08529000
    T←((CLOCK+P(RTR))/60)-PUTORTAKE(X,             08533100
      [UV[X,9]],1,0),[1:17];%                     08533150
    FORMTIME([Q],T×60);                             08533200
    STREAM(J+PUTORTAKE(X,[JAR[X,0]],1,0),          08534000
      K+PUTORTAKE(X,[JAR[X,1]],1,0),X,H+[H],B);%   08534100
    BEGIN DS ← 10 LIT " TIME FOR ";%               08535000
        SI←LOC J; 2(SI←SI+1; DS←7 CHR; DS←LIT "/" );% 08536000
        DI←DI-1; DS←LIT "="; DS←2 DEC;             08537000
                                                08538000
        X←DI; DI←DI-2; DS←FILL; DI←X;              08538500
        DS ← 3 LIT " IS";%                          08539000
        SI←H; SI←SI+8; DS←8 CHR;                   08540000
                                                08540100
        DS ← 3 LIT " IN";                           08540200
        SI←SI+8; DS←8 CHR;                          08540300
        DS←LIT" ";%                                  08541000
    END;%                                            08542000
    SPOUTER(B,UNITNO,1);%                           %WF 08543000
END;%                                              08544000
REAL PROCEDURE ANVIL(IL,Z); VALUE IL,Z; REAL IL,Z;% 08545000
    BEGIN REAL B=Z,U=+1;%                           08546000
    REAL ZZ;%                                        08547000
    LABEL EXIT;%                                    08547050
    ZZ:=Z;%                                         08547100
    NAMEID(U,ZZ);%                                  08547200
    NAMEID(U,ZZ);%                                  08547300
    IF U="/ " THEN%                                  08547400
    BEGIN U:=Z,[15:15]; GO EXIT END ELSE%           08547500
    IF (U ← UNITIN(TINU,B)) ≤ 35 THEN%              08547600
    BEGIN%                                           08548000
    IF LABELTABLE[U] = @114 OR LABELTABLE[U] = @214 THEN% 08549000
    BEGIN%                                           08550000
    STREAM(A:=TINU[U],SAVI:=((TWO(U) AND SAVEDWORD) ≠ 0 ),% 08551000
      X:=Z,[15:15]-1);%                             08551100
    BEGIN%                                           08551200
    SI:=LOC A; SI:=SI+5; DS:=3CHR;                  08551300
    DS:=11LIT" NOT READY";%                         08552000
    SAV(DI:=DI-1; DS:=8LIT"(SAVED)");%              08553000
    END;%                                            08554000
    U←36;%                                          08554100
    END ELSE%                                       08555000
    IF LABELTABLE[U] < 0 THEN%                       08556000
    08557000

```

BEGIN;STREAM(A+TINU[U],X+Z,[15:15]-1);%	08558000
BEGIN SI ← LOC A; SI ← SI+5;%	08559000
DS ← 3 CHR; DS ← 8 LIT " IN USE←";%	08560000
END;%	08561000
U←36;	08562000
END;%	08563000
IF U ≤ 35 THEN	08564000
LABELTABLE[U] ← -(IF IL THEN *P(DUP) ELSE @314);%	08565000
EXIT; END; END;	08566000
PROCEDURE SAVETHEUNIT(B); VALUE B; REAL B;%	08575000
BEGIN REAL A=B,T,U;I;	08576000
LABEL AGAIN,EXIT;	08576100
STREAM(U:=0,B);	08576150
BEGIN SI:=B;	08576200
L: IF SC=" " THEN BEGIN SI:=SI+1; GO TO L END;	08576250
B:=SI;	08576300
IF SC GEQ "0" THEN	08576320
IF SC LEQ "9" THEN TALLY:=1;	08576350
U:=TALLY;	08576400
END;	08576450
B:=P;	08576500
IF P THEN BEGIN SCHEDLOOK(B,3); GO TO EXIT END;	08576520
T:=SPACE(10);	08576550
IF (U:=UNITIN(TINU,A)) GTR 31 THEN	08577000
STREAM(A,T);	08578000
BEGIN DS:=10 LIT"INV KBD SV";	08578100
SI:=A; DS:=3 CHR;	08578200
DS:=LIT"←";	08578300
END ELSE	08578400
BEGIN I ← TWO(U);%	08578500
SLEEP([TOGGLE],STATUSMASK);	08579000
IF LABELTABLE[U] ≥ 0 THEN%	08580000
BEGIN LABELTABLE[U] ← @114;%	08581000
MULTITABLE[U]←RDCTABLE[U]←0;	08582000
RRRMECH ← RRRMECH OR I;%	08582100
READY ← READY OR I;%	08583000
SAVEWORD := SAVEWORD OR I;	08584000
I ← " ";	08584100
END;	08585000
ELSE BEGIN SAVEWORD ← SAVEWORD OR I;%	08586000
I ← " TO BE";%	08587000
END;%	08588000
STREAM(A+TINU[U],I,T);%	08589000
BEGIN DS ← LIT " ";%	08590000
SI ← LOC A; SI ← SI+5; DS ← 3 CHR;%	08591000
SI ← SI+2; DS ← 6 CHR;%	08592000
DS ← 7 LIT " SAVED←";%	08593000
END;%	08594000
END;%	08595000
END;%	08596000
SPOUT(T);	08597000
STREAM(OK:=0,A);	08597050
BEGIN SI:=A;	08597100
3(IF SC=" " THEN JUMP OUT;	08597150
IF SC="," THEN JUMP OUT;	08597200
IF SC="←" THEN JUMP OUT TO L3;	08597250
SI:=SI+1);	08597300
L1: IF SC=" " THEN	08597350
L2: BEGIN SI:=SI+1; GO TO L1 END;	08597400
IF SC="," THEN GO TO L2;	08597450

AGAIN:



```

                A:=SI;
                IF SC#"+ " THEN TALLY:=1;
L3:   OK:=TALLY;
      END;
      A:=P;
      IF P THEN GO AGAIN;
EXIT:
      END;%
BOOLEAN PROCEDURE WHYSLEEP(MASK); VALUE MASK; REAL MASK;
BEGIN
  REAL A, B;
  IF REPLY[P1MIX]=VWY THEN
    BEGIN
      B:=SPACE(KEYMSGSZ);
      DISKWAIT(-B,KEYMSGSZ,MESSAGETABLE[2],[22:26]);
      STREAM(B,MASK,T:=0,O:=0,D:=0,A:=A:=SPACE(4));
      BEGIN
        SI:=LOC MASK;
        B(IF SC="0" THEN GO TO NEXT;
          IF SC=VWY THEN
            BEGIN
              DI:=A; DS:=3LIT" DS"; A:=DI; GO TO NEXT;
            END;
            T:=SI; DI:=LOC 0; DI:=DI+7; DS:=CHR;
            SI:=LOC 0; DI:=LOC D; DI:=DI+6; DS:=2DEC;
            SI:=B;
          R: SI:=SI+6; DI:=DI-2;
            IF SC="*" THEN % END OF FIRST PART OF TABLE
              BEGIN
                SI:=T; GO TO NEXT;
              END;
            IF 2SC NEQ DC THEN GO TO R;
            SI:=SI-6; DI:=A; DS:=LIT" "; DS:=2CHR; A:=DI; SI:=T;
          NEXT: SI:=SI+1);
            DI:=A; DS:=LIT"+ ";
          END STREAM STATEMENT;
          SPOUT(A);
          FORGETSPACE(B);
          END % IF "WY"
        ELSE WHYSLEEP:=TRUE;
      END PROCEDURE WHYSLEEP;
PROCEDURE CHANGEOPTION(BUFF,RS);%
  VALUE BUFF,RS;REAL BUFF,RS;%
BEGIN
  REAL B,T,OP,BUS,MASK,OPTER;
  SLEEP([TOGGLE],HOLDMASK);
  LOCKTOG(HOLDMASK);
  BUS := BUFF,[15:15]=1; B := SPACE(30);
  DISKWAIT(-B,30,DIRECTORYTOP=SYSNO);
  OPTER ← SPACE(OPTIONSZ);
  DISKWAIT(-OPTER,OPTIONSZ,MESSAGETABLE[0],[22:26]);
  STREAM(BUFF,T=0,OPTER,R+{OP});%
  BEGIN%
    SI ← BUFF; 63(IF SC=" " THEN SI ← SI+1 ELSE JUMP OUT%
      TO L); L: IF SC GEQ "0" THEN GO L4;
    63(IF SC# " " THEN SI←SI+1); 16(IF SC=" " THEN SI←SI+1);%
    DI←LOC T;%
    B(IF SC=" " THEN JUMP OUT TO L1 ELSE%
      IF SC="+" THEN JUMP OUT TO L1 ELSE%
      IF SC>"0" THEN JUMP OUT TO L1 ELSE%
      DS←1 CHR);L1;%

```

```

08597500
08597550
08597600
08597650
08597700
08597750
08597900
08598000
08599000
08600000
08601000
08602000
08603000
08604000
08604100
08605000
08606000
08607000
08608000
08609000
08610000
08611000
08612000
08613000
08614000
08615000
08616000
08617000
08617500
08618000
08618500
08619000
08619500
08620000
08620500
08621000
08621500
08621600
08622000
08622500
08623000
08624000
08625000
08626000
08627000
08628000
08629000
08630000
08631000
08631100
08632000
08633000
08634000
08635000
08636000
08637000
08638000
08639000
08640000
08641000

```



SLEEP([TOGGLE],HOLDMASK);	08689200
LOCKTOG(HOLDMASK);	08689300
VASE:=KTR,[15:15]-1;	08689900
OPTER = SPACE(OPTIONSZ)&OPTIONSZ[8:38:10];	08690000
DISKWAIT(=OPTER,OPTIONSZ,MESSAGETABLE[0],[22:26]);	08690020
IF PO THEN	08690080
BEGIN	08690090
STREAM(BUFF:=KTR, T:=0, OPTER, D:=IN);	08690100
BEGIN SI+BUFF;63(IF SC=" " THEN SI+SI+1 ELSE JUMP OUT TO L);L;	08690110
IF SC GEQ "0" THEN GO TO L4;	08690120
DI+LOC T;	08690130
8(IF SC=" " THEN JUMP OUT TO L1 ELSE	08690140
IF SC="+" THEN JUMP OUT TO L1 ELSE	08690150
IF SC>"0" THEN JUMP OUT TO L1 ELSE	08690160
DS+1 CHR); L1;	08690170
TALLY+0; BUFF+SI; SI+OPTER;	08690180
63(DI+LOC T;IF 8 SC#DC THEN	08690190
BEGIN IF SC="+" THEN	08690200
BEGIN TALLY+48; JUMP OUT TO L3 END	08690210
ELSE TALLY+TALLY+1	08690220
END ELSE JUMP OUT TO L2); TALLY+48;GO TO L3;L2;	08690230
IF SC="+" THEN BEGIN SI+BUFF;63(IF SC<"0" THEN SI+SI+1	08690240
ELSE JUMP OUT TO L4); L4; DI+LOC T; SI+SI+1;	08690250
IF SC<"0" THEN BEGIN SI+SI-1; DS+1 OCT END ELSE	08690260
BEGIN SI+SI-1; DS+2 OCT END;	08690270
TALLY+47; T(TALLY+TALLY+63);	08690280
END;	08690290
L3; T+TALLY; SI+LOC T; DI+D; DS+ WDS;	08690300
END;	08690310
IF N LSS OPTER,[8:10] THEN P(SETT,DEL);	08690400
SPOUT(VASE);	08690600
END ELSE	08690800
BEGIN	08691000
STREAM(KTR);	08691200
BEGIN SI:=KTR;	08691400
IF SC="S" THEN TALLY:=1 ELSE	08691600
IF SC="R" THEN TALLY:=2 ELSE TALLY:=3;	08691800
KTR:=TALLY;	08692000
END;	08692200
XI#P; NI#-1;	08692400
NI#N+1;	08692600
IF ((KTR:=SETT) AND X) # 0 THEN	08692800
BEGIN SPOUT(VASE);	08693000
VASE:=SPACE(3);	08693200
GO TO INCR;	08693400
END;	08693600
IF KTR#0 THEN	08693800
IF X#3 THEN	08694000
BEGIN STREAM(X:=X+1, VASE);	08694200
BEGIN DS:=12 LIT" ALL OTHERS ";	08694400
X(DS:=4 LIT"NOT ");	08694600
DS:=4 LIT"SET=";	08694800
END;	08695000
SPOUT(VASE);	08695200
END ELSE FORGETSPACE(VASE)	08695400
ELSE GO TO INCR;	08695600
END;	08695800
FORGETSPACE(OPTER);%	08696000
UNLOCKTOG(HOLDMASK);	08696100
END;%	08697000



```

HEADER[7] := 0;                                08712500
HEADER[3] := XCLOCK + P(RTR);                  08713000
STREAM(ONE:=1, H:=HEADER[6]);                  08713250
BEGIN SI:=LOC ONE; DS:=8 ADD;                  08713500
      DI:=DI+24; 20(DS:=8 LIT"0");             08713750
END;                                            08714000
M[H+7]*PBDROWSZ DIV 3;                         08714110
HEADER[9]*M[H+9]+1;                            08714120
HEADER[10]*M[H+10]*GETUSERDISK(-(PBDROWSZ+1)); 08714130
M[HEADER INX NOT 0] := EUF(-(IF I THEN "PUD   " 08714140
      ELSE "PBD   "),HEADER[6],H-1);          08714150
FORGETSPACE(H);                                08714170
FILEMESSAGE((IF I THEN "PUD   " ELSE          08714300
      "PBD   ")&HEADER[6][24:6:24],          08714310
      "OUT   " &HEADER[6][30:30:18],          08714320
      0,"   ",0,0,0,                          08714330
      (PBDREL OR OPNMESS) OR OPENK);          08714340
END;                                            08714400
END ELSE                                        08714500
      IF HEADER[7] MOD PBDRECS=0 THEN %GET NEW ROW 08715000
      BEGIN H:=SPACE(30); SI=M[HEADER INX NOT 0]; 08715100
      DISKWAIT(-H,30,S);                       08715200
      HEADER[9+HEADER[9]]+*P(DUP)+1)+          08716000
      GETUSERDISK(-(PBDROWSZ+1));             08716010
      M[H+9+HEADER[9]]+HEADER[9+HEADER[9]];    08716100
      M[H+9]+HEADER[9];                        08716110
      M[H+7]+HEADER[7] + PBDROWSZ DIV 3;       08716200
      DISKWAIT(H,30,S);                        08716300
      FORGETSPACE(H);                          08716500
      END;                                     08716600
      STREAM(A+I+HEADER[HEADER[9]+9]+(HEADER[7] MOD 08716800
      PBDRECS)*3,D+POINTER,[CF]=1);          08717000
      BEGIN SI+LOC A; DS+8 DEC END;            %P 08720000
      HEADER[7]+(*P(DUP))+1;                   %P 08721000
      IOD*@141330100477777;                    08722000
      END ELSE %% ON TAPE %%                    %P 08723000
      IOD*@21320500000000&M[POINTER INX NOT 1][3:14:4]; %P 08724000
      IOREQUEST(M[ALPHA],POINTER INX IOD&ALPHA[24:1:1], %P 08726000
      M[POINTER INX NOT 1]);                   %P 08727000
      M[T]+IOD INX M[T]&0[26:26:7]&0[19:19:1] AND NOT M; %P 08728000
      IF H LSS 0 THEN                           08728500
      BEGIN TERMINATE(P1MIX);                   08728600
      TERMINALMESSAGE(H);                       08728700
      END;                                     08728800
      END PBIO;                                %P 08729000
PROCEDURE TIMERELAXER(KTR,TYPE,MIX);%          08730000
VALUE      KTR,TYPE,MIX;%                     08731000
REAL      KTR,TYPE,MIX;%                     08732000
      BEGIN INTEGER BUFF,PRT,IOT,T,P1,I1;%     08733000
      LABEL SPIT;%                             08734000
      ARRAY V[*],VU[*];                       08734100
      $ SET OMIT = NOT(PACKETS)                08734499
      DEFINE UNITNO = PSEUDOMIX[MIX]#;        08734500
      $ POP OMIT                               08734501
      BUFF + KTR,[15:15]=1;%                   08735000
      V:=M[PUTORTAKE(MIX,[JAR[MIX,0]],5,0)]&5[8:38:10]; 08735100
      IF TYPE#VTL THEN                         08736000
      BEGIN                                     08736500
      STREAM(IOT+0,PRT+0,CODE+0; KTR);%       08737000

```

```

BEGIN SI+KTR; IF SC=" " THEN BEGIN L1: SI+SI+1;% 08738000
  IF SC=" " THEN GO L1 END; IF SC="*" THEN GO L5;% 08739000
  IF SC="," THEN GO L2; IF SC<"0" THEN GO EXIT;% 08740000
  KTR+SI; L3: TALLY+TALLY+1; SI+SI+1; 08741000
  IF SC>"0" THEN GO L3; SI+KTR; CODE+TALLY; 08742000
  DI+LOC PRT; DS+CODE OCT; TALLY+0;% 08743000
L5: IF SC=" " THEN BEGIN L4: SI+SI+1;% 08744000
  IF SC=" " THEN GO L4 END; IF SC="," THEN GO L2;% 08745000
  IF SC="*" THEN TALLY+1; GO EXIT;% 08746000
L2: SI+SI+1; IF SC=" " THEN BEGIN L6: SI+SI+1;% 08747000
  IF SC=" " THEN GO L6 END; KTR+SI;% 08748000
  IF SC="*" THEN BEGIN TALLY+1; GO EXIT END;% 08749000
  IF SC<"0" THEN GO EXIT; L7: TALLY+TALLY+1;% 08750000
  SI+SI+1; IF SC>"0" THEN GO L7; DI+LOC IOT;% 08751000
  SI+KTR; CODE+TALLY; DS+CODE OCT; TALLY+1;% 08752000
EXIT: CODE+TALLY;% 08753000
END STREAM;% 08754000
IF NOT P THEN GO SPIT;% 08755000
PRT + P*3600; IOT + P*3600;% 08756000
VU1=[M[PUTORTAKE(MIX,[UV[MIX,0]],14,0)]]&14[8:38:10]; 08757000
IF PRT#0 THEN 08758000
BEGIN 08759000
  T1=(TYPE=VCT)*V[3]-PRT; 08760000
  NT1:=PUTORTAKE(MIX,[UV[MIX,1]],0,VU[1]+T); 08761000
  NT1:=PUTORTAKE(MIX,[UV[MIX,13]],0,VU[13]+T); 08762000
  V[3]:=(P(DUP))-T; 08763000
END; 08764000
IF IOT#0 THEN 08765000
BEGIN 08766000
  T1=(TYPE=VCT)*V[4]-IOT; 08767000
  NT1:=PUTORTAKE(MIX,[UV[MIX,0]],0,VU[0]+T); 08768000
  NT1:=PUTORTAKE(MIX,[UV[MIX,11]],0,VU[11]+T); 08769000
  V[4]:=(P(DUP))-T; 08770000
END; 08771000
STREAM(TEST+0: X+[V[0]],MIX,Z+PRT#0,I+IOT#0,% 08774000
  K1=TYPE=33, T1:=T1+VU,[CF]); % 33=XT 08775000
BEGIN DS+LIT " "; Z(DS+4 LIT "PRT "; TALLY+1;% 08776000
  I(DS+4 LIT"AND "); I(DS+4 LIT "IOT "; TALLY+1);% 08777000
  DS+8 LIT "ESTIMATE"; Z(I(DS+LIT "S"));% 08778000
  DS+8LIT" CHANGED"; K(DI+DI-7; DS+8LIT"EXTENDED");% 08779000
  DS+5LIT" FOR"; SI+X; SI+SI+1; DS+7CHR; SI+SI+1;% 08780000
  DS+LIT"/"; DS+7CHR; DS+LIT"="; SI+LOC Z;% 08781000
  SI+SI-8; DS+2DEC; DS+LIT"+"; TEST+TALLY; 08782000
  DI+DI-3; DS+FILL; 08782500
END STREAM;% 08783000
IF P THEN SPOUTER(T,UNITNO+1) ELSE 08784000
FORGETSPACE(T); 08784100
END; 08785000
IOT + PRT + =0;% 08786000
IF P(V[3],DUP) GEQ @7777777777 THEN P(DEL) ELSE 08787000
P1 + (PRT + P DIV 3600)=60*(PRT + PRT DIV 60);% 08788000
IF P(V[4],DUP) GEQ @7777777777 THEN P(DEL) ELSE 08789000
I1 + (IOT + P DIV 3600)=60*(IOT + IOT DIV 60);% 08790000
STREAM(X+[V[0]],MIX,PRT,P1,IOT,I1,BUFF); 08791000
BEGIN DS+17LIT" TIME LIMITS FOR"; SI+X; SI+SI+1; DS+7CHR;% 08792000
  DS+LIT"/"; SI+SI+1; DS+7CHR; DS+LIT"="; SI+LOC MIX; 08793000
  DS+2DEC; MIX+DI; DI+DI-2; DS+FILL; DI+MIX; 08793500
  DS+10LIT" ARE! PRT="; IF SC="*" THEN 08794000
  BEGIN SI+SI+16; DS+8LIT"NO LIMIT" END ELSE BEGIN% 08795000
  DS+8DEC; DS+LIT"!"; DS+2DEC; BUFF+DI; DI+DI-11;% 08795500

```

```

DS=7FILL; DI=BUFF END; DS=6LIT"; IOT="; IF SC="+" THEN 08796000
DS=10LIT"NO LIMIT,+" ELSE BEGIN DS=8DEC; DS=LIT"i";% 08796500
DS=2DEC; DS=2LIT",+"; DI=DI-13; DS=7FILL END; 08797000
END STREAM;% 08797500
SPIT;% 08798000
V[0]=V[3]; V[1]=V[4]; 08798050
T:=PUTORTAKE(MIX,[JAR[MIX,3]],=-2,V INX 0); 08798100
SPOUTER(BUFF,UNITNO,1); 08798500
END TIMERELAXER; 08799000
PROCEDURE CHANGEFACTOR(BUFF,TF); VALUE BUFF,TF; REAL BUFF; BOOLEAN TF; 08800000
BEGIN REAL FACTOR,B,T; INTEGER TEMP=T; 08801000
LABEL TYPEOUT,EXIT; 08802000
BUFF = ((B*BUFF),[15:15]=1)&M[P(DUP)=1][9:9:9]; 08802500
IF TF THEN GO TYPEOUT; 08803000
STREAM(ANS=0:B); 08804000
BEGIN SI=B; DI=LOC B; DS=8LIT"00000000"; DI=DI-2; 08805000
L1 IF SC = " " THEN BEGIN SI=SI+1; GO TO L END; 08806000
IF SC < "0" THEN GO TO L1; 08807000
IF SC > "9" THEN GO TO L1; 08808000
SI=SI+1; 08809000
IF SC < "0" THEN GO TO ONECHR; 08810000
IF SC ≤ "9" 08810500
THEN BEGIN SI=SI-1; DI=DI-2; DS=2 CHR; END 08811000
ELSE ONECHR; BEGIN SI=SI-1; DI=DI-1; DS=1 CHR; END; 08812000
IF SC = "," THEN GO TO L2 ELSE GO TO L3; 08813000
L1: IF SC ≠ "," THEN GO TO ERROR; 08814000
L2: SI=SI+1; 08815000
IF SC < "0" THEN GO TO ERROR; 08816000
IF SC > "9" THEN GO TO ERROR; 08817000
DS=CHR; 08818000
IF SC ≥ "0" THEN IF SC ≤ "9" THEN DS=CHR; 08819000
L3: IF SC = " " THEN GO CONVERT; 08820000
IF SC = "+" THEN GO CONVERT; 08821000
ERROR:DI=LOC ANS; SKIP 1 DB; DS= 10 SET; GO TO EXITS; 08822000
CONVERT: SI=LOC B; SI=SI+4; DI=LOC ANS; DS=4 OCT; 08823000
EXITS: 08824000
END STREAM; 08825000
P(,FACTOR,+); 08826000
IF FACTOR < 0 THEN GO TO EXIT; 08828000
CORE,[4:14] + FACTOR; 08829000
SLEEP([TOGGLE],HOLDMASK); LOCKTOG(HOLDMASK); 08830000
B := SPACE(30); 08831000
DISKWAIT(=B,-30,DIRECTORYTOP=SYSNO); 08832000
M[B+9] + CORE; * CHANGE FACTOR 08833000
DISKWAIT(B,-30,DIRECTORYTOP=SYSNO); 08834000
FORGETSPACE(B); 08835000
UNLOCKTOG(HOLDMASK); 08836000
SELECTION; 08836500
TYPEOUT: 08837000
STREAM(I+(FACTOR+CORE,[4:14]) DIV 100, FR+(TEMP+FACTOR MOD 100), 08838000
MX+(TEMP+CORE,[CF]×64×FACTOR/100), US+CORE,[FF]×64, BUFF); 08839000
BEGIN DS=9 LIT"FACTOR = "; 08840000
SI=LOC I; DS=2 DEC; I=DI; DI=DI-2; DS=FILL; DI=I; 08841000
DS=LIT", "; DS=2 DEC; 08842000
DS=13 LIT", MAX CORE = "; DS=7 DEC; 08843000
I=DI; DI=DI-7; DS=6 FILL; DI=I; 08844000
DS= 8 LIT", USING "; DS=7 DEC; DS=LIT", "; 08845000
DI=DI-8; DS=6 FILL; 08846000
END STREAM; 08847000
EXIT: SPOUT(BUFF); 08848000

```

```

END CHANGEFACTOR;
PROCEDURE SHEETDIDDLER(BUFF,TYPE,SID); VALUE BUFF,TYPE,SID;
                                REAL  BUFF,TYPE,SID;
COMMENT TYPE = 5: PS == CHANGE PRIORITY OF JOB IN SCHEDULE
              = 7: XS == EXECUTE JOB IN SCHEDULE (FORCE SELECTION)
              = 6: ES == ELIMINATE JOB FROM SCHEDULE (FORCE SELECTION,
                        THEN "DS")
              = 4: TS == TYPE OUT SCHEDULE;
BEGIN REAL IOD,T,PRIORITY;
      INTEGER LEVEL,NEXTLINK,THISLINK,LASTLINK;
      INTEGER ES,EM,EH; DEFINE ET = EH#;
      BOOLEAN LASTPASSED,ATLEASTONE;
      ARRAY S[*],DLNK[*];
$ SET OMIT = NOT(PACKETS)
      DEFINE UNITNO = S[23],[2:6]#; % ORIGINATING UNIT
$ POP OMIT
      LABEL CONTINUE,C1,READIN,GNX,TS,TS1,TS2,
            XSES,ESLL,PS,PS1,PS2,SPIT,EXIT;

      SUBROUTINE GETNEXT; % READS IN NEXT JOB SHEET ENTRY
      BEGIN
      CONTINUE: LASTLINK ← THISLINK;
                IF (THISLINK←NEXTLINK) ≠ 0 THEN GO TO READIN;
      C1:       IF (LEVEL←LEVEL+1)>SHEETMAX THEN
                BEGIN LASTPASSED ← TRUE; GO TO GNX END;
                LASTLINK ← NEXTLINK ← 0;
                IF (THISLINK←SHEET[LEVEL],[CF]) = 0 THEN GO TO C1;
      READIN:  DISKWAIT(←S,[CF],30,THISLINK);
                NEXTLINK ← S[29];
                IF S[0],[36:6]=@14 THEN GO CONTINUE;%PASS LM ENTRY

      GNX:
      END GETNEXT;

      SLEEP([TOGGLE],SHEETMASK); LOCKTOG(SHEETMASK);
      S1=[M[GETSPACE(31,2,5)+2]]&30[8:38:10];
      LEVEL ← "1; LASTPASSED ← FALSE;
      IF TYPE LSS 0 THEN GO ESLL;%ES A JOB ON LOGICAL LINE = SID
      BUFF ← ((T←BUFF),[15:15]-1)&M[P(DUP)-1][9:9:9];
      IF TYPE=4 THEN GO TS; IF TYPE=5 THEN GO PS; GO XSES;
      TS:  ATLEASTONE ← FALSE;
      TS1: GETNEXT; IF LASTPASSED THEN GO TO TS2;
      IF SID NEQ 63 THEN BEGIN IF S[3],[8:10] NEQ SID THEN GO TS1 END ELSE
      IF ATLEASTONE THEN BUFF,[CF] := SPACE(10);
      ET←((CLOCK+P(RTR))/60)←S[23],[24:24];
      ES ← ET MOD 60; ET ← ET DIV 60; EM ← ET MOD 60; EH ← ET DIV 60;
      STREAM(C←LEVEL,J←S[*],ID←S[3],[8:10],J1←(S[0]<0) OR ((S[0] EQV
      "LIBMAIN")≠NOT 0)AND((S[1] EQV "DISK ")≠NOT 0)),
      J2←S[27],EH,EM,ES,A←S[20]×64,BUFF);
      BEGIN SI←LOC C; DS←6 DEC; DI←DI-6; DS←5 FILL; DI←BUFF; DI←DI+6;
      DS←LIT"i"; SI←J; SI←SI+1; DS←7 CHR; DS←LIT"/"; SI←SI+1;
      DS1←7CHR;J1(DS1←LIT" ";SI1←LOC J2;SI1←SI+1;DS1←7CHR);
      DS1←LIT" ";SI1←LOC ID;DS1←2 DEC;
      DS←7 LIT" IN FOR"; SI←LOC EH;
      3(DS ← LIT"i"; DS←2 DEC); ES←DI; DI←DI-9; DS←8 FILL;
      DI←ES; DS←8 LIT", NEEDS ";
      SI←LOC A; DS←5 DEC; DS←LIT" "; DI←DI-6; DS←4 FILL;
      END STREAM;
      SPOUTER(BUFF,IF SID≠63 THEN UNITNO ELSE 0,1);
      IF SID NEQ 63 THEN BEGIN TYPE:=4;GO EXIT END; % 4=TS

```

```

08848500
08849000
08850000
08850100
08850200
08850300
08850400
08850500
08850600
08851000
08852000
08852500
08853000
08854000
08854499
08854500
08854501
08855000
08856000
08857500
08858000
08859000
08860000
08860500
08861000
08862000
08863000
08864000
08865000
08867000
08868000
08869000
08870000
08871000
08880000
08881000
08882000
08882100
08882500
08883000
08884000
08885000
08885500
08886000
08886300
08886600
08887000
08887100
08887200
08888000
08889000
08890000
08890010
08891000
08892000
08893000
08899000
08900000
08901000
08901500

```



ATLEASTONE←TRUE;	08902000
GO TO TS1;	08903000
TS2: IF ATLEASTONE THEN GO TO EXIT;	08904000
IF SID NEQ 63 THEN TYPE1=4 ELSE % 4=TS	08904050
STREAM(BUFF); DS ← 15 LIT " NULL SCHEDULE←"; %	%WF 08905000
SPOUT(BUFF); GO TO EXIT;	08906000
ESLL: GETNEXT;	08907000
IF LASTPASSED THEN GO TO EXIT;	08907500
IF S[26],[40:8] NEQ SID THEN GO TO ESLL;	08908000
S[2],[1:2]=3; DISKWAIT(S,[CF],30,THISLINK); GO TO EXIT;	08908500
	08909000
	08909500
XSES: GETNEXT;	08910000
IF LASTPASSED THEN BEGIN SPOUT(BUFF); GO TO EXIT END;	08911000
IF S[3],[8:10]≠SID THEN GO TO XSES;	08912000
S[2],[1:2]=(IF TYPE=7 THEN 2 ELSE 3); % [1:2]=2(XS),=3(ES)	08913000
DISKWAIT(S,[CF],30,THISLINK);	08915000
GO TO SPIT;	08915100
	08915500
PS: STREAM(PRIORITY:T);	08916000
BEGIN SI←T;	08917000
N: IF SC="+" THEN GO TO X;	08918000
IF SC<"0" THEN BEGIN SI←SI+1; GO TO N; END; T←SI;	08919000
K: IF SC≥"0" THEN IF SC≤"9" THEN	08920000
BEGIN TALLY←TALLY+1; SI←SI+1; GO TO K END;	08921000
SI←T; DI←LOC PRIORITY; T←TALLY; DS←T OCT; GO TO Z;	08922000
X: DI←LOC PRIORITY; SKIP DB; DS←11 SET;	08923000
Z:	08924000
END STREAM;	08925000
IF (PRIORITY←P)<0 THEN BEGIN SPOUT(BUFF); GO TO EXIT END;	08926000
PS1: GETNEXT; IF LASTPASSED THEN BEGIN SPOUT(BUFF); GO TO EXIT END;	08927000
IF S[3],[8:10]≠SID THEN GO TO PS1;	08928000
% DELINK AND RELINK THIS SHEET ENTRY	08929000
DLNK←[M[GETSPACE(31,2,5)+2]]&30[8:38:10];	08930000
IF NEXTLINK = 0 THEN SHEET[LEVEL],[FF] ← LASTLINK;	08931000
IF LASTLINK = 0 THEN BEGIN SHEET[LEVEL],[CF]← NEXTLINK; GO PS2 END;	08932000
DISKWAIT(=DLNK,[CF],30,LASTLINK);	08933000
DLNK[29] ← NEXTLINK;	08934000
DISKWAIT(DLNK,[CF],30,LASTLINK);	08935000
PS2: S[2],[CF] ← IF (S[18]←PRIORITY) > 32767 THEN 32767 ELSE PRIORITY;	08936000
LEVEL ← IF PRIORITY>SHEETMAX THEN SHEETMAX ELSE PRIORITY;	08937000
IF SHEET[LEVEL],[CF] ≠ 0 THEN	08938000
BEGIN DISKWAIT(=DLNK,[CF],30,SHEET[LEVEL],[FF]);	08939000
DLNK[29] ← THISLINK;	08940000
DISKWAIT(DLNK,[CF],30,SHEET[LEVEL],[FF]);	08941000
END ELSE SHEET[LEVEL] ← THISLINK;	08944000
SHEET[LEVEL],[FF] ← THISLINK;	08944500
S[29] ← 0; S[3] ← ABS(S[3]); % TO GET SELECTION TO PRINT MESSAGE;	08945000
DISKWAIT(S,[CF],30,THISLINK);	08946000
FORGETSPACE(DLNK);	08947000
SPIT: IF BUFF≠0 THEN	08947100
% SET OMIT = NOT(PACKETS)	08947199
IF UNITNO GEQ 32 THEN	08947200
BEGIN	08947300
MOVE(9,BUFF+1,BUFF); SPOUTER(BUFF,UNITNO,64);	08947400
END ELSE	08947500
% POP OMIT	08947501
FORGETSPACE(BUFF);	08947600
	08948000
EXIT: UNLOCKTOG(SHEETMASK);	08997000

FORGETSPACE(S);	08998000
IF TYPE#4 THEN BEGIN KEYBOARDCOUNTER + KEYBOARDCOUNTER-1; % 4=TS	08998200
SELECTION;	08998400
KEYBOARDCOUNTER + KEYBOARDCOUNTER+1;	08998600
END ELSE SCHEDLOOK(0,0);%TYPE OUT SCHEDULED TASKS	08998800
END SHEETDIDDLER;	08999000
PROCEDURE LOGOUT;	09000000
BEGIN ARRAY L=LOGARRAY[*];	09001000
REAL DT,BH,RCW#+0;	09002000
INTEGER AH,DA,N;	09003000
LABEL START,EXIT;	09004000
SUBROUTINE UNLOCK;	09004100
BEGIN UNLOCKTOG(HOLDMASK);	09004200
SPOUT(AH);	09004400
FORGETSPACE(DT);	09004500
FORGETSPACE(BH);	09004600
END;	09004700
START;	09005000
DT := SPACE(30);	09006000
IF L[33]=0 THEN	09008000
BEGIN;STREAM(DT); DS+9 LIT "NULL LOG*";	09009000
SPOUT(DT);	09010000
GO TO EXIT	09011000
END;	09012000
SLEEP([TOGGLE],HOLDMASK); LOCKTOG(HOLDMASK);	09012100
DISKWAIT(-DT,-30,DIRECTORYTOP=SYSNO);	09012200
AH:=M[DT+20],[8:10];	09013000
DO	09013600
BEGIN AH:=(AH+1) MOD 1000;	09013700
STREAM(AH, N:=N);	09013800
BEGIN SI:=LOC AH; DS:=8 DEC END;	09013900
N:=N&ACTDATE[6:12:24];	09014000
END	09014100
UNTIL AH=999 OR DIRECTORYSEARCH("N,"LOG "	09014200
\$ SET OMIT = NOT(SHAREDISK)	09014299
&(SYSNO+17)[24:42:6]	09014300
\$ POP OMIT	09014301
,5)=0;	09014400
M[DT+20]:=0&AH[8:38:10];	09014500
SLEEP([L[32]],-0); L[32]+ABS(*P(DUP));	09015000
SLEEP([L[31]],IOMASK);	09016000
AH := SPACE(30);	09017000
MOVE(30,AH-1,AH);	09018000
BH := SPACE(30);	%LOG09019000
IF DA=0 THEN	09019100
IF (M[AH+10]:=PETUSERDISK("L[34],1))=0 THEN	09019200
BEGIN STREAM(N,AH);	09019300
BEGIN DS:=18 LIT"#NO USER DISK FOR ";	09019400
SI:=LOC N; SI:=SI+1; DS:=7 CHR;	09019500
DS:=5 LIT"/LOG*";	09019600
END;	09019700
L[32]:=NABS(*P(DUP));	09019750
UNLOCK;	09019800
DA:=GETUSERDISK(L[34]&2[1:46:2]);	09019900
GO TO START;	09020000
END ELSE ELSE M[AH+10]:=DA;	09020100
M[BH] := NOT 0;	09020200
DISKWAIT(BH,1,M[AH+10]);	09020300
DISKWAIT(-BH,30,L[36]);	09021000
MOVE(10,BH,AH);	09022000

```

M[BH+4]:=OR5[9:45:3];
STREAM(A:=[DATE],B:=BH+3,C:=0);
BEGIN
    SI:=A; DI:=LOC C; DS:=8 OCT; SI:=LOC C;
    SI:=SI+5; DI:=B; DI:=DI+5; DS:=3 CHR;
    END;
L[35]:=M[AH+10];
L[33]:=0;
L[32]:=NABS(*P(DUP));
ENTERUSERFILE(-N,"LOG "
$ SET OMIT = NOT(SHAREDISK)
    &(SYSNO+17)[24:42:6]
$ POP OMIT
    ,BH=1);
M[AH+7] := M[AH+8]x3 - (M[AH+9] := 1);
DISKWAIT(AH,30,L[36]);
STREAM(N,
$ SET OMIT = NOT(SHAREDISK)
    SI=SYSNO+17,
$ POP OMIT
    AH);
BEGIN DS=21LIT "**** NEW LOG FILE IS ";
    SI+LOC N; SI+SI+1; DS+7 CHR;
    DS:=6 LIT"/LOG +";
$ SET OMIT = NOT(SHAREDISK)
    DI:=DI-2;SI:=LOC S;SII:=SI+7;DS:=CHR;
$ POP OMIT
    END;
DISKWAIT(DT:=30,DIRECTORYTOP=SYSNO);
UNLOCK;
TIMEOUT(SPACE(10));
GIMEDATE(SPACE(10),1);
EXIT;
END;
PROCEDURE LOGDISK;
BEGIN REAL FID,MID,C,N; REAL RCW=+0;
    LABEL SEEK;
    C+0;
SEEK;
SEEKNAM(-1,-1,C,MID,FID,N);
IF C#0 THEN
BEGIN
    IF NOT SYSTEMFILE(MID,FID) THEN
    IF (MIDI=DIRECTORYSEARCH(MID,FID,15))#0 THEN FORGETSPACE(MID);
    GO SEEK;
END;
STREAM(N:=N:=SPACE(10));
DS=29 LIT"#DISK FILE LOGGING COMPLETED+";
SPOUT(N);
END;
PROCEDURE LINEMESSAGES(BUFH); VALUE BUFH; REAL BUFH;
BEGIN
REAL LINE,CHRS,I,BF,UZER;
LABEL ERROR,EXIT,CLEANUP;
ARRAY INF[*];
BOOLEAN COLAPSE;

REAL SUBROUTINE SHOVEITOUT;
BEGIN

```

```

09024900
09025000
09025100
09025200
09025300
09025400
09026000
09027000
09027500
09028000
09028049
09028050
09028051
09028100
%LOG09028200
09029000
09030000
09030099
09030100
09030101
09030200
09031000
09032000
09033000
09033099
09033100
09033101
09034000
09036000
09037000
09037100
09037200
09038000
09040000
09050000
09051000
09052000
09052500
09053000
09054000
09055000
09056000
09056500
09057000
09058000
09059000
09060000
09061000
09062000
09064000
09100000
09101000
09102000
09103000
09104000
09105000
09106000
09107000
09108000
09109000

```

IF P(STATABLE[LINE],DIALEDUP,DUP) THEN	09110000
BEGIN	09111000
IF COLAPSE THEN	09112000
BEGIN	09113000
COLAPSE← FALSE;	09114000
STREAM(S← BUFH; BF);	09115000
BEGIN	09116000
SI← S;	09117000
DS:=8 LIT"*=*ATTN:";	09118000
IF SC≠"*" THEN	09119000
BEGIN	09120000
IF SC=" " THEN	09121000
BEGIN	09122000
DS:=CHR;	09123000
BB: IF SC = " " THEN	09124000
BEGIN SI:=SI+1; GO BB; END;	09125000
GO TO X;	09125600
END;	09128000
IF SC = "≠" THEN	09129000
BEGIN DS:=LIT MARK; SI:=SI+1; END ELSE	09129500
DS← CHR;	09130000
GO X;	09131000
END;	09132000
DS:=3 LIT"≠*";	09133000
S← DI;	09134000
END;	09135000
CHRS← ((CHRS← P),[33:15]-BF)×8+CHRS,[30:3];	09136000
END;	09137000
TWXOUT(BF,CHRS,-1,LINE);	09138000
END;	09140000
SHOVEITOUT← P;	09141000
END;	09141500
BF← BUFH,[15:15]-1;	09142000
COLAPSE← TRUE;	09142300
STREAM(RESULT← -1,S← 0; B← BUFH; A← "ALL");	09142600
BEGIN	09143000
SI← B;	09144000
DD: IF SC=" " THEN BEGIN SI← SI+1; GO DD END;	09145000
S← SI; DI←LOC A;	09146000
IF SC<"0" THEN	09147000
BEGIN	09148000
DI← DI+5;	09149000
IF 3 SC=DC THEN	09150000
BEGIN	09151000
IF SC=ALPHA THEN GO U;	09151100
RESULT← TALLY	09151200
END	09152000
ELSE	09152100
U: BEGIN	09153000
DI← LOC RESULT; DS← 8 LIT " + ";	09154000
SI← S; DI← LOC RESULT; DI← DI+1;	09155000
7(IF SC=ALPHA THEN DS←CHR ELSE JUMP OUT);	09156000
END	09157000
END	09158000
END	09159000
ELSE	09160000
BEGIN	09161000
IF SC>9 THEN GO EX;	09162000
SI← SI+1; DI← LOC RESULT;	09163000
IF SC<0 THEN GO ONE;	09164000

	IF SC>9 THEN GO ONE;	09165000
	SI← S; DS← 2 OCT; GO SUC;	09166000
ONE:	SI← S; DS← 1 OCT;	09167000
SUC:	END;	09168000
EXI:	S← SI;	09171000
	END;	09172000
	BUFH← P;	09173000
	IF (LINE≠P)≠(-1) OR SYSDISKADR=0 THEN GO ERROR;	09174000
	IF LINE<0 THEN	09175000
	BEGIN % USERID WAS GIVEN	09176000
	INF←[M[SPACE(SYSDISKRL)]]&SYSDISKRL[8:38:10];	09177000
	UZER← ABS(LINE); I←0;	09178000
	FOR LINE:= 0 STEP 1 UNTIL STATIONMAX DO	09179000
	BEGIN	09180000
	SYSDISKIO(1,LINE,INF);	09181000
	IF (UZER EQV INF[1])≠NOT 0 THEN	09182000
	BEGIN	09183000
	I← I+SHOVEITOUT;	09184000
	END	09186000
	END;	09187000
	SYSDISKIO(1,STATIONMAX+1,INF);	09187500
	FORGETSPACE(INF);	09188000
	IF I=0 THEN	09188500
	BEGIN	09189000
	LINE← NABS(UZER);	09189300
	GO ERROR;	09189500
	END	09190000
	ELSE	09190500
	GO CLEANUP;	09191000
	END;	09192000
	IF LINE GTR STATIONMAX THEN GO ERROR;	09193000
	IF LINE>0 THEN % LINE WAS GIVEN	09194000
	IF SHOVEITOUT THEN GO CLEANUP ELSE	09195000
	BEGIN;	09196000
	STREAM(L←LINE,T←1+(LINE>9),BF);	09197000
	BEGIN	09198000
	DS← 5 LIT "LINE ";	09199000
	SI← LOC L; DS← T DEC;	09200000
	DS← 15 LIT " NOT DIALED-UP-";	09201000
	END;	09202000
	SPOUT(BF);	09203000
	GO EXIT;	09204000
	END;	09205000
	I← 0; % SS ALL	09206000
	FOR LINE := 1 STEP 1 UNTIL STATIONMAX DO	09207000
	I← I+ SHOVEITOUT;	09208000
	IF I=0 THEN	09209000
	BEGIN;	09210000
	STREAM(BF); DS← 19 LIT "NO LINES DIALED-UP-";	09211000
	SPOUT(BF);	09212000
	GO EXIT;	09213000
	END;	09214000
CLEANUP:	FORGETSPACE(BF);	09215000
	GO EXIT;	09216000
ERROR:	IF LINE>0 THEN	09217000
	STREAM(L← LINE,BF);	09218000
	BEGIN	09219000
	DS← 5 LIT "LINE ";	09220000
	SI← LOC L; DS← 2 DEC;	09221000
	DS← 15 LIT " NOT AVAILABLE-";	09222000

```

                END
            ELSE
                IF LINE<(-1) THEN
                    STREAM(L← LINE,BF);
                    BEGIN
                        SI← LOC L; SI← SI+1;
                        DS← 7 CHR;
                        DS← 15 LIT " NOT DIALED=UP←";
                    END;
                SPOUT(BF);
            EXIT;
            END;
        PROCEDURE CALLCANDE(BUFH,TYPE); VALUE BUFH,TYPE; REAL BUFH,TYPE;
        BEGIN
            REAL BUFF,A,L,F;
            BUFF← BUFH.[15:15]-1;
            IF TYPE≠0 THEN P(BUFH) ELSE
                STREAM(BUFH;TYPE);
            BEGIN
                DI← BUFH; DI← DI-3; BUFH← DI; SI← LOC T; SI← SI+6;
                DS← 2 CHR; DS← LIT " ";
            END;
            BUFH← P;
            DO BEGIN
                A← GETAREA(2);
                STREAM(R←0,BUFH;A← A INX 1);
                BEGIN
                    SI← BUFH;
                    2(56(IF SC#" " THEN DS← CHR ELSE
                        BEGIN DS← LIT " "; TALLY← 1; R← TALLY;
                        JUMP OUT 2 TO E END));
                    E; BUFH← SI;
                END;
                BUFH← P; F← P;
                IF L≠0 THEN
                    BEGIN
                        M[A]← P(DUP,LOD) & L [CTF];
                        M[L]← P(DUP,LOD) & A [CTC];
                    END;
                    L;=A;
                END UNTIL F;
                M[A],MESSEND;=1;
                GIVEAWAY(A);
                FORGETSPACE(BUFF);
            END CALLCANDE;
            % SET OMIT = NOT(AUXMEM)
            REAL PROCEDURE NEXTAUXMEMWORD(HEADER,FILEPARAM,DISKADDRESS);
            REAL DISKADDRESS,FILEPARAM; ARRAY HEADER[*];
            BEGIN
                REAL T; LABEL EXIT;
                ARRAY A[*];
                DEFINE EOFPTR = HEADER[7]#,
                ROWSIZE = HEADER[8]#,
                MAXROWS = HEADER[9],[43: 5]#,
                SEGMENTCOUNT = FILEPARAM,[18:15]#,
                WORDCOUNT = FILEPARAM,[12: 6]#,
                ROWCOUNT = FILEPARAM,[ 6: 6]#;
                A := IOQUE & FILEPARAM[CTC];
                IF DISKADDRESS = 0 THEN % FIRST CALL ON PROCEDURE
                    BEGIN

```

```

09223000
09224000
09225000
09226000
09227000
09228000
09229000
09230000
09231000
09232000
09233000
09234000
09300000
09301000
09302000
09303000
09303500
09304000
09305000
09306000
09307000
09308000
09309000
09310000
09311000
09313000
09314000
09315000
09316000
09317000
09318000
09319000
09320000
09321000
09322000
09323000
09324000
09325000
09325500
09325600
09326000
09327000
09328000
09329000
09330000
09400000
09400100
09400200
09400300
09400400
09400500
09400600
09400700
09400800
09400900
09401000
09401100
09401200
09401300
09401400

```

CALL CANDE

```

IF (DISKADDRESS := HEADER[10]) LEQ 0 OR EOFPTR LSS 0 OR
(HEADER[10] EQV @0003600036000101) NEQ (NOT 0) THEN
  BEGIN
    STREAM(INT:=FILEPARAM LSS 0, T:=T:=SPACE(4));
    BEGIN DSI:=26 LIT" IMPROPER AUXMEM MCP FILE*";
      INT(DI:=DI-9; DSI:=3 LIT"INT");
    END;
    SPOUT(T); GO TO EXIT;
  END;
DISKWAIT(-A,[CF],30,DISKADDRESS); SEGMENTCOUNT := 1;
NEXTAUXMEMWORD := NFLAG(A[0]),[CF];
GO TO EXIT;
END;
WORDCOUNT := WORDCOUNT + 1;
IF WORDCOUNT GTR 29 THEN
  BEGIN
    SEGMENTCOUNT := SEGMENTCOUNT+1;
    IF SEGMENTCOUNT GTR ROWSIZE THEN
      BEGIN
        SEGMENTCOUNT := 1;
        ROWCOUNT := ROWCOUNT + 1;
        IF ROWCOUNT GTR MAXROWS THEN GO TO EXIT;
        DISKADDRESS := HEADER[10 + ROWCOUNT];
        IF DISKADDRESS LEQ 0 THEN GO TO EXIT;
      END;
      IF (SEGMENTCOUNT+ROWCOUNT*ROWSIZE-1) GTR EOFPTR THEN GO EXIT;
      DISKWAIT(-A,[CF],30,DISKADDRESS+SEGMENTCOUNT-1);
      WORDCOUNT := 0;
    END;
    NEXTAUXMEMWORD := NFLAG(A[WORDCOUNT]),[CF];
    EXIT; END PROCEDURE NEXTAUXMEMWORD;
PROCEDURE TRANSFERMCP TOAUXMEM(HDRADRS,MAXLOC); VALUE HDRADRS,MAXLOC;
REAL HDRADRS,MAXLOC;
BEGIN
  REAL AUXADDRESS, AUXWORDS, CELLS, CELLVALUE, CODEADDRESS,
  CODESIZE, DISKADDRESS, ESPBITADDRESS, ESPBITCELL, FILEPARAM, PRTCELL,
  RSLT, A=CELLVALUE;
  ARRAY HEADER[*];
  LABEL ERROR,LOOP,STOPTRANSFER;
  HEADER := IOQUE & HDRADRS[CTC];
  FILEPARAM := SPACE(30),[CF]; CODEADDRESS:= SPACE(1024);
  ESPBITCELL := P(,ESPBIT); ESPBITADDRESS:=NFLAG(M[ESPBITCELL]),[CF];
  LOOP:
  $F (PRTCELL:=NEXTAUXMEMWORD(HEADER,FILEPARAM,DISKADDRESS)) > 0 THEN
    BEGIN
      IF PRTCELL LSS @200 OR PRTCELL GEQ MAXLOC THEN
        BEGIN
          ERROR: STREAM(PRTCELL,A:=A:=SPACE(10));
          BEGIN
            DSI:=26LIT"INVALID MCP=AUX PRT CELL,@";
            SI:=LOC PRTCELL;
            16(DSI:=3 RESET; 3(IF SB THEN DSI:=SET ELSE DSI:=RESET; SKIP SB));
            DSI=LIT"@"; DI:=DI-17; DSI:=15 FILL;
            END STREAM;
            SPOUT(A); GO TO LOOP;
          END IF BAD PRT CELL NUMBER;
        END
      IF (CELLVALUE:=NFLAG(M[PRTCELL]),[CF] NEQ ESPBITADDRESS THEN
        BEGIN % CHECK FOR PRESENT, NON-SAVE SEGMENT
          IF P(M[PRTCELL],TOP) THEN BEGIN P(DEL); GO ERROR; END ELSE
          IF NFLAG(P),[117] NEQ @165 THEN GO TO ERROR;

```

```

09401500
09401600
09401700
09401800
09401850
09401900
09401950
09402000
09402100
09402200
09402300
09402400
09402500
09402600
09402700
09402800
09402900
09403000
09403100
09403200
09403300
09403400
09403500
09403600
09403700
09403800
09403900
09404000
09404100
09404200
09404300
09404400
09404500
09404600
09404700
09404800
09404900
09405000
09405100
09405200
09405300
09405400
09405500
09405600
09405700
09405800
09405900
09406000
09406100
09406200
09406300
09406400
09406500
09406600
09406700
09406800
09406900
09407000
09407100
09407200

```

```

END; 09407300
IF CELLVALUE,[6:1] THEN GO TO LOOP; % ALREADY WRITTEN 09407400
IF PRTCELL=P(,DISKORAUERROR) OR PRTCELL=P(,REENTER) THEN GO ERROR; 09407410
CODESIZE := CELLVALUE,[8:10]; 09407500
IF (AUXADDRESS:=AUXILIARYSPACE(CODESIZE)) GTR 0 THEN %SPACE AVAILABLE 09407600
BEGIN 09407700
DISKWAIT(=CODEADDRESS, CODESIZE, CELLVALUE, [FF]+MCPBASE); 09407800
M[CODEADDRESS=1] := 0 & CODESIZE[CTF]; 09407900
DISKIO(RSLT, CODEADDRESS=1, (CODESIZE&1[3:47:1]), 09408000
      =(0&AUXADDRESS[32:36:12])); 09408002
SLEEP([RSLT], IOMASK); 09408005
IF RSLT,[26:7] NEQ 0 THEN % AUXMEM ERROR FOR MIX ZERO 09408010
BEGIN 09408015
STREAM(A:=A:=SPACE(10)); 09408020
DS:=27LIT"AUXMEM TRANSFER TERMINATED*"; 09408025
SPOUT(A); 09408030
GO TO STOPTRANSFER; 09408035
END; 09408040
M[PRTCELL] := (*P(DUP))&AUXADDRESS[CTF]&1[6:47:1]; % 6:1=AUXMEM 09408100
AUXWORDS := AUXWORDS + CODESIZE,[38:6]+1; 09408200
CELLS := CELLS + 1; 09408300
GO TO LOOP; 09408400
END % IF AUXILIARY SPACE AVAILABLE 09408500
ELSE GO TO STOPTRANSFER; 09408600
END; % IF PRTCELL GTR 0 09408700
STOPTRANSFER; 09408800
IF CELLS GTR 0 THEN 09408900
BEGIN 09409000
STREAM(CELLS, AUXWORDS:=(AUXCODE[0]:=AUXWORDS)*16, A:=A:=SPACE(10)); 09409100
BEGIN 09409200
SI:=LOC CELLS; DS:=3 DEC; A:=DI; DI:=DI+3; DS:=2 FILL; DI:=A; 09409300
DS:=37LIT" MCP SEGMENTS TRANSFERRED TO AUXMEM ("; 09409400
DS:=5DEC; DS:=8LIT" WORDS)*"; 09409500
END STREAM; 09409600
SPOUT(A); 09409700
END ; % IF CELLS GTR 0 09409800
FORGETSPACE(FILEPARAM,[CF]); FORGETSPACE(CODEADDRESS); 09409900
END PROCEDURE TRANSFERMCP TO AUXMEM; 09410000
% SET OMIT = NOT(AUXMEM OR MONITOR) 09410100
PROCEDURE SETMONITORFILE(STOP); VALUE STOP; REAL STOP; 09410200
BEGIN 09410300
DEFINE 09410400
SYSMTRFLAG = CTABLE[4],[01:01]#, % ON, IF MONITORING SYSTEM 09410500
HDRDISKADRS = CTABLE[4],[18:15]#, % DISK ADRS OF SYSTEM/MONITOR HEADER 09410600
TABLEOFFSET = CTABLE[4],[33:15]#, % AVAILABLE WORD IN SYSMTR TABLE 09410700
MAXROWADRS = CTABLE[5],[02:23]#, % MAX DISK ADDRESS FOR THIS ROW 09410800
ROWADRS = CTABLE[5],[25:23]#; % CURRENT DISK ADRS FOR THIS SEGMENT 09410900
REAL I,K,HDR,MAXROWS,ROWSIZE,ROWSTART,FILENAME,ROWCOUNT,NEWROW; 09411000
REAL CELL=I, DESC=K, SIZE=HDR, AREA=MAXROWS; 09411100
INTEGER MAXSEG; 09411200
LABEL GETDISK, FORGETHEADER, FIRSTRECORD, STOPP, EXIT; 09411300
FILENAME:="SYSTEM " 09411310
% SET OMIT = NOT SHAREDISK OR OMIT 09411319
&(SYSNO+17)[42:42:6] 09411320
% POP OMIT 09411321
; 09411330
IF STOP THEN GO TO STOPP; % STOP MONITORING AND RELEASE FILE 09411400
SIZE:=(DESC:=NFLAG(M[CELL:=P(,ENTERSYSMTR)])),[8:10]; 09411500
AREA:=GETSPACE(SIZE,65,1)+2; % GET SAVE CORE AREA 09411600
% SET OMIT = NOT(AUXMEM) OR OMIT 09411700

```



```

IF DESC.[6:1] THEN % STORED ON AUXMEM
  DISKWAIT(-AREA,SIZE,=(0&DESC[32:21:12]))
ELSE
  $ POP OMIT
  DISKWAIT(-AREA,SIZE,DESC,[FF]+MCPBASE); % GET CODE SEGMENT
  M[AREA=1]:=CELL&SIZE[CTF]; M[CELL]:=FLAG(DESC)&AREA[CTC];
  AREA:=GETSPACE(64,0,1)+2; % SAVE CORE AREA FOR SYMTR
  CTABLE[6]:=(*P(DUP))&(AREA+1)[CTF]&(AREA+33)[CTC];
  SYMTR:=CTABLE[6],[FF]; CTABLE[6],[1:4]:=8; % USE LOWER CORE AREA
  M[CTABLE[6],[FF] INX 30]:=M[CTABLE[6],[CF] INX 30]:=NOT 0; % MARKERS
  IF (HDR:=DIRECTORYSEARCH(FILENAME,"MONITOR",17))=0 THEN % NOT THERE
    BEGIN % BUILD HEADER
      HDR:=SPACE(30); M[HDR]:=0; MOVE(29,HDR,HDR+1);
      M[HDR]:=00003600036000101; % 30 WORD UNBLOCKED RECORDS
      STREAM(HEADER,H3:=HDR+3);
      BEGIN
        SI:=LOC DATE; DS:=8OCT; % CREATION DATE
        DI:=H3; DS:=2LIT"+#"; % NEW FORMAT, SAVE FACTOR=10
        S1:=H3; SI:=SI+5; DS:=3CHR; % ACCESS DATE
      END STREAM;
      M[HDR+4],[9:1]:=1; % MARK AS DATA FILE
      M[HDR+7]:=299; M[HDR+8]:=300; M[HDR+9]:=20; % [20:300]
      IF (M[HDR+10]:=PETUSERDISK(300&1[2:47:1],1))=0 THEN % NO DISK
        BEGIN
          STREAM(FILENAME,HDR);
          BEGIN SI:=LOC FILENAME; SI:=SI+1;
            DS:=18 LIT "NO DISK SPACE FOR ";
            DS:=7 CHR; DS:=14 LIT "/MONITOR FILE+";
          END STREAM;
          SPOUT(HDR); GO TO EXIT;
        END; % IF NO DISK
        HDRDISKADRS:=EUF(FILENAME,"MONITOR",HDR-1);
        ROWADRS:=M[HDR+10]; MAXROWADRS:=M[HDR+10]+M[HDR+8]-1;
        FORGETSPACE(DIRECTORYSEARCH(FILENAME,"MONITOR",17));
        STREAM(FILENAME,KI:=K:=SPACE(4));
        BEGIN SI:=LOC FILENAME; SI:=SI+1; DS:=LIT ",";
          DS:=7 CHR; DS:=22 LIT "/MONITOR FILE CREATED+";
        END STREAM;
        SPOUT(K);
      FIRSTRECORD;
      M[SYMTR]:=NOT 0; % END OF FILE MARKER
      TABLEOFFSET:=M[SYMTR+1]:=0; MOVE(28,SYMTR+1,SYMTR+2);
      DISKWAIT(SYMTR,30,ROWADRS); % WRITE FIRST RECORD
      SYMTRFLAG:=1; GO TO FORGETHEADER;
      END; % IF SYSTEM/MONITOR WAS NOT PRESENT
      IF (M[HDR INX 0] EQV 00003600036000101) NEQ (NOT 0) THEN
        BEGIN % SHOULD BE 30 WDS/RECORD,UNBLOCKED
          STREAM(FILENAME,HDR:=HDR,[CF]);
          BEGIN SI:=LOC FILENAME; SI:=SI+1; DS:=10 LIT ",IMPROPER ";
            DS:=7 CHR; DS:=14 LIT "/MONITOR FILE+";
          END STREAM;
          SPOUT(HDR,[CF]); GO TO STOPP;
        END; % IF IMPROPER SYSTEM/MONITOR FILE
        HDRDISKADRS:=HDR,[18:15]; ROWSIZE:=M[HDR INX 8];
        MAXROWS := M[HDR INX 9],[43:5];
        I:=MAXROWS+9; WHILE M[HDR INX I]=0 AND I GTR 9 DO I := I - 1;
        IF I LSS 10 THEN % NO DISK ADDRESS
          BEGIN
            I:=10;
          GETDISK;

```

```

09411710
09411720
09411730
09411740
09411750
09411800
09411900
09412000
09412100
09412200
09412300
09412400
09412500
09412600
09412700
09412800
09412900
09413000
09413100
09413200
09413300
09413400
09413500
09413600
09413700
09413800
09413900
09413950
09414000
09414100
09414200
09414400
09414500
09414600
09414700
09414800
09414900
09415000
09415100
09415200
09415300
09415400
09415500
09415600
09415700
09415800
09415900
09416000
09416100
09416200
09416300
09416400
09416500
09416600
09416700
09416800
09416900
09417000
09417100
09417200

```

ROWSTART:=M[HDR INX I]:=PETUSERDISK(ROWSIZE&1[2:47:1],1);	09417300
IF ROWSTART = 0 THEN % NO DISK	09417400
BEGIN	09417500
STREAM(FILENAME,HDR:=HDR:=HDR,[CF]);	09417600
BEGIN SI:=LOC FILENAME; SI:=SI+1;	09417700
DS:=18 LIT "NO DISK SPACE FOR ";	09417800
DS:=7 CHR; DS:=14 LIT "/MONITOR FILE*";	09417850
END STREAM;	09417900
SPOUT(HDR); GO TO STOPP;	09418000
END; % IF NO DISK	09418100
NEWROW:=TRUE; % DONT SEARCH FILE IF NEW DISK ROW	09418200
END ELSE ROWSTART:=M[HDR INX I];	09418300
STREAM(FILENAME,K:=K:=SPACE(4));	09418400
BEGIN SI:=LOC FILENAME; SI:=SI+1; DS:= LIT ",";	09418500
DS:=7 CHR; DS:=20 LIT "/MONITOR FILE FOUND*";	09418600
END STREAM;	09418700
SPOUT(K);	09418800
ROWCOUNT:=I-9; ROWADRS:=ROWSTART; MAXROWADRS:=ROWSTART+ROWSIZE-1;	09418900
MAXSEG:=M[HDR INX 7] MOD (ROWCOUNT*ROWSIZE); % ACTUAL EOF POINTER	09419000
M[HDR INX 7] := ROWCOUNT*ROWSIZE -1; % ADJUST EOF POINTER	09419100
DISKWAIT(HDR,[CF],30,HDRDISKADRS); % REPLACE HEADER	09419200
IF NEWROW THEN GO TO FIRSTRECORD; % DONT SEARCH DISK ROW	09419300
I:= -1;	09419400
DO BEGIN % SEARCH FOR EOF MARKER	09419500
DISKWAIT(-SYSMTR,30,ROWSTART+(I:=I+1));	09419600
IF M[SYSMTR] = NOT 0 THEN % MARKER = EOF	09419700
BEGIN	09419800
ROWADRS:=ROWSTART+I;	09419900
IF ROWADRS = MAXROWADRS THEN % FULL ROW	09420000
BEGIN	09420100
IF ROWCOUNT = MAXROWS THEN % FULL FILE	09420200
BEGIN	09420300
STREAM(FILENAME,HDR:=HDR:=HDR,[CF]);	09420400
BEGIN SI:=LOC FILENAME; SI:=SI+1; DS:=LIT ",";	09420500
DS:=7 CHR; DS:=19 LIT "/MONITOR FILE FULL*"	09420600
END STREAM;	09420700
SPOUT(HDR);	09420800
GO TO STOPP;	09420900
END; % IF FULL FILE	09421000
M[SYSMTR]:=0&62[3:42:6]; % RESTART MARKER	09421100
M[SYSMTR INX 1]:=0&63[3:42:6]; % END OF ROW MARKER	09421200
M[SYSMTR INX 2]:=0; MOVE(27,SYSMTR+2,SYSMTR+3);	09421300
DISKWAIT(SYSMTR,30,ROWADRS); % WRITE LAST RECORD IN ROW	09421400
I:=ROWCOUNT + 10; GO GETDISK;	09421500
END; % IF FULL ROW	09421600
M[SYSMTR]:=0&62[3:42:6]; % RESTART MARKER	09421700
M[SYSMTR+1]:=0; MOVE(28,SYSMTR+1,SYSMTR+2);	09421800
TABLEOFFSET:=1; SYSMTRFLAG:=TRUE;	09421900
GO TO FORGETHEADER;	09422000
END; % IF MARKER WAS FOUND;	09422100
END UNTIL I=ROWSIZE-1 OR I GEQ MAXSEG;	09422200
ROWADRS := ROWSTART; % START AT BEGINNING OF ROW IF NO MARKER	09422300
SYSMTRFLAG:=TRUE; CTABLE[7]:=0; % IO FINISH RESLT DESCH	09422400
FORGETHEADER; FORGETSPACE(HDR INX 0); GO TO EXIT;	09422500
STOPP; SYSMTRFLAG:=0;	09422600
DESC:=NFLAG(M[CELL:=P(,ENTERSYSMTR)]);	09422700
FORGETSPACE(DESC INX 0);	09422800
M[CELL]:=FLAG(DESC&(*P(,ESPBIT))[CTC]);	09422900
FORGETSPACE(CTABLE[6],[FF]=1); CTABLE[6]:=0;	09423000
%F (K:=DIRECTORYSEARCH(FILENAME,"MONITOR",16),[CF])=0 THEN K:=SPACE(4);	09423100

STREAM(FILENAME,K);	09423200
BEGIN S1:=LOC FILENAME; S1:=S1+1; DS:=LIT " ,";	09423300
DS:=7 CHR; DS:=16 LIT "/MONITOR RE-SET";	09423400
END STREAM;	09423500
SPOUT(K);	09423600
EXIT;	09423700
END PROCEDURE SETMONITORFILE;	09423800
PROCEDURE GETMONITORROW;	09423900
% CTABLE[4],[01:01]=MONITOR FLAG (ON, IF MONITORING SYSTEM)	09424000
% CTABLE[4],[18:15]=DISK ADDRESS OF SYSTEM/MONITOR FILE HEADER	09424100
% CTABLE[5],[02:23]=MAXIMUM DISK ADDRESS FOR CURRENT FILE ROW	09424200
% CTABLE[5],[25:23]=DISK ADDRESS OF CURRENT SEGMENT IN FILE	09424300
% CTABLE[6],[01:01]=FLAG FOR LOWER CORE AREA (SYSMTR) IN USE	09424400
% CTABLE[6],[02:01]=FLAG FOR UPPER CORE AREA (SYSMTR) IN USE	09424500
% CTABLE[6],[03:01]=FLAG FOR DISK IO IN PROGRESS	09424600
% CTABLE[6],[04:01]=FLAG FOR MONITOR FILE ROW FULL	09424700
% CTABLE[6],[18:15]=ADDRESS OF LOWER CORE AREA FOR SYSMTR	09424800
% CTABLE[6],[33:15]=ADDRESS OF UPPER CORE AREA FOR SYSMTR	09424900
% CTABLE[7]      =RESULT DESCRIPTOR FROM DISK IO	09425000
BEGIN	09425100
REAL RCW:=+0;	09425200
REAL I,HDR,MAXROWS;	09425300
LABEL STOPIT,EXIT;	09425400
HDR:=SPACE(30);	09425500
M[HDR]:=0&63[3:42:6]; % END OF FILE ROW MARKER	09425600
M[HDR+1]:=0; MOVE(28,HDR+1,HDR+2); % ZERO REMAINDER OF RECORD	09425700
DISKWAIT(HDR,30,CTABLE[5],[25:23]); % WRITE LAST SEGMENT	09425800
DISKWAIT(-HDR,30,CTABLE[4],[18:15]); % GET MONITOR FILE HEADER	09425900
% SEARCH HEADER TO FIND NEXT AVAILABLE SLOT FOR NEW ROW	09426000
I:=(MAXROWS:=M[HDR INX 9],[43:5])+9; % INDEX TO LAST SLOT	09426100
WHILE M[HDR INX I]=0 AND I GTR 9 DO I:=I-1;	09426200
IF I=MAXROWS+9 THEN % FILE IS FULL	09426300
BEGIN	09426400
STREAM(S:="SYSTEM "	09426500
\$ SET OMIT = NOT SHAREDISK OR OMIT	09426509
&(SYSNO+17)[42:42:6]	09426510
\$ POP OMIT	09426511
, I:=I:=SPACE(4));	09426550
BEGIN S1:=S; S1:=S1+1; DS:= LIT " ,";	09426600
DS:=7 CHR; DS:=19 LIT "/MONITOR FILE FULL";	09426700
END STREAM;	09426800
GO TO STOPIT;	09426900
END; % IF FILE IS FULL	09427000
I:=I+1; % INDEX TO NEXT AVAILABLE ROW SLOT IN FILE HEADER	09427100
IF (M[HDR INX I]=PETUSERDISK(M[HDR INX 8]&1[2:47:1],1))=0 THEN	09427200
BEGIN % NO DISK AVAILABLE	09427300
STREAM(S:="SYSTEM "	09427400
\$ SET OMIT = NOT SHAREDISK OR OMIT	09427409
&(SYSNO+17)[42:42:6]	09427410
\$ POP OMIT	09427411
, I:=I:=SPACE(4));	09427420
BEGIN S1:=LOC S; S1:=S1+1; DS:=18 LIT "NO DISK SPACE FOR ";	09427500
DS:=7 CHR; DS:=13 LIT "/MONITOR ROW";	09427600
END STREAM;	09427700
STOPIT: SPOUT(I);	09427800
SETMONITORFILE(1);	09427900
GO TO EXIT;	09428000
END; % IF NO DISK OR FULL FILE	09428100
CTABLE[5],[02:23]:=M[HDR INX I]+M[HDR INX 8]-1; % MAX ROW ADRS	09428200
CTABLE[5],[25:23]:=M[HDR INX I]; % STARTING ROW ADDRESS	09428300

```

M[HDR INX 7]:=(I-9)*M[HDR INX 8]-1; % ADJUST EOF POINTER
DISKWAIT(HDR,30,CTABLE[4],[18:15]); % REPLACE HEADER
M[HDR]:=NOT 0; DISKWAIT(HDR,30,CTABLE[5],[25:23]); % EOF MARK
EXIT: CTABLE[6],[4:1]=0; FORGETSPACE(HDR); KILL([RCW] INX NOT 2);
END PROCEDURE GETMONITOROW;
PROCEDURE ENTERSYSMTR(N); VALUE N; REAL N;
% CTABLE[4],[01:01]=MONITOR FLAG (ON, IF MONITORING SYSTEM)
% CTABLE[6],[01:01]=FLAG FOR LOWER CORE AREA IN USE (SYSMTR)
% CTABLE[6],[02:01]=FLAG FOR UPPER CORE AREA IN USE (SYSMTR);
% CTABLE[6],[03:01]=FLAG FOR DISK IO IN PROGRESS
% CTABLE[6],[04:01]=FLAG FOR MONITOR FILE ROW FULL
% CTABLE[6],[18:15]=ADDRESS OF LOWER CORE AREA FOR SYSMTR
% CTABLE[6],[33:15]=ADDRESS OF UPPER CORE AREA FOR SYSMTR
% CTABLE[7] =RESULT DESCRIPTOR FOR DISK IO FINISH
BEGIN
DEFINE
TABLEOFFSET=CTABLE[4],[33:15];%AVAILABLE WORD IN SYSMTR TABLE
MAXROWADRS =CTABLE[5],[02:23];%MAX DISK ADDRESS FOR THIS ROW
ROWADRS =CTABLE[5],[25:23];%CURRENT DISK ADDRESS FOR THIS ROW
REAL I,SYSADRS;
LABEL EXIT;
IF NOT (CTABLE[4],[01:01]) THEN GO TO EXIT; % MONITOR FLAG IS OFF
IF CTABLE[6],[3:1] THEN % DISK IO WAS IN PROGRESS
IF CTABLE[7],[19:1] THEN % IO COMPLETED
CTABLE[6],[3:1]=CTABLE[7];=0; % RESET FLAG AND ZERO DESC.
IF TABLEOFFSET GTR 28 THEN % TABLE IS FULL
IF CTABLE[6],[3:2] NEQ 0 THEN GO EXIT; % CANT SAVE THIS ONE
M[SYSMTR INX TABLEOFFSET]:=N; % STORE VALUE IN TABLE
TABLEOFFSET:=TABLEOFFSET+1;
IF TABLEOFFSET GTR 29 THEN % FULL TABLE
BEGIN
SYSADRS:=SYSMTR; % SAVE ADDRESS FOR DISK IO, SWAP CORE AREAS
IF CTABLE[6],[1:1] THEN % USING LOWER CORE AREA
BEGIN
SYSMTR:=CTABLE[6],[33:15]; % USE UPPER CORE AREA NEXT
CTABLE[6],[1:3]=3; % MARK DISK IO IN PROGRESS
END
ELSE
BEGIN % USING UPPER CORE AREA
SYSMTR:=CTABLE[6],[18:15]; % USE LOWER CORE AREA
CTABLE[6],[1:3]=5; % MARK DISK IO IN PROGRESS
END;
TABLEOFFSET:=M[SYSMTR];=0; MOVE(29,SYSMTR,SYSMTR+1);
DISKIO(CTABLE[7],[SYSADRS-1],31,ROWADRS); % WRITE OUT SEGMENT
ROWADRS:=ROWADRS+1;
IF ROWADRS=MAXROWADRS THEN % FULL ROW
BEGIN
CTABLE[6],[4:1]=1; % FLAG FOR OBTAINING NEW ROW
FORK(P(,GETMONITOROW),0,0,128,1);
END;
END; % IF FULL TABLE
EXIT: END PROCEDURE ENTERSYSMTR;
% POP OMIT
PROCEDURE CHANGEAUXFILES(BUFF,WA);
VALUE BUFF,WA; REAL BUFF,WA;
BEGIN
LABEL WADO,CAFINI,WAFINI,INTCK,NULLIT,ERROR1,ERROR2;
REAL A,I,J,K;
REAL X
% SET OMIT = SHAREDISK OR OMIT

```

```

09428400
09428500
09428600
09428700
09428800
09428900
09429000
09429100
09429200
09429300
09429400
09429500
09429600
09429700
09429800
09429900
09430000
09430100
09430200
09430300
09430400
09430500
09430600
09430700
09430800
09430900
09431000
09431100
09431200
09431300
09431400
09431500
09431600
09431700
09431800
09431900
09432000
09432100
09432200
09432300
09432400
09432500
09432600
09432700
09432800
09432900
09433000
09433100
09433200
09433300
09433400
09433500
09433510
09433600
09433700
09433800
09433900
09434000
09434020
09434030

```

```

      = J
$ POP OMIT
      ;
      BOOLEAN SUBROUTINE FILEOK;
      BEGIN P(1);
        IF P(DIRECTORYSEARCH(NABS(M[I+(K*2)]),M[I+(K*2)+1],5))=0 THEN
          BEGIN
            P(DEL,0);
            STREAM(A:=[M[I+(K*2)]]1,K,B:=K:=SPACE(5));
            BEGIN
              SI:=A; DS:=14 LIT" NO AUXMEM INT";
              K(DI:=DI-3; DS:=3 LIT"MCP"); DS:=6 LIT" FILE";
              2(SI:=SI+1; DS:=7 CHR; DS:=LIT"/"); DI:=DI-1;
              DS:=LIT" ";
            END;
            SPOUT(K);
          END;
          FILEOK:=P;
        END;
        IF WA THEN GO WADO;
        STREAM(AUXINT:=0,AUXMCP:=0,NULL:=0,OK:=0,ERRTOG:=0;
          A1:=@3145634423474642,A2:="NULL ",B:=0,
          CX:=0,BUFF,DI:=0,E:=0,F:=0,I:=I:=SPACE(4));
        BEGIN
          4(DS:=LIT"0"; DS:=7 LIT" "); SI:=BUFF; DI:=CI; GO TO L1;
        L0: IF SC=" " THEN BEGIN SI:=SI+1; GO L0 END; CI:=CX;
        L1: CX:=CI; GO TO L0; TALLY:=1;
          DI:=LOC A1; DI:=DI+6;
          IF 2 SC=DC THEN % FILES OK
            BEGIN OK:=TALLY; GO TO E1 END;
          DI:=DI+1; SI:=SI-2;
          IF 4 SC=DC THEN % NULL FILES
            BEGIN NULL:=TALLY; GO TO E1 END;
          E1:=CI; SI:=SI-4;
        L2: CX:=CI; GO TO L0;
          DI:=LOC A1; TALLY:=1;
          IF 3 SC=DC THEN
            BEGIN
              AUXINT:=TALLY; TALLY:=0;
              DI:=DI+3; GO TO L3;
            END; SI:=SI-3;
          IF 3 SC=DC THEN
            BEGIN
              AUXMCP:=TALLY; GO TO L3;
            END; GO TO E2;
        E1: GO TO E3;
        L3: BI:=TALLY;
          CX:=CI; GO TO L0;
          IF 2 SC=DC THEN
            BEGIN
              TALLY:=0; F:=TALLY;
        NL: DI:=LOC AUXINT; B(DI:=DI+8); SKIP DB;
              F(SKIP DB); DS:=SET; CX:=CI; GO TO L0;
              IF SC="," THEN BEGIN SI:=SI+1; GO TO L2 END ELSE
              IF SC="+" THEN GO TO E3 ELSE GO TO E4;
        E2: GO TO E4;
        E3: GO TO E5;
          END; SI:=SI-2; DI:=DI+1;
          IF 4SC=DC THEN
            BEGIN TALLY:=1; F:=TALLY; GO NL END;

```

```

09434040
09434050
09434060
09434100
09434200
09434300
09434400
09434500
09434600
09434700
09434800
09434900
09435000
09435100
09435200
09435300
09435400
09435500
09435600
09435700
09435800
09435900
09436000
09436100
09436200
09436300
09436400
09436500
09436600
09436700
09436800
09436900
09437000
09437100
09437200
09437300
09437400
09437500
09437600
09437700
09437800
09437900
09438000
09438100
09438200
09438300
09438400
09438500
09438600
09438700
09438800
09438900
09439000
09439100
09439200
09439300
09439400
09439500
09439600
09439700

```

SI:=SI-4; IF SC#="" THEN GO TO E2;	09439800
SI:=SI+1; CX:=CI; CI:=D; DI:=I;	09439900
B(2(DI:=DI+8)); DI:=DI+1; B:=DI; IF SC=ALPHA THEN	09440000
B(IF SC="/" THEN ELSE IF SC=" " THEN; IF TOGGLE THEN	09440100
JUMP OUT TO L4; IF SC=ALPHA THEN DS:=CHR ELSE JUMP	09440200
OUT);	09440300
E4: GO TO ERR;	09440400
E5: GO TO EXT;	09440500
L4: CX:=CI; CI:=D; IF SC="/" THEN GO TO ERR;	09440600
SI:=SI+1; CX:=CI; CI:=D;	09440700
DI:=B; DI:=DI+8; IF SC=ALPHA THEN	09440800
7(IF SC=ALPHA THEN BEGIN DS:=CHR; GO L5 END; IF SC=","	09440900
THEN ELSE IF SC=" " THEN ELSE IF SC="+" THEN; IF TOGGLE	09441000
THEN JUMP OUT ELSE JUMP OUT TO ERR; L5:); CX:=CI; CI:=D;	09441100
IF SC="+" THEN GO TO EXT; IF SC="," THEN BEGIN SI:=SI+1;	09441200
CI:=E END;	09441300
ERR: TALLY:=1; ERRTOG:=TALLY;	09441400
EXT: END;	09441500
IF P THEN BEGIN FORGETSPACE(I); GO TO ERROR2 END;	09441600
WADO:	09441650
DISKWAIT(=(X:=SPACE(30)),30	09441700
\$ SET OMIT = NOT SHAREDISK OR OMIT	09441720
&(NOT WA)[1:47:1]	09441725
\$ POP OMIT	09441730
>AUXMEMDSK);	09441750
\$ SET OMIT = NOT SHAREDISK OR OMIT	09441790
J:=4*SYSNO+X;	09441800
\$ POP OMIT	09441810
IF WA THEN	09441900
BEGIN A:=(M[X+16+SYSNO] EQV "AUXMEM ") # NOT 0;	09441910
GO TO WAFINI;	09441920
END;	09441930
IF P THEN % OK ALL FILES	09442000
BEGIN	09442100
M[J]:=ABS(*P(DUP));	09442200
GO CAFINI;	09442300
END;	09442400
IF P THEN GO NULLIT; % NULL ALL FILES	09442500
IF P(DUP)#0 THEN % MCP FILE UPDATE	09442600
BEGIN	09442700
IF P(DUP)<0 THEN	09442800
BEGIN	09442900
M[J]:=ABS(*P(DUP)); P(DEL);	09443000
GO TO INTCK;	09443100
END;	09443200
IF P,[2:1] THEN % MAKE MCP FILE NULL	09443300
BEGIN M[J]:=M[J+1]:=0; GO TO CAFINI; END;	09443400
K:=1; IF FILEOK THEN	09443500
BEGIN	09443600
M[J]:=M[I+2];	09443700
M[J+1]:=M[I+3];	09443800
END ELSE	09443900
IF CTABLE[4],[2:1] THEN GO TO ERROR1;	09444000
END ELSE IF CTABLE[4],[2:1] THEN	09444100
GO TO ERROR1 ELSE P(DEL);	09444200
INTCK:	09444300
IF P(DUP)#0 THEN % INT FILE UPDATE	09444400
BEGIN	09444500
IF P(DUP)<0 THEN GO TO CAFINI;	09444600
IF P,[2:1] THEN % MAKE INT FILE NULL	09444700

BEGIN M[J+2]:=M[J+3]:=0; GO TO CAFINI END;	09444800
K:=0; IF FILEOK THEN	09444900
BEGIN	09445000
M[J+2]:=M[I];	09445100
M[J+3]:=M[I+1];	09445200
END ELSE	09445300
IF CTABLE[4],[3:1] THEN GO TO ERROR1;	09445400
END ELSE IF CTABLE[4],[3:1] THEN	09445500
GO TO ERROR1 ELSE P(DEL);	09445600
CAFINI:	09445700
IF M[X+16+SYSNO] ≠ "AUXMEM " THEN	09445800
BEGIN	09445900
NULLIT:  IF (M[ I ] EQV "      ")=NOT 0 THEN M[J+2]:=M[J+3]:=0;	09446000
IF (M[I+2] EQV "      ")=NOT 0 THEN M[J]:=M[J+1]:=0;	09446100
M[X+16+SYSNO]:="AUXMEM ";	09446200
END;	09446300
FORGETSPACE(I);	09446400
DISKWAIT(X,-30,AUXMEMDSK);	09446500
CTABLE[4]:=(*P(DUP)) & P(DUP)[4:2:1] & 0[2:2:2];	09446550
WAFINI:	09446600
STREAM(A, NULLMCP:=M[J]=0, NULLINT:=M[J+2]=0, FILEIDS:=[M[J]],	09446700
T:=BUFF,[15:15]=1);	09446800
BEGIN DS:=17 LIT" AUXMEM FILES ARE";	09446900
A(DS:=14 LIT" NOT SPECIFIED"; JUMP OUT TO LA);	09447000
SI:=FILEIDS; SI:=SI+1; DS:=6 LIT"-MCP: ";	09447100
A:=DI; DS:=7 CHR; DS:=LIT"/"; SI:=SI+1; DS:=7 CHR;	09447200
NULLMCP(DI:=A; DS:=4 LIT"NULL"); DS:=7LIT", INT: ";	09447300
A:=DI; SI:=SI+1; DS:=7 CHR; DS:=LIT"/"; SI:=SI+1;	09447400
DS:=7 CHR; NULLINT(DI:=A; DS:=4 LIT"NULL");	09447500
LA:  DS:=LIT"+";	09447550
END;	09447600
ERROR1:	09447700
FORGETSPACE(X);	09447800
ERROR2:	09447900
END CHANGING AUXMEM FILES;	09448000
\$ POP OMIT	09448001
PROCEDURE WHATINTRNSIC(B); VALUE B; REAL B; FORWARD;	09491000
PROCEDURE INTRINSICTABLEBUILDER(FH); VALUE FH; REAL FH;	09500000
BEGIN	09500500
REAL DISKADDR:=+1, T:=+2, INTLOC:=+3, T17SIZE:=+4, MAXINT:=+5;	09501000
\$ SET OMIT = NOT(AUXMEM)	09501500
REAL HDR=MAXINT+1, FILEPARAM=HDR+1, CODE=FILEPARAM+1,	09502000
INTNUM=CODE+1, AUXADDR=INTNUM+1, AUXWORDS=AUXADDR+1,	09502500
INTCOUNT=AUXWORDS+1, A=DISKADDR, CODESIZE=T,	09503000
MFID=INTCOUNT+1, FID=MFID+1, RSLT=FID+1;	09503100
ARRAY HEADER=T17SIZE[*];	09503500
LABEL LOOP,AUXTRAN,ENDLOOP;	09504000
P(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);	09504500
\$ POP OMIT	09504501
P(0, 0, 0, 0, 0);	09504570
IF (T:=FH,[FF])=0 THEN T:=SPACE(30);	09505000
DISKWAIT(-T, 30, DISKADDR:=M[FH INX 10]);	09505500
MAXINT := M[T] + 1; % NUMBER OF INTRINSICS + 1	09505600
T17SIZE := M[T INX 17],[8:10]+1; % INTR.#17 SIZE+1WD.FOR DISK.ADDR.	09505700
FORGETSPACE(T);	09505750
INTRNSC:=[M[INTLOC:=GETSPACE(MAXINT+T17SIZE,0,1)+2]]&	09506000
(MAXINT+T17SIZE)[8:38:10]; % SPACE FOR INTRINSIC TABLE + INT.#17	09506100
DISKWAIT(-(INTRNSC INX 0),MAXINT,DISKADDR);	09506500
M[INTRNSC INX NOT 0] := 0; MAXINT := MAXINT -1;	09507000
FOR T := 1 STEP 1 UNTIL MAXINT DO	09507500

```

INTRNSC[T]:=NABS((P(*P(DUP),DUP).[8:10]+INTSIZE) &
(P(XCH) INX 0 + DISKADDR)[6:21:27]);
DISKWAIT(=(INTLOC:=INTLOC+MAXINT+2),(T17SIZE=1),INTRNSC[17],[6:27]);
INTRNSC[17] := (*P(DUP))&INTLOC[CTC]; % MARK PRESENT
M[INTLOC-1]:=0&(T17SIZE=1)[CTF]; % DUMMY MARKER FOR DUMP/ANALYZE
DISKADDR:=0&1[4:47:1];
INTRNSC[2]:=*P(DUP) OR DISKADDR;
FOR T:=18 STEP 1 UNTIL 20 DO INTRNSC[T]:=*P(DUP) OR DISKADDR;
$ SET OMIT = NOT(AUXMEM)
CTABLE[9] := M[FH INX 10];
IF (P(RRR).[30:1] AND USED RB) OR (P(RRR).[31:1] AND USED RA) THEN
BEGIN
DISKWAIT(=(T:=SPACE(30)),30,AUXMEMDSK);
IF M[T+SYSNO+16]="AUXMEM " THEN
BEGIN
MFID:=M[T+(SYSNO*4)+2]; FID:=M[T+(SYSNO*4)+3];
FORGETSPACE(T);
IF MFID#0 THEN
IF (HDR:=DIRECTORYSEARCH(MFID,FID,4))#0 THEN
BEGIN
HEADER := IOQUE & HDR[CTC];
FILEPARAM := "SPACE(30); % SIGN BIT INDICATES INTRINSICS
CODE := SPACE(1024);
INTCOUNT := DISKADDR := 0;
LOOP: IF (INTNUM:=NEXTAUXMEMWORD(HEADER,FILEPARAM,DISKADDR))>0 THEN
BEGIN
IF INTNUM GTR MAXINT OR INTNUM LSS 1 THEN
BEGIN
STREAM(INTNUM,T:=T:=SPACE(10));
BEGIN
DS:=23LIT"INVALID INT=AUX NUMBER,";
SI:=LOC INTNUM; DS:=8DEC; DS:=LIT"+";
DI:=DI-9; DS:=7FILL;
END STREAM;
SPOUT(T); GO TO LOOP;
END; % IF BAD INTRINSIC NUMBER
IF INTRNSC[INTNUM].[3:1] OR INTNUM=17 THEN GO TO LOOP;
CODESIZE := INTRNSC[INTNUM].[CF];
IF (AUXADDR:=AUXILIARYSPACE(CODESIZE)) GTR 0 THEN
BEGIN
DISKWAIT(=CODE,CODESIZE,INTRNSC[INTNUM],[6:27]);
M[CODE-1]:=0&CODESIZE[CTF];
DISKIO(RSLT,CODE-1,(CODESIZE&1[3:47:1]),
=(0&AUXADDR[32:36:12]));
SLEEP([RSLT],IOMASK);
IF RSLT.[26:7] NEQ 0 THEN % AUXMEM ERROR FOR MIX ZERO
BEGIN
STREAM(T:=T:=SPACE(10));
DS:=27LIT"AUXMEM TRANSFER TERMINATED+";
SPOUT(T);
GO TO ENDLOOP;
END;
INTRNSC[INTNUM] := % [3:1] INDICATES AUXMEM
(*P(DUP))&0[5:35:13]&CODESIZE[CTC]&AUXADDR[CTF]&1[3:47:1];
AUXWORDS:=AUXWORDS+CODESIZE,[38:6]+1; INTCOUNT:=INTCOUNT+1;
GO TO LOOP;
END; % IF SPACE AVAILABLE
END; % IF ANOTHER INTRINSIC SPECIFIED
ENDLOOP:
IF INTCOUNT GTR 0 THEN

```

```

09508000
09508500
09508600
09508700
09508800
09509000
09509500
09510000
09510500
09510505
09511000
09511100
09511200
09511300
09511400
09511500
09511600
09511650
09511700
09512000
09512500
09513000
09513500
09514000
09514500
09515000
09515500
09516000
09516500
09517000
09517500
09518000
09518500
09519000
09519500
09520000
09520500
09521000
09521500
09522000
09522500
09523000
09523500
09523502
09523505
09523510
09523515
09523520
09523525
09523530
09523535
09523540
09524000
09524500
09525000
09525500
09526000
09526500
09526510
09527000

```



```

BEGIN
STREAM(INTCOUNT, WORDS:=AUXWORDS*16, A:=A+SPACE(10));
  BEGIN
    SII=LOC INTCOUNT; DSI=8DEC; DI:=DI+8; DSI=7FILL;
    SII=A; DI=A; 8(IF SC=" " THEN SII:=SII+1 ELSE DSI=CHR);
    DSI=29LIT" INTRINSICS MOVED TO AUXMEM (";
    SII=LOC WORDS; DSI=5DEC; DSI:=8LIT" WORDS)+";
  END STREAM;
  SPOUT(A); AUXCODE[0]:=(+P(DUP))+AUXWORDS;
  END; % IF INTRINSICS MOVED TO AUXMEM
FORGETSPACE(FILEPARAM);
FORGETSPACE(CODE);
HEADERUNLOCK(MFID, FID, HDR);
END ELSE LBMESS(MFID, FID, 15, 0, 0, 0, 1); % FILE NOT ON DISK
END ELSE FORGETSPACE(T); % AUXMEMDSK NOT INITIALIZED
END; % IF AUXMEM ON LINE AND TO BE USED
$ POP OMIT
INTSIZE:=(INTRNSC[0] + 3 ) DIV 4;
$ SET OMIT = NOT(PACKETS)
TI=SPACE(15); WHATINTRNSIC(T);
STREAM(SI=T, DI=3);
BEGIN
  SII=S; DI:=DI+4; % CMBIT IN M[3],[1:1]
  63(IF SC="." THEN JUMP OUT; SII:=SII+1); S:=SII;
  4(SII:=SII+1; IF SC="." THEN JUMP OUT);
  IF TOGGLE THEN ELSE SII=S; SII:=SII+1;
  3(IF SC<"0" THEN JUMP OUT; TALLY:=TALLY+1; SII:=SII+1);
  SI=TALLY; SII:=SII-S; DI:=DI-S; DSI=S CHR;
END;
FORGETSPACE(T);
$ POP OMIT
END INTRINSIC TABLE BUILDER;
BOOLEAN PROCEDURE SYSTEMFILE(A,B); VALUE A,B; REAL A,B; FORWARD;
PROCEDURE CHANGEINTRNSICFILE(BUFF); VALUE BUFF; REAL BUFF;%
BEGIN
  REAL A,B,I,J,K,L,FH,T,IT;
  REAL SIZE=I, DISKADDR=T, LOC=IT, WI=J;
  LABEL EXIT, WITHOUT, NG;
  BOOLEAN SUBROUTINE NULLMIX;%
  BEGIN POLISH(1);%
    IF INTSIZE#0 THEN BEGIN INTSIZE + 0;%
      FOR I+1 STEP 1 UNTIL MIXMAX DO%
        IF JARROW[I]#0 THEN%
          IF (JARROW[I] INX 0)≥FENCE THEN GO NG ELSE
          IF NOT(JAR[I,9],[1:1]) THEN % NOT "SYSTEM" JOB
        NG: BEGIN P(DEL,0); I+MIXMAX END;
        IF NOT P(DUP) THEN INTSIZE + (INTRNSC[0]+3) DIV 4;%
      END;%
    NULLMIX + POLISH;%
  END NULLMIX;%
  SUBROUTINE FORGETEM;%
  BEGIN WAITSTORE(0); STOREDY[0]+0;
  WHILE (K + M[L]),[CF]#0 DO%
    BEGIN IF K>0 THEN%
      IF K,[3:12]=@700 THEN%
        FORGETSPACE(L+2);%
        L + K,[CF];%
      END;%
    STOREDY[0]+1;
  $ SET OMIT = NOT(AUXMEM)

```

```

09527500
09528000
09528500
09529000
09529500
09530000
09530500
09531000
09531500
09532000
09532500
09533000
09533500
09534000
09534100
09534200
09534201
09534500
09534999
09535000
09535100
09535200
09535300
09535400
09535500
09535600
09535700
09535800
09535900
09536000
09536001
09537000
09550000
09600000
09601000
09602000
09602100
09602200
09603000
09604000
09605000
09606000
09607000
09607100
09608000
09611000
09612000
09613000
09614000
09615000
09616000
09617000
09618000
09619000
09620000
09621000
09622000
09623000
09624000
09624010

```

```

FOR I:=1 STEP 1 UNTIL INTRNSC[0] DO IF INTRNSC[I].[3:1] THEN 09624020
  BEGIN % INTRINSIC ON AUXMEM 09624030
    FORGETAUXILIARYSPACE(INTRNSC[I],[CF],INTRNSC[I],[FF]); 09624040
    AUXCODE[0]:=(P(DUP))=(INTRNSC[I],[CF]),[38:6]=1; 09624050
    END; 09624060
$ POP OMIT 09624061
  FORGETSPACE(INTRNSC INX 0); INTRNSC=0 09624100
  END FORGETEM;% 09625000
  DEFINE ERROR = GO TO EXIT#;% 09626000

SLEEP([TOGGLE], FREEMASK); INTFREE ← FALSE;% 09627000
T ← BUFF;% 09629000
NAMEID(A,T); NAMEID(B,T); NAMEID(B,T);% 09630000
IF (FH:=DIRECTORYSEARCH(A,B,17))=0 THEN ERROR; 09631000
IF (J+M[FH+4],[36:6])≠0 THEN 09632000
  IF J≠DCINTYPE AND J≠TSSINTYPE THEN 09633000
  BEGIN % DONT ALLOW CI ON KNOWN NON-INTRINSICS FILE 09633100
    STREAM(A,B,NT1:=BUFF,[15:15]=1); 09633200
    BEGIN DS:=2LIT"# "; SI:=LOC A; 09633300
      SI:=SI+1; DS:=7CHR; DS:=LIT"/"; 09633400
      SI:=SI+1; DS:=7 CHR; 09633500
      DS:=24 LIT" NOT AN INTRINSICS FILE*"; 09633600
    END; 09633700
    FORGETSPACE(FH); 09633800
    FORGETSPACE(DIRECTORYSEARCH(A,B,16)); 09633900
    ERROR; 09634000
  END; 09634100
  IF NOT NULLMIX THEN COMPLEXSLEEP(NULLMIX); 09634200
  IF INTRNSC≠0 THEN FORGETEM; 09634300
$ SET OMIT = SHAREDISK 09635000
  IF MCPFREE=0 THEN SLEEP([TOGGLE],MCPMASK); 09636000
  LOCKTOG(MCPMASK); 09636999
$ POP OMIT 09637000
  T:=SPACE(30); 09638000
  DISKWAIT("T",30,0); 0963800$
  I:=T+13+5×SYSNO; 09639000
  IF (IT:=DIRECTORYSEARCH(M[I],M[I+1],16))≠0 THEN 09640000
  FORGETSPACE(IT); 09641000
  M[I]:=A; 09642000
  M[I+1]:=B; 09643000
  DISKWAIT(T,30,0); 09644000
$ SET OMIT = SHAREDISK 09645000
  UNLOCKTOG(MCPMASK); 09646000
$ POP OMIT 09646999
$ SET OMIT = NOT(AUXMEM) 09647000
  DISKWAIT("T",30,AUXMEMDSK); IF (P(RRR).[30:1] AND 09647001
  USEDRA) OR (P(RRR).[31:1] AND USEDRA) THEN 09647999
  IF M[T+SYSNO+16]="AUXMEM " THEN IF M[T+(SYSNO×4)+2]≠0 THEN 09648000
  BEGIN 09648100
    STREAM(I:=I:=SPACE(2)); DS:=14 LIT"CA INT PLEASE*"; 09648200
    SPOUT(I); CTABLE[4],[3:1]=1; 09648300
    COMPLEXSLEEP(NOT CTABLE[4],[3:1]); 09648400
  END; 09648500
$ POP OMIT 09648600
  FORGETSPACE(T); 09648700
% 09648701
  INTRINSICTABLEBUILDER(FH,[CF]); 09648800
  FORGETSPACE(FH); 09650000
  WHATINTRNSIC(BUFF,[15:15]); 09657000
  09658000
  09659000

```

```

        STREAM(B:=BUFF,[15:15]=1); DS:=8 LIT" NEW ";
EXIT: SPOUT(BUFF,[15:15]=1);%
      INTFREE + TRUE;%
END CHANGING INTRINSIC FILES ON USER DISK WITH MANY PRECAUTIONS;%
PROCEDURE CHANGEMCP(BUFF); VALUE BUFF; REAL BUFF;
BEGIN
  REAL RCW:=+0,A=RCW+1,B=A+1,T=B+1,Z=T+1,BASE=Z+1;
  LABEL EXIT;
  P(0,0,0,0,0);
  T:=BUFF;
  NAMEID(A,T); NAMEID(B,T); NAMEID(B,T);
  $ SET OMIT = SHAREDISK
  IF MCPFREE=0 THEN SLEEP([TOGGLE],MCPMASK);
  LOCKTOG(MCPMASK);
  $ POP OMIT
  Z:=SPACE(30);
  DISKWAIT(-Z,-30,0);
  BASE:=Z+10+5*SYSNO;
  IF (A EQV M[BASE])#NOT 0 OR
    (B EQV M[BASE+1])#NOT 0 THEN
  BEGIN
    IF (T:=DIRECTORYSEARCH(A,B,17))=0 THEN
    BEGIN;
      STREAM(A:=[A],T:=BUFF,[15:15]=1);
      BEGIN DS:=13 LIT"#NO MCP FILE ";SI:=A;SI:=SI+1;
        DS:=7 CHR;DS:=LIT"/";SI:=SI+1;DS:=7 CHR;
        DS=LIT"+";
      END;
      GO TO EXIT;
    END;
    IF (NT1:=M[T+4],[36:6])#0 THEN IF NT1#MCPTYPE THEN
    BEGIN % DONT ALLOW CM ON KNOWN NON-MCP FILE
      STREAM(A:=[A],T:=BUFF,[15:15]=1);
      BEGIN DS:=2LIT"# ";SI:=A;SI:=SI+1;
        DS:=7CHR;DS:=LIT"/";SI:=SI+1;
        DS:=7CHR;DS:=12LIT" NOT AN MCP+";
      END;
      FORGETSPACE(T);
      FORGETSPACE(DIRECTORYSEARCH(A,B,16));
      GO TO EXIT;
    END;
    IF M[BASE+2]=2#MCPBASE THEN
      FORGETSPACE(DIRECTORYSEARCH(M[BASE],M[BASE+1],16));
      M[BASE]:=A;
      M[BASE+1]:=B;
      M[BASE+2]:=M[T+10];
  $ SET OMIT = NOT(AUXMEM)
  DISKWAIT(-(T:=T INX 0),-30,AUXMEMDSK);
  M[T+(SYSNO*4)]:=NABS(*P(DUP)); % NOTE MCP CHANGE
  DISKWAIT( T,-30,AUXMEMDSK);
  $ POP OMIT
  FORGETSPACE(T);
  END;
  STREAM(A:=[A],T:=BUFF,[15:15]=1);
  BEGIN DS:=18 LIT " NEXT MCP WILL BE ";SI:=A;SI:=SI+1;
    DS:=7 CHR;DS:=LIT"/";SI:=SI+1;DS:=7 CHR;
    DS=LIT"+";
  END;
  M[3]+NABS(*P(DUP)); % SET FLAG FOR WM
EXIT:

```

```

09670000
09676000
09677000
09679000
09679100
09679200
09679300
09679400
09679600
09679800
09679900
09679999
09680000
09680100
09680101
09680200
09680300
09680400
09680500
09680600
09680700
09680800
09680900
09681000
09681100
09681200
09681250
09681300
09681400
09681500
09681505
09681510
09681515
09681520
09681525
09681530
09681535
09681540
09681545
09681550
09681555
09681600
09681650
09681700
09681800
09681900
09681909
09681910
09681920
09681930
09681931
09682000
09682100
09682200
09682300
09682400
09682450
09682500
09682550
09682600

```

DISKWAIT(Z,-30,0);	09682610
\$ SET OMIT = SHAREDISK	09682619
UNLOCKTOG(MCPMASK);	09682620
\$ POP OMIT	09682621
FORGETSPACE(Z);	09682700
SPOUT(BUFF,[15:15]-1);	09682800
BUFF:=0;	09683000
END CHANGING OF THE MCP;	09683100
BOOLEAN PROCEDURE SYSTEMFILE(A,B); VALUE A,B; REAL A,B;%	09700000
BEGIN LABEL DISK,LOG,TRUTH,DIR,SYS,REM,DECK,MASK,TEST;	09701000
LABEL DMP;	09701500
LABEL SCHED,FIL;	09701510
LABEL MAINT;	09701550
\$ SET OMIT = NOT(STATISTICS)	09701599
LABEL STATS;	09701600
\$ POP OMIT	09701601
DEFINE T=P(TRUTH)#;%	09702000
IF (B EQV P(DISK))=T THEN%	09703000
P(((A EQV P(DIR))=T) OR	09704000
((A EQV P(DMP))=T) OR	09704500
((A EQV P(LOG))=T))	09704550
ELSE IF (B EQV P(LOG))=T THEN%	09705000
P(((A EQV P(SYS))=T) %	09706000
\$ SET OMIT = SHAREDISK	09706049
OR ((A EQV P(MAINT))=T)%	09706050
OR ((A EQV P(REM))=T)%	09706100
\$ POP OMIT	09706101
)%	09706150
\$ SET OMIT = NOT(SHAREDISK)	09706199
ELSE IF (B EQV P(LOG) & (SYSNO+17)[24:42:6])=T THEN%	09706200
P(((A EQV P(MAINT))=T) %	09706300
OR ((A EQV P(REM))=T))%	09706400
\$ POP OMIT	09706401
ELSE IF (A EQV P(DECK))=T THEN%	09707000
P(((B AND P(MASK)) EQV P(TEST))=T)%	09708000
\$ SET OMIT = NOT(STATISTICS)	09708099
ELSE IF (B EQV P(STATS))= T THEN%	09708100
P((A EQV P("SYSTEM "	09708200
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	09708299
&(SYSNO+17)[42:42:6]	09708300
\$ POP OMIT	09708301
))=T)	09708400
\$ POP OMIT	09708401
ELSE IF (B EQV P(SCHED))=T THEN	09709000
P(((A,[6:18]) EQV P(FIL))=T);	09709100
P(RTN);%	09710000
DISK !!! "DISK ";%	09711000
LOG !!! "LOG ";%	09712000
TRUTH!!! @37777777777777777777;%	09713000
DIR !!! "DIRCTRY";%	09715000
SYS !!! "SYSTEM ";%	09716000
REM !!! "REMOTE ";%	09717000
DECK !!! "DECK ";%	09718000
MASK !!! @770000000077777777;%	09719000
TEST !!! @12000000003714;%	09720000
DMP !!! "DMPAREA";%	09720500
MAINT!!! "MAINT ";%	09720650
SCHED!!! "SCHEDUL";%	09720700
FIL!!! "0000FIL";	09720750
\$ SET OMIT = NOT(STATISTICS)	09720799

```

STATS::: "STATS ";%
$ POP OMIT
END;%
PROCEDURE WHATSGOINGON(BUFH); VALUE BUFH; REAL BUFH;
BEGIN
REAL I,MX,LM,BUFF,LINE,S,T,PRTM,ELTM,Q,Z;
LABEL NOONE,OK,EXIT,SPUUTIT,EGRESS,MIXUP;
ARRAY INF[*],MIXER[*];
BOOLEAN GETBUFF;
REAL MIX,RCW=+0;
MIX I= BUFH,[9:6];
LMI=STATIONMAX+1; BUFF:=BUFH,[15:15]=1;
STREAM(L=+1:B+BUFH);
BEGIN
SI← B; DI←LOC L;
DD: IF SC=" " THEN BEGIN SI← SI+1; GO DD END; B← SI;
IF SC>"0" THEN IF SC<"9" THEN
BEGIN
SI← SI+1;
IF SC>"0" THEN IF SC<"9" THEN
BEGIN
SI← B; DS← 2 OCT;
GO EG;
END;
SI← B; DS← OCT;
END;
EG: END;
IF (LINE:=P)=0 OR LINE GTR STATIONMAX THEN
GO EXIT;
IF SYSDISKADR NEQ 0 THEN
INF:=[M[SPACE(SYSDISKRL)]]&SYSDISKRL[8:38:10];
MIXER:=[M[SPACE(MIXMAX+1)]]&(MIXMAX+1)[8:38:10];
IF (I← MIX)≠0 THEN GO MIXUP;
MIXER[0]← 1;
MOVE(MIXMAX,MIXER,MIXER INX 1);
IF SYSDISKADR NEQ 0 THEN
BEGIN
IF LINE>0 THEN
BEGIN
SYSDISKIO(1,-LINE,INF);
I:=LINE;
IF NOT (IF LINE GTR LMAX THEN 0 ELSE SCHEDLINE[LINE]) THEN
IF NOT STABLE[LINE],DIALEDUP THEN
BEGIN
GETBUFF← TRUE;
STREAM(BUFF); DS←8 LIT " NULL ";
SPOUT(BUFF);
GO EGRESS;
END;
IF INF[1] LEQ 0 THEN GO TO NOONE ELSE
GO SPUUTIT;
END;
FOR I← 0 STEP 1 UNTIL LM DO
BEGIN
SYSDISKIO(1,I,INF);
IF I LEQ STATIONMAX THEN
IF STABLE[I],DIALEDUP THEN
IF INF[1] GTR 0 THEN GO TO OK ELSE
IF NOT LINETABLE[
$ SET OMIT = TWXONLY

```

```

09720800
09720801
09721000
%DS%09800000
09801000
09802000
09803000
09804000
09805000
%DS%09805100
%DS%09805200
09806000
09807000
09808000
09809000
09810000
09811000
09812000
09813000
09814000
09815000
09816000
09817000
09818000
09819000
09820000
09821000
09822000
09823000
09823100
09824000
09825000
09825100
09826000
09827000
09827100
09827200
09828000
09829000
09830000
09830500
09831000
09831500
09832000
09833000
09834000
09835000
09835500
09836000
09836500
09837000
09838000
09839000
09840000
09841000
09841400
09841500
09842000
09842300
09842399

```

```

IF I GTR LMAX THEN STABLE[I],LEENKER ELSE 09842400
$ POP OMIT 09842401
  I],DIRECTLINE THEN 09842500
    BEGIN 09843000
NOONE:  INF[1]:="NOBODY "; 09843500
OK:  IF GETBUFF THEN 09844000
      BUFF:=SPACE(10) 09845000
    ELSE 09846000
      GETBUFF← TRUE; 09847000
      QI:=1; 09847050
      SI:=IF I GTR LMAX THEN 0 ELSE SCHEDULE[I]; 09847060
      IF (MXI=STABLE[I],MIXNR) GTR 0 THEN 09847100
      IF MX≠CANDYINX THEN 09847110
      BEGIN TABCNT[MX]:=TABCNT[MX]+1; 09847150
        IF (T:=PUTORTAKE(MX,[JAR[MX,3]],1&1[2:47:1],0))=NOT 09847200
        THEN QI:=0 ELSE 09847210
        BEGIN 09847220
          T:=T+PUTORTAKE(MX,[PROCTIME[MX]],1,0); 09847230
          Q:=PUTORTAKE(MX,[JAR[MX,0]],1,0); 09847240
          Z:=PUTORTAKE(MX,[JAR[MX,1]],1,0); 09847250
          IF MX=P2MIX THEN T← T+CLOCK+P(RTR); 09847300
          FORMTIME([PRTM],T); 09847350
          T← ((CLOCK+P(RTR))/60)-PUTORTAKE(MX,[LUV[MX,9]],1,0) 09847400
            ,[1:17]; 09847450
          FORMTIME([ELTM],T×60); 09847500
        END; 09847550
        TABCNT[MX]:=TABCNT[MX]-1; 09847600
      END ELSE MXI=0; 09847650
      STREAM(X:=Q=0, S, U:=INF[1], I, 09849000
        Q, Z, M:=MX, N:=PRTM, O:=ELTM, B:=BUFF); 09850000
      BEGIN 09851000
        SI←LOC U; SI←SI+1; DS←LIT " "; 09852000
        DS←7 CHR; DS←4 LIT" ON "; U←DI; 09853000
        DS← 2 DEC; B← DI; DI← U; DS← FILL; DI← B; 09854000
        S(DS:=8 LIT"-(SCHED)"); 09854500
        M(DS← 7 LIT " USING "; 09855000
        X(DS:=15 LIT"HUNG JOB: MIX ="; SI:=LOC M; DS:=2 DEC; 09855100
          U:=DI; DI:=DI-2; DS:=2 FILL; DI:=U; 09855200
          JUMP OUT 2 TO EXIT); 09855300
        SI← SI+1; DS← 7 CHR; DS← LIT "/"; 09856000
        SI← SI+1; DS← 7 CHR; DS← LIT "="; 09857000
        DS←2 DEC; U←DI; DI←DI-2; DS←FILL; DI←U; 09857500
        DS←5 LIT " PST="; DS←8 CHR; 09858000
        DS← 4 LIT " IN "; DS← 8 CHR; JUMP OUT); 09858100
      END; 09859000
    EXIT; 09860000
      END; 09861000
      SPOUT(BUFF); 09861100
      IF REPLY[MX]<0 THEN 09861150
      IF Q≠0 THEN 09861200
        BEGIN 09861300
          REPLY[MX]← VWY; 09861350
          BRINGBACK(MX); 09861360
          CLICK:=CLOCK+P(RTR)+240; 09861370
          SLEEP([REPLY[MIX]],-0); 09861400
        END; 09862000
      IF LINE>0 THEN GO EGRESS; 09863000
      MIXER[MX]← 0; 09864000
    END; 09865000
  END; 09865100
END;

```

FOR I← 1 STEP 1 UNTIL MIXMAX DO	09866000
IF MIXER[I]≠0 THEN	09867000
IF JAR[I,*]≠0 THEN	09868000
MIXUP:	09869000
BEGIN	09869250
TABCNT[I]:=TABCNT[I]+1;	09869500
Q:=0;	09870000
IF GETBUFF THEN	09871000
BUFF:=SPACE(10)	09872000
ELSE	09873000
GETBUFF← TRUE;	09873100
IF (T:=PUTORTAKE(I,[JAR[I,3]],1&1[2:47:1],0))≠NOT 0	09873110
THEN STREAM(I,BUFF);	09873120
BEGIN DS:=4 LIT" MIX ";	09873130
SI:=LOC I; DS:=2 DEC;	09873140
DS:=9 LIT" IS HUNG*";	09873150
DI:=DI-11; DS:=2 FILL;	09873160
END ELSE	09873200
BEGIN T:=T+PUTORTAKE(I,[PROCTIME[I]],1,0);	09873300
IF I=P2MIX THEN T← T+CLOCK+P(RTR);	09873400
FORMTIME([PRTM],T);	09873500
T← ((CLOCK+P(RTR))/60)=PUTORTAKE(I,[UV[I,9]],1,0)	09873600
,[1117]);	09873700
FORMTIME([ELTM],T×60);	09873800
Q:=1;	09874000
STREAM(U← PUTORTAKE(I,[USERCODE[I]],1,0),	09875000
Q←PUTORTAKE(I,[JAR[I,0]],1,0),	09876000
Z← PUTORTAKE(I,[JAR[I,1]],1,0), M← I,	09876100
N← PRTM, O← ELTM, B← BUFF);	09877000
BEGIN	09878000
SI←LOC U; SI←SI+1; DS←LIT " "; DS←7 CHR;	09879000
DS← 7 LIT " USING ";	09880000
SI← SI+1; DS← 7 CHR; DS← LIT "/";	09881000
SI← SI+1; DS← 7 CHR; DS← LIT "=";	09882000
DS← 2 DEC; DS← 5 LIT " PST="; DS← 8 CHR;	09882100
DS← 4 LIT " IN "; DS← 8 CHR; DS← LIT " ";	09883000
END;	09883500
END;	09883750
TABCNT[I]:=TABCNT[I]-1;	09884000
SPOUT(BUFF);	09884100
IF REPLY[I]<0 THEN	09884150
IF Q THEN	09884200
BEGIN	09884300
REPLY[I]← VWY;	09884400
BRINGBACK(I);	09884410
CLICK:=CLOCK+P(RTR)+240;	09884420
SLEEP([REPLY[MIX]],-0);	09884500
END;	09884600
IF MIX≠0 THEN GO EGRESS;	09885000
END;	09886000
EGRESS:	09887000
FORGETSPACE(MIXER);	09887100
IF SYSDISKADR NEQ 0 THEN	09888000
FORGETSPACE(INF);	09889000
IF GETBUFF THEN	09890000
BEGIN STREAM(B:=BUFF:=SPACE(5)); DS I= 7 LIT"END SM*"; END	09891000
ELSE STREAM(BUFF); DS I= 8 LIT"NOTHING*";	09892000
EXIT; SPOUT(BUFF);	09894000
END;	09900000
PROCEDURE FENCEMOVER(B, BUFF);	09901000
VALUE B, BUFF;	

REAL B, BUFF;	09902000
BEGIN LABEL EXIT;	09903000
INTEGER T, A;	09904000
;STREAM(T:=0; B);	09905000
BEGIN SI:=B; IF SC=" " THEN BEGIN	09906000
L: SI:=SI+1; IF SC=" " THEN GO TO L END;	09907000
IF SC>"0" THEN IF SC<"9" THEN	09908000
BEGIN BI:=SI; SI:=SI+1; TALLY:=1;	09909000
7(IF SC<"0" THEN JUMP OUT; IF SC>"9" THEN JUMP OUT;	09910000
SI:=SI+1; TALLY:=TALLY+1);	09911000
SI:=B; BI:=TALLY; DI:=LOC T; DS:=B OCT;	09912000
END END STREAMING;	09913000
T := POLISH;	09914000
T := @100000-(((P(DUP)=T)DIV CHUNKZIZE)*CHUNKZIZE);	09915000
IF T<@20000 OR T>@70000 THEN	09916000
BEGIN SPOUT(BUFF=1); GO TO EXIT END;	09917000
SLEEP([TOGGLE],HOLDMASK); LOCKTOG(HOLDMASK);	09918000
DISKWAIT(=(A:=SPACE(30)),=-30,DIRECTORYTOP-SYSNO);	09919000
M[A+19]:=T;	09920000
DISKWAIT(A,-30,DIRECTORYTOP-SYSNO);	09920100
UNLOCKTOG(HOLDMASK);	09921000
STREAM(T,A);	09922000
BEGIN DS:=22 LIT " FENCE TO BE MOVED TO ";	09923000
A:=DI; SI:=LOC T; DS:=8 DEC;	09924000
DS:=LIT "+"; DI:=A; DS:=7 FILL;	09925000
END STREAMING;	09926000
SPOUT(A); FORGETSPACE(BUFF=1);	09927000
EXIT; END FENCEMOVER;	09928000
PROCEDURE LINECLEAR(KTR);	09955000
VALUE KTR; REAL KTR;	09956000
BEGIN LABEL START,BADLINE;	09957000
REAL BUFF,LINE,X;	09958000
START; ;	09958500
STREAM(LINE:=1, X:=0, KTR);	09959000
BEGIN SI:=KTR; DI:=LOC LINE;	09960000
L: IF SC="+" THEN GO TO EXIT;	09961000
IF SC<"0" THEN BEGIN SI+SI+1; GO TO L END;	09962000
IF SC>"9" THEN BEGIN SI+SI+1; GO TO L END;	09963000
SI:=SI+1;	09964000
IF SC<"0" THEN GO TO ONE;	09965000
IF SC<"9" THEN	09966000
BEGIN SI+SI-1; DS+2 OCT END	09967000
ELSE BEGIN	09968000
ONE: SI+SI-1; DS+1 OCT END;	09969000
IF SC="&" THEN	09969600
BEGIN SI:=SI+1; TALLY:=8; X:=TALLY; END;	09969700
KTR+SI;	09970000
EXIT; END STREAM;	09971000
KTR+P;	09972000
X:=P; LINE:=P;	09972500
IF LINE GEQ 0 THEN	09973000
BEGIN	09973500
BUFF:=SPACE(5);	09974000
IF LINE#0 AND LINE LEQ LMAX THEN	09974500
BEGIN	09975000
IF SCHEDLINE[LINE] THEN	09975100
BEGIN IF (X=8) THEN SCHEDBUSY[LINE]:=1;	09975110
IF SCHEND[LINE] THEN X:=0	09975115
ELSE IF (X:=SEQARRAY[LINE],[CF])GTR 511 THEN	09975120
BEGIN SLEEP([M[X+32]],=-0); M[X+32]:=ABS(*P(DUP)) END;	09975125



```

STREAM(Y:=(X GTR 511),LINE,BUFF); 09975130
BEGIN SI:=LOC LINE; DS:=9 LIT" STATION "; DS:=3 DEC; 09975140
LINE:=DI; DI:=DI-3; DS:=2 FILL; DI:=LINE; 09975145
Y(DS:=7LIT" CLEAR"; JUMP OUT TO LL); 09975150
DS:=6LIT" IDLE"; 09975160
LL: END; 09975170
SPOUT(BUFF); 09975180
IF X LEQ 511 THEN GO TO START; 09975185
IF (NT1:=STATABLE[LINE],MIXNR) GTR 0 THEN 09975190
IF NT1 NEQ CANDYINX THEN 09975200
BEGIN TERMINATE(NT1&3[CTF]); 09975210
HALT; NOPROCESSTOG:=NOPROCESSTOG-1; 09975220
END ELSE SHEETDIDDLER(0,-1,LINE); 09975230
M[X+77]:=M[X+78]; 09975240
STREAM(BUFF:=BUFF:=SPACE(5)); 09975250
DS:=30LIT" **TASK TERMINATED BY OPERATOR"; 09975260
M[X+32]:=NABS(*P(DUP)); 09975270
SCHEDIO(30,1,BUFF&LINE[CTF]); 09975280
FORGETSPACE(BUFF); 09975290
GO TO START; 09975300
END; 09975310
IF LINETABLE[LINE]#0 THEN 09980000
IF BLASTREAD(LINE,X+7) THEN 09981000
BEGIN 09982000
IF LINEDISC[LINE]=MULTI THEN ENTERREADYQ(LINE); 09982300
IF STATABLE[LINE],OUTPUTANKING THEN 09982400
IF TANKLINE[LINE]=0 AND TAILOUT#LINE THEN 09982500
BEGIN 09982600
TANKLINE[TAILOUT]:=LINE; 09982700
TAILOUT:=LINE; 09982800
STARTWORKING; 09982900
END; 09983000
X:=0; 09983100
END ELSE X:=2 09983200
ELSE 09983300
BADLINE: X:=1; 09983400
STREAM(X,LINE,BUFF); 09983500
BEGIN SI:=LOC LINE; 09983600
DS:=6 LIT" LINE "; DS:=3 DEC; 09983700
BUFF:=DI; DI:=DI-3; DS:=2 FILL; DI:=BUFF; 09983800
CI:=CI+X; GO TO OK; GO TO BADNUM; 09983900
DS:=14 LIT" DID NOT CLEAR"; GO TO L; 09984000
OK: DS:= 6 LIT" CLEAR"; GO TO L; 09984100
BADNUM: DS:=15 LIT" DOES NOT EXIST"; 09984200
L: DS:=LIT"+"; 09984300
END; 09984400
SPOUT(BUFF); 09984600
GO TO START; 09984800
END ELSE GO TO BADLINE; 09985000
END; 09985500
END LINE CLEARING; 09986000
& SET OMIT = NOT(DEBUGGING) 09999999
COMMENT GNC RETURNS CHARACTER AT B AND BUMPS B BY ONE ;% 10000000
REAL STREAM PROCEDURE GNC(B);% 10001000
BEGIN LOCAL T; SI + B; DI+ LOC T; DS+ WDS;% 10002000
SI + T; SI+SI+1; DI+LOC GNC ; DI+DI+7; DS+CHR ; SI+SI-1;% 10003000
DI+ B ; T+ SI ; SI+ LOC T ; DS + WDS;% 10004000
END GNC ;% 10005000
COMMENT TAN TEST C FOR ALPHA AND RETURNS RESULT;% 10006000
BOOLEAN STREAM PROCEDURE TAN(C) ; VALUE C ;% 10007000

```

BEGIN%	10008000
SI←LOC C; SI←SI+7; IF SC=ALPHA THEN TALLY←1;%	10009000
TAN← TALLY;%	10010000
END TEST ALPHAMERIC ;%	10011000
REAL PROCEDURE EXP;	10012000
BEGIN%	10013000
REAL X,T,I,P;%	10014000
LABEL G,BOOB00;%	10015000
T ← 0; X ← "+";%	10016000
IF C = "+" OR C = "-" THEN%	10017000
BEGIN X←C; T← LP; C ← GNC(B); END;%	10018000
DO BEGIN%	10019000
P ← 0;%	10020000
IF C="+" OR C="-" THEN C←GNC(B);%	10021000
IF C = "*" THEN BEGIN P←LP;C←GNC(B) END;%	10022000
ELSE IF C = "/" THEN BEGIN P ← M[LP],[18;15]; C ← GNC(B) END;%	10023000
ELSE IF C = "=" THEN BEGIN P ← M[LP],[33;15]; C ← GNC(B) END;%	10024000
ELSE IF C ≤ 7 THEN DO P←P×8 + C UNTIL (C← GNC(B))>7;%	10025000
ELSE IF C = "," THEN%	10026000
BEGIN C←GNC(B); DO P←P×10+C UNTIL (C←GNC(B))>9 END;%	10027000
ELSE IF TAN(C) THEN%	10028000
BEGIN DO P← P×64+C UNTIL NOT (TAN(C←GNC(B)));%	10029000
FOR I← 0 STEP 2 UNTIL NSYMS DO%	10030000
IF TBL[I],[6;42] = P THEN%	10031000
BEGIN P←TBL[I+1] ; GO TO G END;%	10032000
ERROR ← "NO SYMB" ; GO TO BOOB00 ;%	10033000
END;%	10034000
ELSE IF C = "[" THEN%	10035000
BEGIN C ← GNC(B);%	10036000
P← EXP ; IF ERROR ≠ 0 THEN GO BOOB00 ;%	10037000
P ← POLISH([MEMORY[P]],LOD,0,INX) ;%	10038000
IF C ≠ "]" THEN%	10039000
BEGIN ERROR← "NO BRKT";%	10040000
DO UNTIL (C←GNC(B))="←" END ELSE C←GNC(B);%	10041000
END;%	10042000
BOOB00:	10043000
G: T← T + (IF X="+" THEN P ELSE -P);%	10044000
END UNTIL (( X ←C)≠"+" AND C ≠ "-" ) ;%	10045000
EXP ← T;%	10046000
END EXP ;%	10047000
STREAM PROCEDURE FRMT1(Q,T,M,WB); VALUE Q,T,M;%	10048000
BEGIN%	10049000
SI← LOC T; SKIP SB; DI←WB; DI←DI+1;%	10050000
IF SB THEN%	10051000
BEGIN%	10052000
SI ← SI+0;%	10053000
7(IF SC = "0" THEN SI←SI+1 ELSE DS←CHR) ;%	10054000
END;%	10055000
ELSE%	10056000
BEGIN%	10057000
SI←SI+4; SKIP 3 SB;%	10058000
5(DS←3RESET)3(IF SB THEN DS←SET ELSE DS←RESET;%	10059000
SKIP SB)) END;%	10060000
DS ← LIT "=";%	10061000
SI←LOC Q; SI←SI+7;%	10062000
IF SC≠"0" THEN BEGIN%	10063000
IF SC = "D" THEN BEGIN%	10064000
SI← LOC M; DS← 8 DEC END ELSE BEGIN%	10065000
SI← LOC M; DS← 8 CHR END; DS←LIT"←" END;%	10066000
ELSE BEGIN%	10067000
SI ← LOC M;%	

IF SB THEN%	10068000
BEGIN%	10069000
DS + 3 RESET;3(IF SB THEN DS+SET ELSE DS+RESET;	10070000
SKIP SB);%	10071000
3(5(DS+3 RESET;3(IF SB THEN DS+SET ELSE%	10072000
DS+RESET ; SKIP SB));%	10073000
DS + LIT " " ) ;%	10074000
END%	10075000
ELSE%	10076000
2(8(DS+3 RESET;3(IF SB THEN DS+SET ELSE DS+RESET	10077000
; SKIP SB)); DS+ LIT " " );%	10078000
DI+ DI-1; DS + LIT"+";%	10079000
END%	10080000
END ;%	10081000
PROCEDURE DT;	10082000
BEGIN%	10083000
REAL CL,LITC,I,T,N;%	10084000
LABEL LA,LB,FOUND,BANG,EXIT;%	10085000
DEFINE READSPO = DO UNTIL WAITIO(1 INX RBX INX @40000000,@377,	10086000
25) = 0 #;%	10087000
DEFINE WRITESPO = DO UNTIL WAITIO(WB INX 0,@377,25) = 0 #;%	10088000
REAL RCW=+0;%	10089000
POLISH(0,RDS,5,-,0,XCH,CFX,STF);%	10090000
HALT;%	12000000
SLEEP([TOGGLE],KEYBOARDMASK); LOCKTOG(KEYBOARDMASK);	12001000
STREAM(C+(RCW INX 0)&RCW[30:10:2];WB);%	12002000
BEGIN%	12003000
DI+ WB; DS+ LIT""; DS+6 LIT"C:L = " ;%	12004000
SI+ C; SI+SI+2; C+SI;%	12005000
SI+ LOC C; SI+SI+5; SKIP 3 SB;%	12006000
5(DS+ 3 RESET;3(IF SB THEN DS+SET ELSE DS+RESET;SKIP SB))	12007000
; DS+LIT"!" ; SI+LOC C; SI+SI+5;%	12008000
DS+4RESET;2(IF SB THEN DS+SET ELSE DS+RESET; SKIP SB);%	12009000
DS+LIT"+";%	12010000
END;%	12011000
POLISH(,CL,+);%	12012000
TBL[35]+M[RCW,[18:15]],,[18:15];%	12013000
WRITESPO;%	12014000
LITC+0;%	12015000
FOR I+1 STEP 1 UNTIL NSTOP DO%	12016000
IF STOPS[I],[30:18] = CL THEN LITC+ STOPS[I],[18:10];%	12017000
READSPO ;%	12018000
NOBACKTALK+FALSE;%	12019000
ERROR+0;%	12020000
B + RBX INX 1;	12021000
STREAM(Q+B);%	12022000
BEGIN%	12023000
SI+Q; DI+Q; DI+DI+1;%	12024000
LI IF SC#"+" THEN%	12025000
BEGIN%	12026000
IF SC=@14 THEN SI+SI+1 ELSE%	12027000
IF SC= " " THEN SI+SI+1 ELSE DS+CHR;%	12028000
GO L;%	12029000
END;%	12030000
DS+ CHR;%	12031000
END;%	12032000
C+GNC(B);%	12033000
IF C="+" THEN GO TO LA;%	12034000
IF C=">" THEN BEGIN TYPETOG+GNC(B); C+GNC(B) END;%	12035000
IF C="&" THEN%	12036000

BEGIN%	12037000
C+GNC(B);%	12038000
T+EXP; IF ERROR # 0 THEN GO LB;%	12039000
STREAM( L+0;T,S+POLISH(,DT,4,x,2,+),V+0);%	12040000
BEGIN%	12041000
SI+T;%	12042000
Q: SI+SI+0; L+SI; SI+SI+1; SKIP 4 SB;%	12043000
IF SB THEN GO Q; SKIP SB;%	12044000
IF SB THEN GO Q;%	12045000
SI+SI+0; SI+SI-2;%	12046000
DI+ LOC L; DI+DI +3;%	12047000
DS+ 2 CHR; DI+ L;%	12048000
SI+ LOC S; SI+SI+6; DS+ 2 CHR;%	12049000
END;%	12050000
POLISH([STOPS[NSTOP+NSTOP+1]],+);%	12051000
STREAM( CL+STOPS[NSTOP],WB);%	12052000
BEGIN%	12053000
DI+ WB; DI+DI+1; SI+LOC CL;%	12054000
SI+ SI+5; SKIP 3 SB;%	12055000
5( DS+ 3 RESET; 3(IF SB THEN DS+SET ELSE DS+ RESET	12056000
;SKIP SB));%	12057000
DS+ LIT"; SI+ LOC CL; SI+SI+5;%	12058000
DS+ 4 RESET;%	12059000
2( IF SB THEN DS+SET ELSE DS+RESET; SKIP SB);%	12060000
DS+LIT"+";%	12061000
END;%	12062000
WRITESPO;%	12063000
GO TO LA;%	12064000
END;%	12065000
IF C="@" THEN%	12066000
BEGIN%	12067000
C+GNC(B);%	12068000
T+EXP; IF ERROR#0 THEN GO TO LB;%	12069000
IF C="!" THEN T+T&(C+GNC(B))[30;46;2];%	12070000
FOR I+1 STEP 1 UNTIL NSTOP DO%	12071000
IF STOPS[I],[30;18] = T THEN GO FOUND;%	12072000
ERROR+ "NO STOP"; GO TO LB;%	12073000
FOUND;%	12074000
STREAM ( L+STOPS[I],V+0);%	12075000
BEGIN DI+L;SI + LOC L; SI+SI+3; DS + 2 CHR END;%	12076000
GO TO LA;%	12077000
END;%	12078000
IF C = "x" THEN%	12079000
BEGIN%	12080000
C + GNC(B);%	12081000
FOR I + 0 STEP 2 UNTIL NSYMBS DO%	12082000
IF TBL[I],[1;5] = C THEN BEGIN%	12083000
FRMT1(TYPETOG,=TBL[I],MEMORY[TBL[I+1]],WB);%	12084000
WRITESPO;%	12085000
IF NOBACKTALK THEN GO TO LA;%	12086000
END;%	12087000
GO TO LA;%	12088000
END;%	12089000
IF C = ";" THEN GO EXIT;%	12090000
T+ EXP ; IF ERROR # 0 THEN GO LB;%	12091000
IF C = ";" THEN%	12092000
BEGIN%	12093000
C+GNC(B);%	12094000
N+EXP-1; IF ERROR #0 THEN GO LB ELSE GO BANG;%	12095000
END;%	12096000

```

N ← 0;%
IF C = ";" THEN%
  BEGIN%
    CL ← 0; I ← 0;%
    IF (C+GNC(B)) ≤ 9 THEN CL ← C;%
    DO I ← I × 64 + C UNTIL NOT(TAN(C+GNC(B)));%
  NSYMB ← NSYMB + 2;%
  TBL[NSYMB] ← I & CL[1;43;5]; TBL[NSYMB+1] ← T;%
  GO BANG;%
  END;%
IF C = "=" THEN%
  BEGIN;%
  IF (C+GNC(B)) ≤ 9 THEN BEGIN;%
  STREAM( I ← 0;%
    WB ← B, L ← 0, K ← 0);%
    BEGIN%
      SI ← WB;%
      16( IF SC ≥ "0" THEN BEGIN TALLY ← TALLY + 1; SI ← SI + 1 END%
        ELSE JUMP OUT TO Z ); Z;%
      L ← TALLY; TALLY ← 16;%
      L( TALLY ← TALLY + 63); K ← TALLY; DI ←%
        LOC I; K(SKIP 3 DB);%
      SI ← WB;%
      L(SKIP 3 SB; 3( IF SB THEN DS ← SET ELSE DS ← RESET;%
        SKIP SB ));%
    END;%
    POLISH( [MEMORY[T]], ← );%
  END ELSE%
  IF C = "." THEN%
    BEGIN I ← 0; WHILE (C+GNC(B)) ≤ 9 DO I ← I × 10 + C;%
      MEMORY[T] ← I;%
  END ELSE BEGIN I ← 0; DO I ← I × 64 + C UNTIL NOT (TAN(C+GNC(B)));%
    MEMORY[T] ← I END;%
  GO BANG;%
  END;%
IF C ≠ "." THEN%
  BEGIN ERROR ← "ILL EXP"; GO LB END;%
BANG;%
FOR I ← 0 STEP 1 UNTIL N DO%
  BEGIN%
    FRMT( TYPETOG, LP ← T + I, POLISH( [MEMORY[LP]], LOD ), WB );%
    WRITESP;%
    IF NOBACKTALK THEN GO TO LA;%
  END;%
  GO TO LA;%
LB:
  STREAM( ERROR, WB, RBX );%
  BEGIN%
    SI ← RBX; DI ← WB; DI ← DI + 1;%
    SI ← SI + 9;%
    L: IF SC ≠ "." THEN BEGIN DS ← CHR; GO TO L END;%
    DS ← 6LIT "ERROR "; SI ← LOC ERROR; SI ← SI + 1;%
    DS ← 7 CHR; DS ← LIT ".";%
  END;%
  WRITESP;%
  GO TO LA;%
EXIT;%
M[WB] ← NOT 0; WRITESP;%
UNLOCKTOG( KEYBOARDMASK );%
NOPROCESSTOG ← NOPROCESSTOG - 1;%
POLISH( LITC, RTN );%

```

```

12097000
12098000
12099000
12100000
12101000
12102000
12103000
12104000
12105000
12106000
12107000
12108000
12109000
12110000
12111000
12112000
12113000
12114000
12115000
12116000
12117000
12118000
12119000
12120000
12121000
12122000
12123000
12124000
12125000
12126000
12127000
12128000
12129000
12130000
12131000
12132000
12133000
12134000
12135000
12136000
12137000
12138000
12139000
12140000
12141000
12142000
12143000
12144000
12145000
12146000
12147000
12148000
12149000
12150000
12151000
12152000
12153000
12154000
12155000
12156000

```

```

        END;%
$ POP OMIT
REAL PROCEDURE PRNPBTSPECASE1(Z);
%
% THIS PROCEDURE HANDLES THE FOLLOWING FUNCTIONS FOR COM19, DEPENDING
% ON THE VALUE OF Z:
% 0 FINDS THE NEXT REEL OF TAPE,
% 1 FINDS THE NEXT REEL OF A BACK-UP DISK FILE,
% 2 HANDLES THE QT + OR = MESSAGE,
% 3 INITIALIZES A NEW FILE (OR PACKET),
% 4 HANDLES TERMINATION OF A FILE,
%
VALUE Z; REAL Z;
BEGIN
    REAL RCW=+0, MSCW=-2, COMMON=-4;
    ARRAY INREC=+1[*];
    ARRAY FPB=INREC+1[*], LOGINFO=FPB+1[*], HEADER=LOGINFO+1[*];
    REAL UNIT=HEADER+1, V=UNIT+1, COPY=V+1, MFID=COPY+1, FID=MFID+1,
        IOD=FID+1, T=IOD+1, B=T+1;
    REAL SEARCHVAL=B+1, CURROW=SEARCHVAL+1, FIRSTFID=CURROW+1,
        SEGNR=FIRSTFID+1;
    REAL X=SEARCHVAL, NUM=CURROW, RECOUNT=SEGNR;
    BOOLEAN SIGNEDON=SEGNR+1, FORMTOG=SIGNEDON+1, ABORTED=FORMTOG+1;
    BOOLEAN TERMFLAG=LOGINFO, NOCUNT=FIRSTFID;
$ SET OMIT = NOT PACKETS
    BOOLEAN STOG=ABORTED+1;
    REAL PCOPY=STOG+1, PFIRSTFID=PCOPY+1;
$ SET OMIT = PACKETS
    REAL PFIRSTFID=FIRSTFID, PCOPY=COPY;
$ POP OMIT OMIT

    LABEL RD, RED, SPACEND, NOMORE, NOFILE, AUT, BOMBER, NEXTFILE,
        PNCHLK, PRINTITAGAIN, EOF, PRNTDS, PNCHDS, TAPEND, CONTINUE,
        RETURNFALSE, REMOVE, TEST, TAPECL, STOPTIME, RETURNTRUE,
        RETURNTOCOM19;
    LABEL LOOK4TAPE, NOMOREELS, QTSPEC, INITIALIZE, STARTANNEWFILE;
    SWITCH SW :=
        LOOK4TAPE, NOMOREELS, QTSPEC, INITIALIZE, STARTANNEWFILE;
    DEFINE DSED = (TERMSET(P1MIX))#,
        QTED = (PRT[P1MIX,@25]#0)#,
        VF = 4315#,
        UNITF = 3815#,
        COPYF = 3018#,
        NUMF = 2218#,
        NOTP = 2911#,
        COPYO = 2111#;
$ SET OMIT = PACKETS
    REELNO = 3018#,
$ SET OMIT = NOT PACKETS
    REELNO = 4216#,
$ POP OMIT OMIT
    SEPARATION = 46#; % FOR 6 LPI, SET IT TO 70 FOR 8 LPI.
%*****
SUBROUTINE RDYTAPE;
BEGIN
    B,[18:9];=#54;
    P(WAITIO(@4200000000,0,UNIT),DEL);
    P(WAITIO(B,0,UNIT),WAITIO(B,@40,UNIT),DEL,DEL);
    RECOUNT:=@77777;

```

```

12157000
12157001
12500000
12500100
12500110
12500120
12500130
12500140
12500150
12500160
12500170
12500180
12500500
12501000
12501500
12502000
12502500
12503000
12503500
12504000
12504500
12505000
12505500
12506000
12506500
12507000
12507500
12509500
12510000
12511500
12512000
12512500
12513000
12513500
12513750
12514000
12514500
12515000
12515500
12516000
12516100
12516150
12516200
12516250
12516300
12516350
12516500
12517000
12517500
12518000
12518500
12519000
12519500
12520000
12520500
12521000
12521500
12522000
12522500
12523000

```

```

END;
12523500
12524000
*****12524500
SUBROUTINE REWIND;
12525000
BEGIN
12525500
  STOPTIMING(0,1023);
12526000
  P(WAITIO(@4200000000,0,UNIT),DEL);
12526500
  IF (SAVEWORD AND TWO(UNIT))=0 AND PRNTABLE[UNIT],[1:1]
12527000
    AND NOT (SVPBT OR QTED OR NOCONT) THEN
12527500
      FORK(P(,PURGEIT),UNIT,-2,128,1)
12528000
    ELSE
12528500
      BEGIN LABELTABLE[UNIT]*@114;
12529000
        MULTITABLE[UNIT]*RDCTABLE[UNIT]*0;
12529500
        SLEEP([TOGGLE],STATUSMASK);
12530000
        READY*READY AND NOT NT1* TWO(UNIT);
12530500
        RRRMECH*NOT NT1 AND RRRMECH OR NT1 AND SAVEWORD;
12531000
      END;
12531500
END;
12532000
12532500
12533000
*****12533500
BOOLEAN SUBROUTINE LOOKFORTAPE;
12534000
BEGIN
12534500
  T:=RDCTABLE[UNIT];
12535000
  REWIND;
12535500
  IF SIGNEDON THEN FPB[4]:=FPB[4]-LOGINFO[18]-CLOCK+P(RTR);
12536000
  IF P((T:=FINDINPUT(MFID,@122212342546447,T,[14:10]+1,T,[24:17],
12537000
    =0,0,T:=0,0,0,0)) GEQ 0, DUP) THEN
12537500
    BEGIN
12538000
      RDCTABLE[UNIT:=T],[8:6]:=P1MIX;
12538500
      LABELTABLE[UNIT]:=FID;
12539000
      FPB:=PRT[P1MIX,3];
12539500
      % FINDINPUT CALLS STARTIMING
12540000
      IF SIGNEDON THEN FPB[4]:=FPB[4]+LOGINFO[24]+CLOCK+P(RTR);
12540500
      RDYTAPE;
12541000
    END;
12541500
    LOOKFORTAPE:=P;
12542000
  END;
12542500
*****12543000
REAL SUBROUTINE READTAPE;
12543500
BEGIN
12544000
RD:
12544500
  IF DSED OR PRT[P1MIX,@25]=5 THEN BEGIN P(5); GO TO RED END;
12545000
  IF WAITIO(B,@2000040,UNIT),[42:1] THEN
12545500
    BEGIN
12546000
      P(WAITIO(B,@2000040,UNIT),DEL);
12546500
      IF M[B INX 3] THEN
12547000
        IF LOOKFORTAPE THEN GO TO RD;
12547500
      P(3);
12548000
      GO TO RED;
12548500
    END;
12549000
    FOR T:=17 STEP 18 UNTIL 89 DO
12549500
      IF M[B INX T],[20:1] THEN T:=256;
12550000
    P(T>200);
12550500
RED:
12551000
  READTAPE:=P;
12551500
  END;
12552000
*****12552500
12553000

```

```

BOOLEAN SUBROUTINE SPACETOFILE;
BEGIN
  X:=NUM;
  WHILE (X:=X-1) GEQ 0 DO
  BEGIN
    DO UNTIL (T:=READTAPE);
    IF T GEQ 3 THEN BEGIN P(1); GO TO SPACEND END;
  END;
  P(0);
SPACEND:
  SPACETOFILE:=P;
END;
*****
BOOLEAN SUBROUTINE FINDFILE;
BEGIN
  IF (HEADER:=DIRECTORYSEARCH(MFID,"FID",SEARCHVAL)) LSS 64 THEN
  GO TO NOMORE;
  HEADER:=[M[HEADER]]&30[8:38:10];
  SEGNR:=0;
  CURROW:=10;
  IF ABORTED:=HEADER[5],[2:1] THEN
  IF HEADER[7]=0 THEN
  BEGIN
    P(1);
    GO TO NOFILE;
  END;
  LABELTABLE[V]:=NABS(FID);
  P(0);
NOFILE:
  FINDFILE:=P;
END;
*****
BOOLEAN SUBROUTINE NOMOREREELS;
BEGIN
  IF HEADER.[CF] GEQ 64 THEN FORGETSPACE(HEADER);
  IF FID.[REELNO]=0 THEN
  BEGIN HEADER:=0;
    P(1);
  END ELSE
  BEGIN
    STREAM(ONE:=1, F:=[FID]);
    BEGIN SI:=LOC ONE; DSI:=8 ADD END;
    P(FINDFILE);
  END;
  NOMOREREELS:=P;
END;
$ SET OMIT = NOT PACKETS
*****
BOOLEAN SUBROUTINE NOMOREFILES;
BEGIN
  IF NOT P(FID,[30:12]="99" OR COMMON,[NOTP].DUP) THEN
  BEGIN
    P(DEL);
    STREAM(ONE:=1, F:=[FID]);
    BEGIN SI:=LOC ONE; SI:=SI+6; DI:=DI+5;

```

```

12553500
12554000
12554500
12555000
12555500
12556000
12556500
12557000
12557500
12558000
12558500
12559000
12559500
12560000
12560500
12561000
12561500
12562000
12562500
12563000
12563500
12564000
12564500
12565000
12565500
12566000
12566500
12567000
12567500
12568000
12568500
12569000
12569500
12570000
12570500
12571000
12571500
12572000
12572500
12573000
12573500
12574000
12574500
12575000
12575500
12576000
12576500
12577000
12577500
12578000
12578500
12579000
12579500
12580000
12580500
12581000
12581500
12582000
12582500
12583000

```



```

                DS:=2 ADD; DS:=LIT"1";
                END;
                FIRSTFID:=FID;
                P(FINDFILE);
            END;
            NOMOREFILES:=P;
        END;
        $ POP OMIT
%*****
        SUBROUTINE REMOVEIT;
        BEGIN
            T:=DIRECTORYSEARCH(=MFID,=(FID:=PFIRSTFID),SEARCHVAL);
            IF T GEQ 64 THEN
        $ SET OMIT = NOT PACKETS
            DO BEGIN
        $ POP OMIT
                DO IF FID=IOD THEN GO AUT UNTIL NOMOREREELS;
        $ SET OMIT = NOT PACKETS
                END UNTIL NOMOREFILES;
        $ POP OMIT
        AUT:
            END;
%*****
        SUBROUTINE PAGEJECT;
        BEGIN
            P(WAITIO(@4000100000,0,V), DEL);
        END;
%*****
        SUBROUTINE WRITEB;
        BEGIN
            P(WAITIO(B INX @210104000000,0,V), DEL);
        END;
%*****
        SUBROUTINE SETUPINREC;
        BEGIN
            INREC:=[M[B INX (UNIT=18)]]&18[8:38:10];
            INREC[17]:=0;
        END;
%*****
        SUBROUTINE INVALIDNUM;
        BEGIN
            FILEMESS("INVALID","FILE ",0,"NUMB #",NUM+1,0,0);
        END;
%*****
        P(DEL,Z,MSCH,STF);
        GO TO SW[P];
%
% LOOKFORTAPE FINDS THE NEXT REEL. THE FIRST RECORD IS A LABEL SO

```

```

12583500
12584000
12584500
12585000
12585500
12586000
12586500
12587000
12587500
12588000
12588500
12589000
12589500
12590000
12590500
12591000
12591500
12592000
12592500
12593000
12593500
12594000
12594500
12595000
12595500
12596000
12596500
12597000
12597500
12600500
12601000
12601500
12602000
12602500
12603000
12603500
12607000
12607500
12613000
12613500
12614000
12614500
12615000
12615500
12616000
12616500
12617000
12617500
12618000
12618500
12618750
12619000
12619250
12619500
12620000
12620500
12621000
12621500
12621900
12621910

```

```

% INREC IS MOVED DOWN TO SKIP IT, 12621920
LOOK4TAPE: 12621930
P(LOOKFORTAPE); 12622000
INREC:=(NOT 17) INX INREC; 12622100
RECOUNT:=0; 12622500
GO RETURNTOCOM19; 12623000
12623500
NOMOREELS: 12624000
12624400
P(NOMOREREELS); 12624500
GO RETURNTOCOM19; 12624600
12625000
QTSPEC: 12625500
12625900
PRT[P1MIX,@25]:=0; 12626000
P(T); 12626100
IF UNIT#18 THEN % BE CAREFUL OF THIS, 12626250
% DISK PORTION 12626500
BEGIN NT2:=(T,[9:24] DIV 5)&T[1:2:1]); 12626750
IOD:=(HEADER[8] DIV 3)*3; % CALCULATE TRUE ROW SIZE 12627000
IF (T:=3*NT2+SEGNR) LSS 0 THEN % SPACE BACKWARD 12627500
DO IF (CURROW:=CURROW-1) LSS 10 THEN % TO A PREVIOUS FILE 12628000
BEGIN 12628500
IF FID=FIRSTFID THEN GO TO BOMBER; 12629000
IF SEARCHVAL=3 THEN P(DIRECTORYSEARCH(-MFID,FID,13),DEL); 12629500
FORGETSPACE(HEADER); 12630000
STREAM(ONE:=1, FI:=[FID]); 12630500
BEGIN SI:=LOC ONE; DS:=8 SUB END; 12631000
IF (HEADER:=DIRECTORYSEARCH(MFID,FID,5)) LSS 64 12631500
THEN GO BOMBER; 12632000
HEADER:=[M[HEADER]]&30[8:38:10]; 12632002
CURROW:=HEADER[9],[43:5]+9; 12632500
WHILE HEADER[CURROW]=0 DO CURROW:=CURROW-1; 12633000
IF CURROW<10 THEN 12633500
BEGIN 12634000
BOMBER: NT1:="RANGE +"; 12634500
IF (NT2:=PRT[P1MIX,@25]),[2:1] THEN 12635000
NT1:=NT1&"="[42:42:6]; 12635500
FILEMESS("INVALID","QT ",0, 12636000
NT1,NT2,[9:24],0,0); 12636500
PRT[P1MIX,@25]:=5; % FORCE A QT 12637000
P(DEL); % LEFT AT 12626500 12637500
GO RETURNFALSE; 12638000
END; 12638500
END UNTIL (T:=IOD+T) GEQ 0 12639000
ELSE % SPACE DISK FORWARD 12639500
BEGIN 12640000
IF T GEQ IOD THEN % TO ANOTHER ROW, 12640500
DO % CHECKING FOR NEW FILE 12641000
IF (CURROW:=CURROW+1) GEQ (HEADER[9],[43:5]+10) THEN 12641500
IF NOMOREREELS THEN GO TO BOMBER 12642000
UNTIL (T:=T+IOD) LSS IOD; 12642500
IF (CURROW-10)*IOD+T GTR HEADER[7]*3 THEN 12643000
GO TO NEXTFILE; 12643500
END; 12644000
SEGNR:=T; 12644500
P(19); 12645000
END ELSE % TAPE PORTION 12645500
BEGIN 12646000
NEXTFILE: 12646500

```



```

END;
RDCTABLE[V],[8:6]:=P1MIX;
STARTIMING(5,V);
STARTIMING(0,UNIT:=COMMON,[UNIT]);
FPB:=PRT[P1MIX,3];
COPY:=COMMON,[COPY];
IF UNIT=18 THEN
BEGIN
  MFID:=IF V=22 THEN "PUD " ELSE "PBD ";
  FIRSTFID:=LABELTABLE[V],[6:42];
$ SET OMIT = NOT PACKETS
  IF NOT COMMON,[NOTP] THEN BEGIN PCOPY:=COPY; COPY:=0 END;
  PFIRSTFID:=
$ POP OMIT
  FID:=FIRSTFID;
  SEARCHVAL:=3;
  IF FINDFILE THEN GO RETURNFALSE;
END ELSE
BEGIN
  ABORTED:=0;
  NOCONT:=((NUM:=COMMON,[NUM]) OR COPY)#0;
  MFID:=MULTITABLE[UNIT];
  IF LABELTABLE[UNIT],[1:5]#021 THEN % UNIT WAS CL-ED WHILE
  BEGIN ABORTED:=2; % WE WERE SCHEDULED,
  GO RETURNFALSE;
  END;
  FID:=LABELTABLE[UNIT]:=(*P(DUP))&0[5:47:1];
  RDCTABLE[UNIT],[8:6]:=P1MIX;
  RRRMECH:=RRRMECH OR TWO(UNIT);
  RDYTAPE;
  IF SPACETOFILE THEN
  BEGIN
    IF T=3 THEN INVALIDNUM; % SET BY READTAPE IF EOT,
    GO RETURNFALSE;
  END;
END;
SETUPINREC;
GO RETURNTRUE;

STARTANNEWFILE;

% HANDLES THE END OF A FILE AND FIGURES OUT WHAT TO DO NEXT, BUT
% FIRST, THE LOG MUST BE TAKEN CARE OF. (DONT USE T BETWEEN HERE AND
% THE TEST AT 12705750,)
%
IF ABORTED=2 THEN GO TO TAPECL;
IF SIGNEDON THEN
BEGIN
  SIGNEDON:=LOGINFO[16]+PROCTIME[P1MIX]+CLOCK+P(RTR);
  PROCTIME[P1MIX]:=(*P(DUP))-SIGNEDON;
  FORMTIME([LOGINFO[10]],SIGNEDON);
  FORMTIME([LOGINFO[11]],
    IOTIME[P1MIX]=(IOTIME[P1MIX]+LOGINFO[17]));
  FORMTIME([LOGINFO[12]],XCLOCK+P(RTR));
  SIGNEDON:=LOGINFO[18]+CLOCK+P(RTR);
  FPB[4]:=(*P(DUP))-SIGNEDON;
  FPB[9]:=(*P(DUP))-SIGNEDON;
  STREAM(DI:=DSED, CI:=P(DUP) OR COPY>1, LOGINFO);
  BEGIN $I:=LOGINFO; $I:=$I+40; $I:=$I+40;
    DI:=DI+32; DSI=LIT"="; DSI:=8 CHR;

```

```

12673000
12675000
12675250
12675500
12675750
12676000
12676500
12677000
12677500
12679500
12680000
12680250
12680500
12681000
12681500
12682000
12682500
12684000
12684500
12686000
12686500
12687000
12687300
12687400
12687500
12687600
12687700
12688000
12688500
12689000
12690500
12691000
12691500
12692000
12692500
12693000
12693500
12694000
12694400
12694500
12694600
12694610
12694620
12694630
12694640
12694800
12695000
12695500
12696000
12696500
12697000
12697500
12698000
12698500
12698750
12699000
12699250
12699500
12700000
12700500

```

```

DS:=6 LIT", IOT="; DS:=8 CHR; 12701000
DS:=11 LIT"; DONE AT "; DS:=5 CHR; DS:=LIT"+"; 12701500
C(DI:=DI-15; CI:=CI+D; GO TO CPY; 12702000
DS:=5 LIT"DS=ED"; JUMP OUT; 12702500
CPY: DS:=4 LIT"COPY"); 12703000
END; 12703500
SPOUTIT(LOGINFO,(EOJMESS OR PBEQJK) AND 12704000
NOT(JAR[P1MIX,9],[2:1])); 12704100
SIGNEDON:=0; 12704500
END; 12705000
% 12705100
% IF DSED OR QTED, SKIP THE CHECKS FOR COPIES, 12705110
% 12705120
IF (TERMFLAG:=DSED OR QTED*3) THEN 12705250
IF V=22 THEN GO TO PNCHDS ELSE GO TO PRNTDS; 12705500
% 12705600
% T IS SET IF THE FIRST GET FAILS, THIS SHOULD ONLY HAPPEN AT THE END 12705610
% OF A BACK-UP TAPE. NOTE THAT IF A FILE NUMBER IS SPECIFIED, INITIAL= 12705620
% IZE ONLY SPACES TO ITS START, SO WE MAY CATCH AN INVALID NUMBER 12705630
% HERE, SINCE ONLY ONE FILE IS PRINTED WHEN A NUMBER IS GIVEN, IF WE 12705640
% ARRIVE HERE, IT MUST HAVE BEEN A BAD NUMBER, IF IT IS DESIRED TO 12705650
% CONTINUE DOWN THE TAPE AFTER THE SPECIFIED FILE, THIS TEST WILL NEED 12705660
% TO BE CHANGED. 12705670
% 12705680
IF T THEN % FIRST GET FAILED 12705750
IF UNIT#18 THEN 12706000
BEGIN 12706250
IF COMMON,[NUMF]#0 THEN INVALIDNUM; 12706500
GO TO TAPEND; 12706750
END ELSE GO REMOVE; 12707000
% 12707100
IF (COPY:=COPY-1) GTR 0 THEN % MORE COPIES OF FILE RQD, 12707250
BEGIN 12707500
IF V=22 AND PUNCHLCK THEN 12707750
BEGIN 12708000
PNCHLK: STREAM(P1MIX, T:=T:=SPACE(10)); 12708250
BEGIN DS:=25 LIT"#PNCH LOCKED:PRNPBT/DISK="; 12708500
SI:=LOC P1MIX; DS:=2 DEC; DS:=LIT"+"; 12708750
DI:=DI-3; DS:=FILL; 12709000
END; 12709250
SPOUT(T); 12709500
REPLY[P1MIX]:=NABS(T:=VOK&VWY[36:42:6]&VQT[30:42:6]); 12709750
COMPLEXSLEEP(REPLY[P1MIX]>0 OR DSED); 12710000
IF NOT WHYSLEEP(T) THEN GO TO PNCHLK; 12710250
IF DSED OR QTED THEN GO STARTANWFIL; 12710500
END; 12710750
IF UNIT=18 THEN % DISK 12711000
BEGIN 12711250
$ SET OMIT = NOT PACKETS 12711500
STOG:=SEARCHVAL=3; 12712000
$ POP OMIT 12712500
PRINTITAGAIN; 12713000
FID:=FIRSTFID; 12713500
SEARCHVAL:=5; 12714000
IF FINDFILE THEN GO TO EOF ELSE GO TO CONTINUE; 12714500
END; 12715000
% 12715400
IF RDCTABLE[UNIT],[14:10]#1 THEN % TAPE 12715500
BEGIN % THIS ISNT FIRST REEL 12716000
RDCTABLE[UNIT],[14:10]:=0; 12716500

```

```

        IF NOT LOOKFORTAPE THEN GO TO EOF;
    END ELSE
        RDYTAPE;
    IF SPACETOFILE THEN GO TO EOF ELSE GO TO CONTINUE;
END;
$ SET OMIT = NOT PACKETS
    IF UNIT=18 THEN
        BEGIN
            IF STOG THEN BEGIN SEARCHVAL:=3; STOG:=0 END;
            IF NOMOREFILES THEN
                IF (PCOPY:=PCOPY-1) GTR 0 THEN
                    BEGIN
                        FIRSTFID:=PFIRSTFID;
                        GO PRINTITAGAIN;
                    END ELSE
                ELSE GO CONTINUE;
            END;
        $ POP OMIT
EOF:
%   AT THIS POINT, WE ARE THROUGH WITH THIS FILE OR PACKET. CLEAN UP
%   THE OUTPUT BEFORE COING ON.
%
    IF V#22 THEN
        BEGIN
            PAGEJECT;
PRNTDS:
            IF SEPARATE THEN
                BEGIN
                    STREAM(U:=UNIT=18, XI=P(DUP)=17, FI=FID, AI=TINU[V],
                        B);
                    BEGIN
                        U(SI:=LOC F; SI:=SI+1; DS:=4 CHR; DS:=LIT" ");
                        SI:=LOC A; SI:=SI+5; DS:=3 CHR;
                        DS:=3 LIT" --"; X(DS:=7 LIT"B5700--"); DS:=LIT" ";
                        SI:=SI-3; DS:=3 CHR;
                        U(DS:=LIT" "; SI:=LOC F; SI:=SI+1; DS:=4 CHR);
                    END;
                    FOR TI=1 STEP 1 UNTIL 5 DO WRITEB;
                    FOR TI=2 STEP 2 UNTIL SEPARATION DO
                        P(WAITIO(@4002000000,0,V), DEL);
                    FOR TI=1 STEP 1 UNTIL 5 DO WRITEB;
                    PAGEJECT;
                END;
            END;
PNCHDS:
            IF UNIT#18 THEN
                BEGIN
                    IF TERMFLAG OR NOCONT OR ABORTED THEN
TAPEND:
                        REWIND;
                        GO TO TEST;
                    END ELSE
                BEGIN
                    NUM:=NUM+1;
                    RECOUNT:=@77777;
                    SETUPINREC;
                CONTINUE;
                RETURNFALSE;

```

```

12717000
12717500
12718000
12718500
12719000
12719500
12720000
12720500
12721000
12721500
12722000
12722500
12723000
12723500
12724000
12724500
12725000
12725500
12725900
12726000
12726100
12726110
12726120
12726130
12726500
12727000
12727500
12728000
12728500
12729000
12729500
12730000
12730500
12731000
12731500
12732000
12732500
12733000
12733500
12734000
12737000
12737500
12738000
12738500
12739000
12739500
12740000
12740500
12741000
12741500
12742000
12742500
12743000
12743500
12744000
12744500
12745000
12745500
12746000
12746500

```

```

        P(0);
        GO RETURNTOCOM19;
    END;
END;
REMOVEM;
% DISK = CLOSE THE OPENED FILES AND, IF NOT QTED, REMOVE THEM.
%
    IOD:=IF SEARCHVAL=3 THEN FID ELSE NOT 0;
    SEARCHVAL:=13; REMOVEIT;
    FPB[4]:=(P(DUP))+CLOCK+P(RTR);
    IF TERMFLAG#3 THEN % NOT QT-ED
    BEGIN
        IOD:=NOT 0;
        SEARCHVAL:=7; REMOVEIT;
TEST: % FOR CONTINUATION FOR AUTOPRNT,
        IF AUTOPRINT AND NOT (FORMTOG OR TERMFLAG) AND
        (TWO(V) AND SAVEWORD)=0 THEN
            IF (COMMON:=PRINTORPUNCHWAIT(-V,-0))#0 THEN GO TO STOPTIME;
    END;
TAPECL;
    COMMON:=0;
    FORGETSPACE(B);
    SETNOTINUSE(V,FORMTOG);
STOPTIME;
    STOPTIMING(5,1023);
RETURNTRUE;
    P(1);
RETURNTOCOM19;
    P(0,RDS,1,SUB,0,XCH,CFX,STF);
END OF FIRST PRINTER BACKUP SPECIAL CASES PROCEDURE;
PROCEDURE PRNPBTSPECASE2(Z);
%
% THIS PROCEDURE HANDLES ADDITIONAL THINGS FOR COM19, VALUES OF Z ARE:
% 0 INITAILIZE LOGGING,
% 1 WRITE ABORT OR DSED MESSAGE AND CONSTRUCT ENDING LABEL,
% 2 HANDLE PARITY ON INPUT FILE,
%
VALUE Z; REAL Z;
BEGIN
    REAL RCW=+0, MSCW=-2, COMMON=-4;
    ARRAY INREC=+1[*];
    ARRAY FPB=INREC+1[*], LOGINFO=FPB+1[*], HEADER=LOGINFO+1[*];
    REAL UNIT=HEADER+1, V=UNIT+1, COPY=V+1, MFID=COPY+1, FID=MFID+1,
        IOD=FID+1, T=IOD+1, B=T+1;
    REAL SEARCHVAL=B+1, CURROW=SEARCHVAL+1, FIRSTFID=CURROW+1,
        SEGNR=FIRSTFID+1;
    REAL X=SEARCHVAL, NUM=CURROW, RECOUNT=SEGNR;
    BOOLEAN SIGNEDON=SEGNR+1, FORMTOG=SIGNEDON+1, ABORTED=FORMTOG+1;
    BOOLEAN NOCONT=FIRSTFID;
    $ SET OMIT = NOT PACKETS
    BOOLEAN STOG=ABORTED+1;
    REAL PCOPY=STOG+1, PFIRSTFID=PCOPY+1;
    $ SET OMIT = PACKETS
    REAL PFIRSTFID=FIRSTFID, PCOPY=COPY;
    $ POP OMIT OMIT

    LABEL SLEAP, WHY, EXITTOCOM19;
    LABEL SIGNIN, ABORTMSG, PARERR;

```

```

12747000
12747500
12748000
12748500
12748900
12749000
12749100
12749110
12749120
12749500
12750000
12750250
12750500
12751000
12751500
12752000
12752500
12753000
12753500
12755500
12756000
12756400
12756500
12757000
12757500
12757750
12758000
12758250
12758500
12759000
12759500
12760000
12800000
12800100
12800110
12800120
12800130
12800140
12800150
12800500
12801000
12801500
12802000
12802500
12803000
12803500
12804000
12804500
12805000
12805500
12806000
12806500
12807000
12807500
12809500
12810000
12811500
12812000
12812500
12813000

```

```

SWITCH SW := 12813500
SIGNIN, ABORTMSG, PARERR; 12814000
DEFINE DSED = (TERMSET(P1MIX))#, 12814500
QTED = (PRT[P1MIX,@25]#0)#, 12815000
VF = 4315#, 12815100
UNITF = 3815#, 12815200
COPYF = 3018#, 12815300
NUMF = 2218#, 12815400
NOTP = 2911#, 12815500
COPYO = 2111#; 12815600
***** 12820500
***** 12821000
SUBROUTINE FM; %% BUILD AND SPOUT FORMS MESSAGE %% 12822000
BEGIN 12822500
STREAM(U:=TINU[V], P1MIX, INREC, D:=T:=SPACE(10)); 12823000
BEGIN DS:=LIT"#"; 12823500
SI:=LOC U; SI:=SI+5; DS:=3 CHR; 12824000
DS:=20 LIT" FM RQD:PRNPBT/DISK="; DS:=2 DEC; 12824500
U:=DI; DI:=DI-2; DS:=FILL; DI:=U; 12825000
SI:=INREC; DS:=5 LIT" FOR "; 12825500
SI:=SI+1; DS:=7 CHR; DS:=LIT"/"; 12826000
SI:=SI+1; DS:=7 CHR; DS:=4 LIT" OF "; 12826500
SI:=SI+1; DS:=7 CHR; DS:=LIT"/"; 12827000
SI:=SI+1; DS:=7 CHR; 12827500
DS:=LIT"+"; 12828000
END; 12828500
SPOUT(T); 12829000
REPLY[P1MIX] := 12829500
NABS(T:=VOK&VWY[36:42:6]&VQT[30:42:6]&VFM[24:42:6]); 12830000
END FM SUBROUTIN; 12830500
***** 12831000
***** 12831500
SUBROUTINE BADFM; %BUILD AND SPOUT BAD FM MESSAGE % 12832000
BEGIN 12832500
STREAM(A:=TINUT, MX:=P1MIX, T:=T:=SPACE(10)); 12833000
BEGIN DS:=19 LIT"INVALID INPUT UNIT "; 12833500
SI:=LOC MX; DS:=2 DEC; DS:=2 LIT"FM"; 12834000
SI:=LOC A; SI:=SI+5; DS:=3 CHR; 12834500
DS:=LIT "+"; DI:=DI-8; DS:=FILL; 12835000
END; 12835500
SPOUT(T); 12836000
END BADFM SUBROUTIN; 12836500
***** 12837000
***** 12837500
SUBROUTINE WRITEBANDEJECT; 12838000
BEGIN 12838500
P(WAITIO(B INX @210104000000,0,V),DEL); 12839000
IF V#22 THEN P(WAITIO(@4000100000,0,V),DEL); 12843000
END; 12843500
***** 12844500
***** 12845000
***** 12845500
***** 12846000
***** 12846500
***** 12847000
***** 12847500
***** 12847600
***** 12848000
***** 12848100
SIGNINI

```



```

% HANDLES FIRST RECORD OF FILE, PICKING UP LOGGING INFO AS WELL AS 12848110
% COPIES OR FORM SPECIFICATIONS, NOTE THAT LABEL INFO IS SAVED IN 12848120
% LOGARRAY FOR USE AT ABORTMSG, TIMING IS STARTED AT INITAILIZE AND 12848130
% STOPPED IN REWIND, AT REMOVEM OR AT STOPTIME FOR TAPE, DISK AND THE 12848140
% OUTPUT UNIT RESPECTIVELY, LOGARRAY IS USED TO REMOVE THE TIME 12848150
% ASSOCIATED WITH A GIVEN BACK UP FILE FROM THE TIMING IN THE FPB AND 12848160
% LOG IT TO THE USER, THAT IS DONE IN SIGNOUT, THUS, THE TIME LOGGED 12848170
% AT PRNPBT/DISK EOJ IS OVERHEAD TIME OCURRING DURING SWITCHING FROM 12848180
% FILE TO FILE, 12848190
% 12848200
% 12848500
% 12849000
% 12849500
% 12850000
% 12850500
% 12851000
% 12851500
% 12852000
% 12852500
% 12853000
% 12853500
% 12854000
% 12854500
% 12855000
% 12860500
% 12861000
% 12861500
% 12862000
% 12862500
% 12863000
% 12863500
% 12864000
% 12864500
% 12865000
% 12865500
% 12866000
% 12866500
% 12873000
% 12873500
% 12874000
% 12874500
% 12875000
% 12875500
% 12876000
% 12876500
% 12877000
% 12877500
% 12878000
% 12878500
% 12879000
% 12879100
% 12879500
% 12879600
% 12879610
% 12879620
% 12879630
% 12879640
% 12880000
% 12880500
% 12881000

LOGINFO1=[M[GETSPACE(20,0,1)+2]]&20[8:38:10];
IF FORMTOG=[INREC[13] THEN FM;
IF COPY LEQ 0 AND NOT COMMON,[COPY0] THEN
COPY1=IF (INREC[14] AND NOT @377)=0 THEN INREC[14]+1 ELSE 0;
STREAM(INREC, LBL1=(NOT 14) INX INREC, LOGINFO1);
BEGIN SI=[INREC]; SI=[SI+17];
DS1=12 LIT" PRNPBT FOR ";
DS1=7 CHR; DS1=LIT"/"; SI=[SI+1]; DS1=7 CHR; DS1=5 LIT", PST";
SI=[INREC]; DS1=2 WDS; SI=[LBL]; DS1=10 WDS;
END;
LOGINFO[16]=-(PROCTIME[P1MIX]+CLOCK+P(RTR));
LOGINFO[17]= IOTIME[P1MIX];
LOGINFO[18]=CLOCK-P(RTR);
MAKELOG(INREC INX 3,PBCARD);
IF FORMTOG THEN
SLEAP:
BEGIN COMPLEXSLEEP(REPLY[P1MIX] GEQ 0 OR DSED OR QTED);
IF NOT WHYSLEEP(T) THEN
BEGIN FM; GO TO SLEAP END;
IF REPLY[P1MIX],[CF]=VFM THEN
IF (T=[REPLY[P1MIX],[FF]) NEQ 20 AND T NEQ 21 THEN
BEGIN % ILLEGAL UNIT,
LABELTABLE[T]=@114;
BADFM;
READY=[READY AND (T=[NOT TWO(T))];
RRRMECH=[RRRMECH AND T];
SAVEWORD=[SAVEWORD AND T]; FM; GO SLEAP
END ELSE
IF T#V THEN
BEGIN % SWITCH UNITS,
LABELTABLE[T] := LABELTABLE[V];
RDCTABLE[T] := RDCTABLE[V];
MULTITABLE[T] := MULTITABLE[V];
LABELTABLE[V] := MULTITABLE[V] := RDCTABLE[V] := 0;
FPB[8],[36:6]= (V=[T])+1;
END;
END;
FORMTOG=(FORMTOG OR PUNCHLCK AND V=22) AND NOT (DSED OR QTED);
SIGNEDON=[TRUE];
GO EXITTocom19;

ABORTMSG:
% ABORTED=3 IMPLIES ABORT HAS OCCURRED, CURRENTLY, NOTHING ATTEMPTS TO
% DISTINGUISH BETWEEN 1 AND 3, BUT ABORTED MUST BE SET HERE FOR TAPE
% SO WHY NOT MAKE IT DIFFERENT,
%
ABORTED=[3];
STREAM(T=[DSED OR QTED, B]);
BEGIN

```

```

DS:=8 LIT"#" ; SI:=8; DS:=16 WDS; DI:=8; 12881500
CI:=CI+T; GO TO AB; 12882000
DI:=DI+24; 12882500
DS:=34 LIT" BACK-UP TERMINATED BY OPERATOR "; 12883000
GO TO LEND; 12883500
AB; DI:=DI+34; DS:=11 LIT" ABORTED "; 12884000
LEND; 12884500
END; 12885000
WRITEBANDEJECT; 12885500
IF V#22 AND SIGNEDON THEN 12886000
BEGIN 12886500
  STREAM(LOGINFO, B); 12887000
  BEGIN SI:=LOGINFO; SI:=SI+32; 12887500
    DS:=8 LIT" LABEL "; DS:=12 WDS; DS:=LIT" "; 12888000
    SI:=LOGINFO; SI:=SI+12; DS:=15 CHR; 12888500
    DS:=12 LIT" "; 12889000
  END; 12889500
  WRITEBANDEJECT; 12890000
END; 12890500
GO TO EXITTocom19; 12891000

PARERR; 12891100
% 12891500
% BUILDS ERROR MESSAGE FOR OUTPUT AND ALLOWS OPERATOR TO OK OR DS, 12891600
% T IS USED TO PASS BACK WHETHER OR NOT TO TERMINATE. 12891620
% 12891630
IF V=22 THEN GO TO WHY; 12892000
STREAM(A:=UNIT, T:=T+SPACE(15)); 12892500
BEGIN 22(DS:=2 LIT ">>"); SI:=LOC A; SI:=SI+7; 12893000
  IF SC="B" THEN DS:=6 LIT " DISK " ELSE 12893500
  DS:=6 LIT " TAPE "; 12894000
  DS:=26 LIT "PARITY ON PRINTER BACK UP "; 12894500
  22(DS:=2 LIT ">>"); 12895000
END STREAM; 12895500
P(WAITIO(T&16[CTF],0,V),DEL); 12897500
FORGETSPACE(T); 12898000
WHY; 12898500
FILEMESS("#PARITY",0,0,"ERROR ",0,0,0); 12899000
REPLY[P1MIX]:=VQT&VWY[36:42:6]&VOK[30:42:6]; 12899500
COMPLEXSLEEP(REPLY[P1MIX] GEQ 0 OR DSED OR QTED); 12900000
IF NOT WHYSLEEP(VQT&VWY[36:42:6]&VOK[30:42:6]) THEN GO TO WHY; 12900500
T:=DSED OR QTED; 12901000
EXITTocom19; 12901500
P(0,RDS,0,XCH,CFX,STF); 12902000
END OF SECOND GROUP OF PRINTER BACKUP SPECIAL CASES; 12902500
PROCEDURE COM19; 13000000
% 13000100
% COM19, TOGETHER WITH PRNPBTSPECASE1 AND PRNPBTSPECASE2 WHICH SHARE 13000110
% ITS STACK, ARE THE WORKING PART OF PRINTER BACK-UP. INFORMATION IS 13000120
% PASSED TO COM19 IN COMMON AND LABELTABLE, AS FOLLOWS: 13000130
% COMMON,[43:5] LOGICAL UNIT NUMBER OF OUTPUT UNIT, 13000140
% [38:5] INPUT UNIT NUMBER, IF DISK, THE LABELTABLE ENTRY FOR 13000160
% THE OUTPUT UNIT CONTAINS THE FILE ID, 13000170
% [30:8] NUMBER OF COPIES SPECIFIED IN PB MESSAGE, 13000180
% [22:8] IF TAPE, STARTING FILE NUMBER GIVEN IN PB MESSAGE, 13000190
% IF DISK, =0 IF ENTIRE PACKET IS TO BE PRINTED, =1 IF 13000200
% NOT, 13000210
% [21:1] ON IF "#0" APPEARED IN PB MESSAGE, 13000215
% 13000250
BEGIN 13001000

```

REAL	RCW=+0, COMMON=-4;	13002000	
ARRAY	INREC[*], FPB[*], LOGINFO[*], HEADER[*];	13003000	
REAL	UNIT, V, COPY, MFID, FID, IOD, T, B;	13004000	
REAL	SEARCHVAL, CURROW, FIRSTFID, SEGNR;	13005000	
REAL	X=SEARCHVAL, NUM=CURROW, RECOUNT=SEGNR;	13006000	
BOOLEAN	SIGNEDON, FORMTOG, ABORTED;	13007000	
BOOLEAN	NOCONT=FIRSTFID;	13008000	
\$ SET OMIT	= NOT PACKETS	13009000	
BOOLEAN	STOG;	13010000	
REAL	PCOPY, PFIRSTFID;	13011000	
\$ SET OMIT	= PACKETS	13012000	
REAL	PFIRSTFID=FIRSTFID, PCOPY=COPY;	13013000	
\$ POP OMIT	OMIT	13014000	
%		13017100	
%	THE LOCAL VARIABLES ARE USED AS FOLLOWS:	13017110	
%	ARRAYS	13017120	
%	INREC	ARRAY DESCRIPTOR FOR THE CURRENT RECORD,	13017130
%	FPB	FPB ARRAY, INPUT IS THE FIRST FILE; OUTPUT THE 2ND,	13017140
%	LOGINFO	LABEL INFORMATION AND STARTING TIMES FOR LOGGING,	13017150
%	HEADER	DISK FILE HEADER,	13017180
%	REALS		13017190
%	UNIT	LOGICAL UNIT NUMBER FOR INPUT,	13017200
%	V	LOGICAL UNIT NUMBER FOR OUTPUT,	13017210
%	COPY	NUMBER OF COPIES OF THIS FILE TO BE PRINTED. IF IT IS	13017220
%		NOT SPECIFIED, IT EQUALS 0,	13017230
%	MFID	MULTI-FILE ID OF INPUT FILE,	13017240
%	FID	FILE ID OF INPUT FILE,	13017250
%	IOD, T	TEMPORARY STORAGE,	13017260
%	B	ADDRESS OF 90 WORD BUFFER FOR INPUT,	13017270
%	BOOLEANS		13017280
%	SIGNEDON	ON IF LOGGING IS INITIALIZED, THIS SHOULD BE OFF ONLY	13017290
%		FOR FILES WHICH DO NOT START AT THE BEGINING, E.G.,	13017300
%		WHEN A STARTING REEL IS SPECIFIED ON DISK,	13017310
%	FORMTOG	ON IF FORM IS SPECIFIED OR PNCHLOCK IS SET,	13017320
%	ABORTED	=1, DISK ABORTED BY H/L, CHECK IN GET TO FIND OUT WHERE,	13017330
%		=2, TERMINATION DUE TO CL OF INPUT TAPE WHILE SCHEDULED,	13017335
%		=3, TAPE ABORTED BY H/L, FOUND BY RECOUNT MISMATCH,	13017340
%			13017350
%	THE FOLLOWING APPLY ONLY TO DISK FILES:		13017360
%	SEARCHVAL	THIRD PARAMETER FOR DIRECTORYSEARCH, IT IS 3 OR 5 DURING	13017370
%		PRINTING, DEPENDING ON WHETHER IT IS THE FIRST COPY OR	13017380
%		NOT, AND 13 OR 7 DURING FILE TERMINATION,	13017390
%	CURROW	INDEX OF THE ROW CURRENTLY BEING PRINTED,	13017400
%	FIRSTFID	FILE ID OF FIRST REEL, USED FOR MULTIPLE COPIES OF	13017410
%		MULTI-REEL FILES,	13017420
%	SEGNR	NUMBER OF NEXT SEGMENT TO READ FROM THE CURRENT ROW,	13017430
%			13017440
%	THE FOLLOWING APPLY ONLY TO TAPES:		13017450
%	X	TEMPORARY STORAGE,	13017460
%	NUM	NUMBER OF CURRENT FILE ON TAPE, USED FOR COPIES,	13017470
%	RECOUNT	NUMBER OF RECORDS PRINTED IN THIS FILE, THIS IS CHECKED	13017480
%		AGAINST THE C-FIELD OF THE IO DESCRIPTORS IN THE FILE TO	13017490
%		SPOT ABORTS,	13017500
%	NOCONT	TRUE IF CONTINUATION FROM FILE TO FILE IS NOT ALLOWED,	13017510
%			13017520
%	THE FOLLOWING APPLY ONLY TO PACKETS:		13017530
%	PCOPY	NUMBER OF COPIES FROM PB MESSAGE, WHICH MAY APPLY TO THE	13017540
%		ENTIRE PACKET, "COPY" IS SET ONLY FROM LABEL EQUATION,	13017550
%	PFIRSTFID	FILE ID OF FIRST FILE IN THE PACKET, USED FOR COPIES OF	13017560
%		THE PACKET, FIRSTFID APPLIES TO INDIVIDUAL FILES WITHIN	13017570

```

%          THE PACKET AND IS USED FOR COPIES SPECIFIED VIA LABEL      13017580
%          EQUATION.                                                  13017590
%          SET DURING THE FIRST PRINTING OF THE PACKET IF ONE OF     13017600
%          THE FILES SPECIFIES MULTIPLE COPIES, IT IS USED TO       13017610
%          RESTORE THE VALUE OF 3 TO SEARCHVAL WHEN THE FILE IS      13017620
%          COMPLETED.                                               13017630
%                                                                      13017670
%          LABEL TRYNEXT, TAPERDR, TAPERD, TAPECHK, ABORT, NOGET,     13018000
%          START, RESTART, MAINLOOP, QUIT, TESTEND;                 13019000
%          DEFINE DSED = (TERMSET(P1MIX))#,                          13020000
%          QTED = (PRT[P1MIX,@25]#0)#;                               13021000
%          DEFINE LOOKFORTAPE = PRNPBTSPECASE1(0)#,                 13022000
%          NOMOREREELS = PRNPBTSPECASE1(1)#,                       13023000
%          QTSPEC = P(PRNPBTSPECASE1(2),DEL)#,                      13024000
%          INITIALIZE = PRNPBTSPECASE1(3)#,                        13025000
%          STARTANNEWFILE = PRNPBTSPECASE1(4)#,                   13026000
%          SIGNIN = PRNPBTSPECASE2(0)#,                            13027000
%          ABORTMSG = PRNPBTSPECASE2(1)#,                          13028000
%          PARERR = PRNPBTSPECASE2(2)#;                             13029000
%                                                                      13030000
%*****13031000
%          BOOLEAN SUBROUTINE GET;
%          BEGIN
%          IF INREC[17],[20:1] THEN GO TO NOGET;
%          IF (INREC!=(NOT 17) INX INREC),[CF] GEQ B,[CF] THEN
%          IF UNIT#18 THEN GO TO TAPECHK ELSE
%          ELSE % READ NEXT BLOCK
%          IF UNIT=18 THEN
%          BEGIN
%          IF SEGNR > HEADER[7]*3 THEN GO TRYNEXT; % END OF FILE
%          IF (SEGNR GEQ HEADER[8]-1) THEN
%          BEGIN % END OF ROW
%          IF (CURROW=(CURROW+1) GEQ HEADER[9],[43:5]+10 THEN
%          TRYNEXT: IF NOMOREREELS THEN GO TO NOGET;
%          SEGNR:=0;
%          END;
%          INREC:=90 INX INREC;
%          DISKIO(IOD,#B,90,HEADER[CURROW]+SEGNR);
%          SEGNR:=SEGNR+3;
%          SLEEP([IOD],IOMASK);
%          IF IOD,[28:1] THEN
%          BEGIN PARERR;
%          IF T THEN GO TO NOGET; % DSED OR QTED
%          END;
%          IF ABORTED THEN % TEST FOR BAD IO DESC.
%          IF (M[B INX 18],[6:42] EQV " ")#NOT 0 THEN
%          GO ABORT;
%          END ELSE
%          BEGIN % TAPE
%          TAPERDR: X:=0;
%          TAPERD: IF (IOD:=WAITIO(B,@2000040*UNIT)),[43:1] THEN
%          BEGIN PARERR;
%          IF T THEN GO TO NOGET; % DSED OR QTED
%          END;
%          IF IOD,[42:1] OR X THEN
%          BEGIN
%          IF (X=#NOT X) THEN GO TO TAPERD;
%          IF M[B INX 3] THEN
%          IF LOOKFORTAPE THEN GO TO TAPERDR ELSE GO NOGET;

```

END;	13071000
IF (X:=M[B INX NOT 0])#90 THEN	13072000
IF (X AND @7775)=16 THEN   % OLD FORMAT TAPE	13073000
BEGIN	13074000
INREC,[CF]:=B INX 1;	13075000
INREC[17]:=M[B]&0[20:20:7];	13076000
END ELSE GO TO NOGET	13077000
ELSE	13078000
BEGIN	13079000
INREC:=90 INX INREC;	13080000
IF (RECOUNT:=RECOUNT INX 1) # INREC[17],[CF] THEN	13081000
BEGIN	13082000
ABORTMSG;	13083000
P(0);	13084000
GO TO GOTTEN;	13085000
END;	13086000
END;	13087000
END;	13088000
P(1);	13089000
GOTTEN: GET:=P;	13090000
END;	13091000
%	13092000
% START IS USED FOR A NEW FILE (OR NEW PACKET), RESTART IS USED FOR	13092010
% A COPY (OR A NEW FILE WITHIN A PACKET),	13092020
	13092030
START:	13093000
IF COMMON#0 THEN GO TO INITIATE;	13094000
IF INITIALIZE THEN	13095000
BEGIN	13096000
RESTART: IF GET THEN	13097000
BEGIN	13098000
IF INREC[17],[1:11]=0 THEN SIGNIN;	13099000
IF UNIT#18 THEN RECOUNT:=INREC[17],[CF];	13101000
END ELSE                           % BAD FIRST BLOCK, SHOULD BE EOT	13102000
BEGIN P(1);	13103000
GO TO TESTEND;	13104000
END;	13105000
MAINLOOP:	13106000
IF PRTRW[P1MIX],[PSF] > 1 THEN STOPM;	13107000
IF (T:=PRT[P1MIX,@25])#0 OR DSED THEN	13108000
BEGIN	13109000
IF T<0 THEN                   % + OR = SPECIFIED	13110000
BEGIN	13111000
QTSPEC;	13112000
GO TO MAINLOOP;	13113000
END;	13114000
ABORTMSG;                   % DSED OR QTED	13115000
GO TO QUIT;	13116000
END;	13117000
IF GET THEN                   % VALID REC, WRITE IT & CONTINUE	13118000
BEGIN	13119000
P(WAITIO(INREC[17]&(INREC)[CTC]&8[21:42:6],0,V),DEL);	13128000
GO TO MAINLOOP;	13129000
END;	13130000
END;	13131000
QUIT:	13132000
P(0);	13133000
TESTEND:	13134000
T:=P;                       % T=1 IF FIRST GET FAILS, ELSE 0,	13135000
IF STARTANWFIL THEN GO TO START ELSE GO TO RESTART;	13136000

```

END OF PRINTING BACKUP TAPE AND DISK FILES;
PROCEDURE SPOSET(TYPE, BUFH);
VALUE TYPE, BUFH; REAL TYPE, BUFH;
BEGIN
REAL LINE, LAST, BUFF, UZER;
LABEL EXIT, OKED, CARRYON;
ARRAY INF[*];
BOOLEAN OK;
BUFF ← BUFH.[15:15]=1;
IF SYSDISKADR=0 THEN GO EXIT;
INF ← [M[SPACE(SYSDISKRL)]]&SYSDISKRL[B:38:10];
IF NOT REMOTE THEN GO EXIT;
IF TYPE.[1:1] THEN %THIS ALLOWS QUITTER TO RESET A SPO
BEGIN LINE:=BUFH;
BUFF:=SPACE(10);
GO TO CARRYON;
END;
STREAM(L←-1; B←BUFH; A←"SPO");
BEGIN
DI ← LOC A; DI ← DI+5;
SI ← B;
DD: IF SC=" " THEN BEGIN SI ← SI+1; GO DD END; B ← SI;
IF SC<"0" THEN
BEGIN
IF 3 SC=DC THEN
L ← TALLY;
GO EGRESS;
END;
IF SC>"9" THEN GO EGRESS;
SI ← SI+1; DI ← LOC L;
IF SC≤"9" THEN IF SC≥"0" THEN
BEGIN
SI ← B; DS ← 2 OCT;
END
ELSE
BEGIN
SI ← B; DS ← OCT;
END;
EGRESS: END STREAM;
IF (LINE ← P) < 0 THEN GO EXIT;
CARRYON:
LAST ← ABS(SPOWORD);
IF LINE=0 THEN %SWITCH THE SPO
BEGIN
IF LAST=0 THEN GO EXIT;
SPOWORD.[1:1] ← NOT TYPE;
SYSDISKIO(3, -0, INF);
INF[3] ← SPOWORD;
SYSDISKIO(0, -0, INF);
GO OKED;
END;
IF LINE GTR STATIONMAX THEN GO EXIT;
SYSDISKIO(1, -LINE, INF);
UZER ← INF[1];
SYSDISKIO(3, -0, INF);
IF TYPE THEN
IF NOT STABLE[LINE], DIALEDUP THEN
OK ← FALSE
ELSE
BEGIN

```

```

13137000
13200000
13201000
13202000
13203000
13204000
13205000
13206000
13207000
13207050
13207100
13208000
13208100
13208200
13208300
13208400
13208500
13209000
13210000
13211000
13212000
13213000
13214000
13215000
13216000
13217000
13218000
13219000
13220000
13221000
13222000
13223000
13224000
13225000
13226000
13227000
13228000
13229000
13230000
13231000
13231500
13232000
13234000
13235000
13236000
13237000
13238000
13239000
13240000
13241000
13242000
13243000
13244000
13244300
13244350
13245000
13246000
13247000
13248000
13249000

```

IF OK* (UZER=0) THEN	13250000
BEGIN	13250100
SPOWORD,[40:8]* LINE;	13251000
INF[3]* SPOWORD;	13252000
END	13252100
END	13253000
ELSE	13254000
IF LAST=LINE THEN	13255000
OK* NOT(SPOWORD* INF[3]* 0)	13256000
ELSE	13257000
OK* FALSE;	13258000
SYSDISKIO(0,=0,INF);	13259000
IF OK AND TYPE AND (LAST#0) THEN	13260000
BEGIN;	13261000
STREAM(L* LINE,T* LAST,B* BUFF);	13265000
BEGIN	13266000
DS:=8 LIT "STATION "; SI:= LOC L;	13267000
DS* 2 DEC; B* DI; DI* DI-2; DS* FILL; DI* B;	13268000
DS* 11 LIT " REPLACING "; DS* 2 DEC; B* DI;	13269000
DI* DI-2; DS* FILL; DI* B;	13270000
DS:=18 LIT " AS ALTERNATE SPO*";	13271000
END;	13272000
TwxOUT(BUFF,37,1,LAST);	13274000
END	13275000
ELSE IF OK THEN	13276000
OKED: STREAM(L* LINE,S* TYPE,[47:1],B* BUFF);	13277000
BEGIN	13278000
L(SI:= LOC L; DS:= 8 LIT "STATION "; DS:= 2 DEC;	13279000
B* DI; DI* DI-2; DS* FILL; DI* B; JUMP OUT TO D);	13280000
DS* 3 LIT "SPO";	13281000
D: S(DS* 4 LIT " SET"; JUMP OUT TO D1);	13282000
DS* 6 LIT " RESET";	13283000
D1: DS* 8 LIT " AS SPO*";	13284000
END;	13285000
EXIT: SPOUT(BUFF);	13286000
IF OK AND (LINE#0) THEN	13286100
BEGIN	13286150
IF TYPE THEN	13286200
BEGIN	13286250
SYSDISKIO(3,=LINE,INF);	13286300
INF[3]* (NOT 0),[12:36];	13286350
INF[1]* MCP;	13286400
SYSDISKIO(0,=LINE,INF);	13286450
\$ SET OMIT = TWXONLY	13286454
IF STABLE[LINE],STATIONTYPE THEN %SCREEN DEVICE	13286455
BEGIN TNAOG[SEQARRAY[LINE],[26:6],[14:34]]:=0;	13286460
IF STABLE[LINE],OUTPUTANKING THEN	13286465
IF TANKLINE[LINE] = 0 AND TAILOUT # LINE THEN	13286470
BEGIN TANKLINE[TAILOUT]:=LINE;	13286475
TAILOUT:=LINE;	13286480
STARTWORKING;	13286485
END; END;	13286490
\$ POP OMIT	13286491
LINE* LAST;	13286500
END;	13286550
IF LINE#0 THEN	13286600
BEGIN	13286650
SYSDISKIO(3,=LINE,INF);	13286700
INF[3]* INF[1]* 0;	13286750
SYSDISKIO(0,=LINE,INF);	13286800

```

$ SET OMIT = TWXONLY                                13286809
IF STABLE[LINE],STATIONTYPE THEN %SCREEN DEVICE    13286810
TNAOG[SEQARRAY[LINE],[26:6]]:=(*P(DUP))&INF[0][14:20:14]; 13286820
$ POP OMIT                                          13286821
END;                                               13286850
END;                                               13286900
IF SYSDISKADR NEQ 0 THEN                            13286950
FORGETSPACE(INF);                                  13287000
END SETSPO;                                        13288000
REAL PROCEDURE ANALYSIS;%                          14000000
BEGIN%                                             14001000
REAL ICW,IRCW,INCW,CL,T1,C,T2=SYLLABLE ;%         14002000
$ SET OMIT = NOT(NEWLOGGING)                       14002099
REAL MCPROCTEMP;                                   14002100
$ POP OMIT                                          14002101
LABEL GETOUT;%                                     14003000
COMMENT ANALYSIS EXAMINS THE SYLLABLE WHICH CAUSED THE INTURRUPT AND% 14004000
FROM THE RELATIVE ADDRESS OF THE SYLLABLE (INCLUDING% 14005000
VARIANT OPERATOR CONSIDERATIONS) COMPUTES THE LOCATION,C, 14006000
OF A COPY OF THE DESCRIPTOR ON THE TOP OF THE STACK,% 14007000
THE PREVIOUS TWO SYLLABLES ARE FETCHED BY THE STREAM% 14008000
STATEMENT GETSYLLABLES WHICH ALSO ADJUSTS THE C-L REGIST- 14009000
ERS PROPERLY,%                                     14010000
FINALLY THE STACK IS ADJUSTED AS FOLLOWS;%        14011000
DECREASE S BY 1,IF OPDC OR DESC%                  14012000
XCH A AND B REGISTERS,IF COC OR CDC%              14013000
OTHERWISE LEAVE THE SAME.                          ;14014000
CHECKSTACKSPACE;%                                  %WF 14014100
$ SET OMIT = NOT(NEWLOGGING)                       14014199
IF P1MIX>0 THEN                                    14014200
IF NOT LOGSTOPPED[P1MIX] THEN                     14014300
IF NOT MCPROCTIME[P1MIX],[1:1] THEN                14014400
BEGIN                                               14014500
MCPROCTEMP+PROCTIME[P1MIX]+CLOCK+P(RTR);          14014600
MCPROCTIME[P1MIX]+NABS(*P(DUP));                  14014700
END;                                               14014800
$ POP OMIT                                          14014801
INCW + PRT[P1MIX,8];%                               14015000
POLISH(,INCW,IOR);%                                14016000
IRCW + * INCW ;%                                    14017000
ICW + *( NOT 0 ) INX INCW);%                       14018000
CL + (IRCW INX 0) & IRCW[30:10:2];%                14019000
STREAM (T1+0,T2+0,CL+X+0);%                        14020000
BEGIN%                                             14021000
SI+CL; SI+SI=2 ; CL + SI; DI + LOC T2; DI+DI+6;%   14022000
DS + 2 CHR; SI + SI=3;%                            14023000
IF SC = "/" THEN%                                  14024000
BEGIN%                                              14025000
SI+SI=1; IF SC = "0" THEN%                         14026000
BEGIN TALLY+1; T1+TALLY ;CL + SI END;%            14027000
END;%                                              14028000
END GETSYLLABLE ;%                                 14029000
POLISH(,CL,+,T2,+,T1,+);%                          14030000
IF INCW,[32:1] THEN%                               14031000
BEGIN COMMENT P-BIT IN CHARACTER MODE ;%          14032000
IF T2 = @4441 THEN%                                14033000
BEGIN COMMENT ENTER CHARACTER MODE;%              14034000
P(M[(IRCW + *(NOT 0 INX INCW + PRT[P1MIX,8] +%     14035000
(NOT 1 INX INCW)&0[32:1:1]),[18:15]]&%           14036000
1[16:47:1]&0[18:18:15],(NOT 0)INX INCW,+); 14037000

```



```

                C ← INCW INX 0 =2;%          14038000
            END ELSE BEGIN%                14039000
        IF MEMORY[ C ← IRCW,[18:15]=T2.[36:6]], [113] = 4% 14040000
        THEN%                              14041000
        BEGIN%                              14042000
            IF T2,[42:6]= @53 THEN BEGIN% 14043000
        COMMENT CONTROL WORD MEANS CHARACTER MODE RELEASE;% 14044000
            T1←PRT[P1MIX,9]+M[(*(NOT 1)INX INCW)), [18:15]], [33:15]];% 14045000
            POLISH(MIT1],0,0)];%          14046000
        IF MIT1],[20:1] THEN CONTINUITYBIT;% 14047000
            PROGRAMRELEASE;%            14048000
            END%                          14049000
            END;%                          14050000
            IF T2 = 0 THEN GO TO GETOUT;% 14051000
            END%                          14052000
        END%                              14053000
        ELSE%                              14054000
        BEGIN%                              14055000
            IF T2,[46:1] THEN%           14056000
                BEGIN%                   14057000
                    C ← ICW,[33:15]];%    14058000
                    POLISH(ICW, (NOT 1)INX INCW, ←,IRCW,% 14059000
                    PRT[P1MIX,8]+INCW + (NOT 0)INX INCW ←,)];% 14060000
                    END OPDC DESC PART% 14061000
                ELSE%                    14062000
                    BEGIN%               14063000
                        C ← INCW INX 0 =2;% 14064000
                        IF (NT1 ← T2 AND @77) = @41 THEN% 14065000
                            BEGIN C ←C-1 ];% 14066000
                            POLISH(MEMORY[C],MEMORY[C+1],[MEMORY[C]], ← , [MEMORY[C+1]
                                ],←)];% 14067000
                            END COC CDC PART% 14068000
                            ELSE IF NT1 = @31 THEN% 14069000
                                BEGIN COMMENT THIS IS A BRANCH;% 14070000
                                GETOUT: CL ← P([PRT[P1MIX,1]],DUP,T2,XCH,←) INX @600000;% 14071000
                                END BRANCH PART% 14072000
                                ELSE IF NT1 = @35 THEN GO TO GETOUT; COMMENT RETURN;% 14073000
                                END ALL SYLLABLES BUT OPDC DESC ];% 14074000
                            END WORD MODE INTURRUPT ];% 14075000
                            POLISH(IRCW & CL[33:33:15]&CL[10:30:2],INCW,←) ];% 14076000
                            ANALYSIS ← C ];% 14077000
                            $ SET OMIT = NOT(NEWLOGGING) 14078000
                            IF MCPROCTEMP≠0 THEN 14078099
                                BEGIN 14078100
                                    MCPROCTEMP←PROCTIME[P1MIX]+CLOCK+P(RTR)-MCPROCTEMP; 14078200
                                    IF MCPROCTEMP<0 THEN MCPROCTEMP←0; 14078300
                                    MCPROCTIME[P1MIX]+ABS(←P(DUP))+MCPROCTEMP; 14078400
                                END; 14078500
                            $ POP OMIT 14078600
                            END ANALYSIS OF P BIT ]% 14078601
                            DEFINE CODEADDRESS(CODEADDRESS1, CODEADDRESS2) = 14079000
                                ACTUALOVERLAYADDRESS(1, CODEADDRESS1, CODEADDRESS2)#, 14100000
                                DATADDRESS(DATADDRESS1, DATADDRESS2) = 14101000
                                ACTUALOVERLAYADDRESS(0, DATADDRESS1, DATADDRESS2)#; 14102000
                                14103000
                            SAVE INTEGER PROCEDURE ACTUALOVERLAYADDRESS(TYPE, MIX, LOC); 14104000
                                VALUE TYPE, MIX, LOC; 14105000
                                INTEGER TYPE, MIX, LOC; 14106000
                                BEGIN INTEGER T = +1; 14107000
                                $ SET OMIT = NOT(AUXMEM) 14108000
                                14108999

```

LABEL	AUXMEM;	14109000
\$ POP OMIT	IF TYPE THEN % CODE,...	14109001
	BEGIN	14110000
\$ SET OMIT = NOT(AUXMEM)	IF LOC,[5:1] THEN GO TO AUXMEM;	14111000
		14111010
\$ POP OMIT	LOC := LOC INX 0;	14111020
	T := JAR[MIX,LOC DIV (T:=JAR[MIX,8])+10]+LOC MOD T;	14111021
	END ELSE % BETTER BE DATA,...	14112000
	BEGIN	14113000
\$ SET OMIT = NOT(AUXMEM)	IF LOC,[33:3]=7 THEN	14114000
AUXMEM:	T := -(0 & LOC[32:36:12]) ELSE	14115000
\$ POP OMIT	T := DALOC[MIX,LOC,[33:6]+P(DUP)=1]+LOC,[39:9]	14115010
	END;	14115020
END;		14116000
\$ SET OMIT = NOT(AUXMEM)		14116001
PROCEDURE AUXILIARYMEMORYCASUALTYRECOVERY;		14117000
FORWARD;		14118000
PROCEDURE AUXILIARYTABLEINITIALIZE;		14119000
FORWARD;		14119999
\$ POP OMIT		14120000
COMMENT	THE SEGMENT DICTIONARY IS CONSTRUCTED BY THE%	14121000
	COMPILERS AND EACH ENTRY HAS THE FORMAT: %	14122000
	[ 1: 1] = 1 FOR TYPE 2 SEGMENTS, = 0 OTHERWISE, %	14123000
	[ 2: 1] = 1 FOR INTRINSICS, = 0 OTHERWISE, %	14123001
	[ 3: 1] = 1 IF BEING MADE PRESENT, = 0 OTHERWISE	14125000
	(INTERLOCK FOR RE-ENTRANT CODE)	14126000
	[ 4: 2] = 0 FOR NORMAL SEGMENTS	14127000
	= 3 FOR SEGMENTS OVERLAID TO AUX, MEM,	14128000
	= 2 FOR SEGMENTS TO BE OVERLAID TO	14128100
	AUXILIARY MEMORY WHICH HAVEN'T BEEN	14128200
	[ 6: 1] = 1 FOR COBOL68 FILE TANK,	14128300
	[ 7: 1] = 1 FOR COBOL68 READ ONLY ARRAY,	14128400
	[ 8:10] = LINK TO PRT FOR 1ST DESCRIPTOR FOR%	14128500
	THIS SEGMENT, %	14128600
	[16:15] = SEGMENT SIZE (<1024) FOR ABSENT	14128700
	SEGMENTS, %	14128800
	= CORE ADDRESS OF PRESENT SEGMENTS, %	14129000
	= 1 FOR NEVER-PRESENT INTRINSICS, %	14130000
	[33:15] = DISK ADDRESS OF SEGMENT, %	14131000
	= INTRINSIC-NUMBER FOR INTRINSICS, %	14132000
	THE PRT FOR PROGRAM SEGMENTS IS CONSTRUCTED BY THE%	14133000
	COMPILERS IN THE FORMAT : %	14134000
	[ 0:5] = PROGRAM DESCRIPTOR BITS, %	14135000
	[ 6:1] = STOPPER BIT WHICH DEFINES THE [ 7:11] %	14136000
	FIELD, %	14137000
	[ 7:11] = LINK TO NEXT DESCRIPTOR THAT BELONGS TO %	14138000
	THIS SEGMENT, IF STOPPER FALSE, %	14139000
	= SEGMENT NUMBER, IF STOPPER TRUE, %	14140000
	[18:15] = R-REGISTER FIELD USED AT RUN TIME IN %	14141000
	LABEL AND ACCIDENTIAL DESCRIPTORS, %	14142000
	= SEGMENT NUMBER FOR WORD MODE AND %	14143000
	CHARACTER MODE DESCRIPTORS, %	14144000
	[33:15] = CORE ADDRESS FOR PRESENT SEGMENTS, %	14145000
	= RELATIVE ADDRESS FOR ABSENT SEGMENTS, %	14146000
	I.E. RELATIVE TO BEGINNING OF SEGMENT, %	14147000
	EACH PRT (R+4) CONTAINS A DESCRIPTOR WHICH POINTS %	14148000
		14149000
		14150000
		14151000
		14152000

```

                                TO THE SEGMENT DICTIONARY.%
;%
PROCEDURE MAKEPRESENT(C); VALUE C; REAL C;%
BEGIN%
  REAL MIXX=P1MIX;
  REAL SAVEBIT, MINE;%
  REAL P1MIX; REAL YECCHH=-2;%
  REAL D,MOTHER,MOM,LOC,SIZE;%
  INTEGER DISKADDR = SAVEBIT;%
  BOOLEAN REENRANT;
  DEFINE LINK= [ 7;11]#,STOPPER=[ 6; 1]#,PROGRAMDESC=[5;1]#;%
  DEFINE NOTOPEN =[25;1] #;%
  ARRAY NAME DD ;%
  ARRAY AIT[*];
  ARRAY PRTR[*] ;%
  REAL SEGNO=MOTHER, X=MOM,IOD ;%
  REAL SPACE;% SPACE FOR SEGMENT NUMBERS (INTRINSICS) BY MIX
  BOOLEAN NOT13=SPACE;
  REAL I,J;
  $ SET OMIT = NOT(NEWLOGGING)
  REAL MCPROCTEMP;
  $ POP OMIT
  LABEL EXIT; % ALL AVENUES MUST LEAD TO HERE
  LABEL WRAP,AROUND,TESTREADY;%
  LABEL OPEN,CLOSE;%
  LABEL CODEIN;
  LABEL DLOOP, NG;
  DEFINE REVERSE =[22;1]#,READY =[19;1]#,PRESENT =[2;1]#;%
  COMMENT MAKEPRESENT HAS THE FOLLOWING ACTIONS,DEPENDING ON THE TYPE%
  OF DESCRIPTOR CAUSING PRESENCE BIT :%
  DATA DESCRIPTOR :%
    IF MOTHER ABSENT THEN GET CORE SPACE AND SET%
      MOTHER PRESENT WITH PROPER CORE ADDRESS%
      THEN IF INITIAL ACCESS,ZERO THE SPACE ELSE%
      READ IN FROM DISK AND RETURN DISK SPACE%
      THEN SET 1ST MEMORY LINK TO SAVE OR NOT SAVE%
      AND SET 2ND LINK TO ADDRESS OF MOTHER%
    IN ANY EVENT, SET COPY PRESENT WITH CORRECT CORE%
      ADDRESS,%
    IO DESCRIPTOR;%
    PROGRAM DESCRIPTOR;%
    ;%
  SUBROUTINE RUNAROUND;%
  BEGIN WHILE NOT (PRTR[X] + ((LOC+2) INX PRTR[X]))%
    OR MEMORY),STOPPER DO X + PRTR[X],LINK;%
  END RUNAROUND;%
  $
  $ SET OMIT = NOT(NEWLOGGING)
  IF MIXX>0 THEN
    IF NOT LOGSTOPPED[MIXX] THEN
      IF NOT MCPROCTIME[MIXX],[1;1] THEN
        BEGIN
          MCPROCTEMP+PROCTIME[MIXX]+CLOCK+P(RTR);
          MCPROCTIME[MIXX]+NABS(*P(DUP));
        END;
  $ POP OMIT
  IF (P1MIX+YECCHH,[CF])=0 THEN P1MIX+MIXX;
  IF (D + MCJ),[1;1] THEN%
  IF D,[6;2]=1 THEN %TYPE 13 INTRINSIC
  BEGIN X:=[INTRNSC(SEGNO;=NFLAG(D) INX 0)];

```

```

14153000
14154000
14155000
14156000
14156500
14157000
14157500
14158000
14159000
14159100
14160000
14161000
14162000
14162500
14163000
14164000
14164100
14164200
14164300
14164399
14164400
14164401
14164500
14165000
14166000
14166100
14166200
14167000
14168000
14169000
14170000
14171000
14172000
14173000
14174000
14175000
14176000
14177000
14178000
14179000
14180000
14181000
14182000
14183000
14184000
14185000
14185100
14185199
14185200
14185300
14185400
14185500
14185600
14185700
14185800
14185801
14185900
14186000
14186010
14186020

```

```

IF MEMROW[P1MIX] INX 0 LSS FENCE THEN 14186030
BEGIN 14186040
SEGNO:=SEGNO-1; 14186050
STREAM(T:=SEGNO AND 3,I:=[INTABLE[P1MIX,SEGNO DIV 4]]); 14186060
BEGIN DI:=DI+T; DI:=DI+T; SKIP 1 DB; DS:=SET; END; %MARK TYPE18 14186070
IF X GTR 0 THEN SLEEP([X],-0); 14186080
IF (X INX 0) LEQ 1023 THEN 14186090
BEGIN P(ABS(X),[X],1=); 14186100
SIZE:=X INX 0; 14186110
MINE:=(NFLAG(D) INX 0)&SIZE[8:38:10]&3[1:46:2]; 14186120
$ SET OMIT = MONITOR OR NOT(AUXMEM) 14186121
AUXTRACE(1,(NFLAG(D) INX 0)); 14186122
$ POP OMIT 14186123
LOC:=GETSPACE(SIZE,13,0); 14186130
$ SET OMIT = NOT(AUXMEM) 14186132
IF X,[3:1] THEN %INTRINSIC ON AUXMEM 14186134
DISKADDR := -(O&X[32:21:12]); 14186136
ELSE 14186137
$ POP OMIT 14186138
DISKADDR := X,[6:27]; 14186139
GO TO CODEIN; 14186140
END ELSE BEGIN M[C],[CF]:=INTRNSC[(NFLAG(D)) INX 0],[CF]; 14186150
M[C],[2:1]:=1; 14186160
GO EXIT; 14186170
END 14186180
END ELSE 14186190
BEGIN 14186200
SIZE:=INTABLEROW[P1MIX],[8:10]-1; 14186210
FOR I:=INT13START STEP 1 UNTIL SIZE DO 14186220
IF INTABLE[P1MIX,I],[FF]=SEGNO THEN 14186230
BEGIN J:=I; SIZE:=0;END; 14186240
IF J NEQ 0 THEN 14186250
BEGIN 14186260
M[C],[CF]:=INTABLE[P1MIX,J],[CF]; 14186270
M[C],[2:1]:=1; 14186280
GO EXIT; 14186290
END ELSE 14186300
BEGIN %SEARCH FOR EMPTY SLOT 14186310
FOR I:=INT13START STEP 1 UNTIL SIZE DO 14186320
IF INTABLE[P1MIX,I]=0 THEN BEGIN J:=I; SIZE:=0; END; 14186330
IF J=0 THEN %EXPAND INTABLE ROW 14186340
BEGIN LOC:=INTABLEROW[P1MIX] INX 0; 14186350
INTABLEROW[P1MIX]:=[M[GETSPACE(SIZE+INT13SIZE,1,1) 14186360
+2]]&(SIZE+INT13SIZE)[8:38:10]; 14186365
MOVE(SIZE+INT13SIZE,INTABLEROW[P1MIX],[CF]=1, 14186370
INTABLEROW[P1MIX]); 14186375
MOVE(SIZE,LOC,INTABLEROW[P1MIX]); 14186380
J:=I; 14186390
END; 14186400
IF(SIZE:=INTRNSC[NFLAG(D) INX 0] INX 0) GTR 1023 THEN 14186410
SIZE:=M[SIZE-1],[FF]; 14186415
$ SET OMIT = NOT(AUXMEM) 14186416
IF X,[3:1] THEN %INTRINSIC ON AUXMEM 14186417
DISKADDR+-(O&X[32:21:12]) 14186418
ELSE 14186419
$ POP OMIT 14186420
DISKADDR:=X,[6:27]; 14186422
MINE+-(NFLAG(D) INX 0)&SIZE[8:38:10]&1[4:47:1]&1[2:47:1]; 14186425
LOC:=GETSPACE(SIZE,13,0); 14186430
M[LOC]:=(*P(DUP))&P1MIX[9:42:6]; 14186440

```

INTABLE[P1MIX,JJ]:=0&SEGN0[CTF]&(LOC+2)[CTC];	14186450
GO TO CODEIN;	14186460
END	14186470
END	14186475
END ELSE	14186480
BEGIN PRTR ← PRT[P1MIX,*]; LOC ← NFLAG(D)&0[5:5:1];	14187000
DO IF LOC.PROGRAMDESC THEN SEGN0 ← LOC,[18:15];	14188000
ELSE IF LOC.STOPPER THEN SEGN0 ← LOC,LINK;	14189000
ELSE LOC ← NFLAG(PRTR[LOC,LINK]);	14190000
UNTIL SEGN0#0;	14191000
DD ← SEGN0 INX PRTR[4];	14192000
IF DD[0],[3:1] THEN COMPLEXSLEEP((NOT DD[0],[3:1]));	14193000
IF (SIZE + (MINE + DD[0]),[18:15]) ≤ 1023 THEN	14194000
BEGIN DD[0],[3:1] ← 1;	14195000
NOT13:=TRUE;	14195100
IF MINE<0 THEN	14196000
IF PRTR[X ← MINE,[8:10]],[2:1] THEN GO AROUND;	14197000
IF MINE,[2:1] THEN	14198000
BEGIN X ← [INTRNSC[MINE INX 0]];	14198100
IF REENRANT:=(X,[4:1] OR (MEMROW[P1MIX] INX 0)	14198200
LSS FENCE) OR ((MINE INX 0)=17) THEN	14198210
BEGIN IF X>0 THEN SLEEP([X], -0);	14198300
IF (X INX 0)>1023 THEN	14198400
BEGIN LOC ← (SIZE + X INX 0)-2;	14198500
DD[0],[FF] ← SIZE; GO AROUND;	14198600
END ELSE P(ABS(X), [X], -);	14198700
END;	14199000
SIZE := X INX 0;	14200000
\$ SET OMIT = MONITOR OR NOT(AUXMEM)	14200010
AUXTRACE(1,(MINE INX 0));	14200020
\$ POP OMIT	14200030
\$ SET OMIT = NOT(AUXMEM)	14200100
IF X,[3:1] THEN % INTRINSIC ON AUXMEM	14200200
DISKADDR := -(0&X[32:21:12]);	14200300
ELSE	14200400
\$ POP OMIT	14200401
DISKADDR := X,[6:27];	14200500
END ELSE IF JAR[P1MIX,10]=0 THEN	14201000
DISKADDR := DATADDRESS(P1MIX, MINE)	14202000
ELSE DISKADDR := CODEADDRESS(P1MIX, MINE);	14202100
LOC ← GETSPACE(SIZE,1 + (MINE<0) + 70×REENRANT,	14203000
MINE<0 AND MINE,[6:1]);	14204000
CODEIN;	14205000
\$ SET OMIT = NOT(STATISTICS)	14205001
COUNTUP((LOC GTR FENCE)+19,(SIZE+29) DIV 30);	14205099
\$ POP OMIT	14205100
DISKIO(IOD,-LOC-1,SIZE,DISKADDR);	14205101
SLEEP([IOD],IOMASK);	14205200
IF IOD,[26:7] NEQ 0 THEN	14205300
BEGIN	14205400
IF MINE,[2:1] OR (D,[6:2]=1) THEN	14205500
INTRNSC[MINE INX 0] := NABS(*P(DUP));	14205700
DD[0],[3:1] := 0;	14205750
GO TO NG;	14205800
END;	14205850
X := MINE,[8:10];	14206000
IF NOT13 THEN	14206010
OLAYCTR[P1MIX]:=DOWNLAY(SIZE);	14206100
	%R3814206200

```

$ SET OMIT = NOT(STATISTICS)
CODEPBITS[P1MIX]:=*P(DUP)+1;
$ POP OMIT
  IF D.[6:2]=1 THEN
  BEGIN
  M[C],[CF]+LOC+2; M[C],[2:1]+1;
  IF MEMROW[P1MIX] INX 0 LSS FENCE THEN BEGIN
  INTRNSC[MINE INX 0]:=-( *P(DUP) )&(LOC+2)[CTC];
  M[LOC]:=(*P(DUP))&0[9:9:6]&0[2:47:1] END ELSE
  M[LOC],[2:1]=0;
  M[LOC+1]+      0&MINE[8:38:10]&SIZE[CTF];
  GO EXIT;
  END;
  IF MINE>0 THEN BEGIN RUNAROUND;%
  M[C] + ((LOC+2) INX D) OR MEMORY;%
  IF REENRANT THEN
  INTRNSC[MINE,[CF]] + -( *P(DUP) )&(LOC+2)[CTC] ELSE
  IF (X + PRTR[4],[18:6])≠0 THEN%
  M[LOC] + (*P(DUP))&X[9:42:6];%
$ SET OMIT = NOT(AUXMEM)
  IF NOT DISKADDR,[1:1] THEN
$ POP OMIT
  M[LOC+1]=0 & SIZE[CTF];
  M[LOC+1]=(*P(DUP))&SEGNO[CTC];
  IF MINE,[2:1] THEN M[LOC+1] + (*P(DUP))&MINE[8:38:10];%
  DD[0],[18:15] + LOC+2;%
  END PROGRAM CODE SEGMENTS%
ELSE BEGIN
  M[C] + PRTR[X] + M OR ((LOC+2)%
  &(M[LOC+1] + [PRTR[X] INX 0])[18:33:15]%
  &(MINE,[7:1]*24) [3:43:5] % COBOL68 READ ONLY
  &SIZE[8:38:10]);%
  IF MINE,[6:1] THEN % COBOL68 FILE TANK
  IF NOT P(M[LOC+4],TOP,XCH,DEL) THEN% BUILD FIB PTR
  BEGIN
  P([M[LOC+4]],DUP,DUP,LOD,XCH,INX,M[C],FFX,
  @100026,DIA 32,DIB 2,TRB 16,XCH,+);
  WHILE (AIT+PRTR[AITNDX]),PBIT=0
  DO MAKEPRESENT([PRTR[AITNDX] INX 0);
  IF AIT,[8:10] < AIT[0]+2 THEN
  BEGIN P(AIT,0,0); INTERRUPT(1);% PHONEY INVALID
  P(DEL,DEL,DEL); % INDEX ON AIT
  AIT + PRTR[AITNDX];
  END;
  AIT[AIT[0]:=*P(DUP)+1];=-(1&PRT[P1MIX,16]
  [8:38:10]&M[C][FTF]);
  END;
$ SET OMIT = NOT(STATISTICS)
DATAPBITS[P1MIX]:=*P(DUP)+1;
$ POP OMIT
  END TYPE TWO DATA SEGMENTS;%
  IF NOT MINE,[6:1] THEN M[LOC],[2:1] + 0;
  END ABSENT SEGMENTS%
  ELSE BEGIN LOC + SIZE-2;%
  REENRANT + DD[0],[2:1];
  AROUND: IF DD[0]>0 THEN%
  IF NOT PRTR[X + DD[0],[8:10]],[2:1] THEN RUNAROUND;%
  M[C] + IF DD[0]>0 THEN ((SIZE INX D) OR M)%
  ELSE PRTR[DD[0],[8:10)];%
  END;%

```

```

14206299
%R6314206300
14206301
14206310
14206320
14206330
14206332
14206334
14206336
14206338
14206340
14206350
14206360
14207000
14208000
14209100
14209200
14210000
14211000
14212000
14212010
14212011
14212020
14212030
14212100
14213000
14214000
14215000
14216000
14217000
14217500
14218000
14218010
14218025
14218027
14218030
14218035
14218040
14218045
14218050
14218055
14218060
14218065
14218070
14218075
14218080
14218090
14218099
%R6314218100
14218301
14219000
14220000
14221000
14222000
14222100
14223000
14224000
14225000
14226000
14227000

```

```

IF REENTRANT THEN
BEGIN SIZE * (DD[0] INX 0)=1;%
IF MEMROW[P1MIX] INX 0 GEQ FENCE THEN
SIZE:=REENTRANTINTABLEMAP(SIZE+1);
STREAM(SEGNO, T * SIZE AND 3,%
I * [INTABLE[P1MIX,SIZE DIV 4]]);%
BEGIN
SII=1; SII=SI+T; SII=SI+T; SKIP 1 SB;
IF SB THEN; % REMEMBER TYPE 13 REFERENCE
DI:=DI+T; DI:=DI+T; T:=DI; SII=LOC T;
SII=SI-2; DS:=2 CHR;
IF TOGGLE THEN BEGIN DI:=T; SKIP 1 DB; DS:=SET; END;
END;
END;%
DD[0],[3:1] * 0; GO EXIT;
END;%
IF (MOM:=D,[3:5])#0 AND (MOM AND @33)#@30 THEN
BEGIN%
COMMENT I/O DESCRIPTOR;%
IF JAR[P1MIX,2] < 0 THEN
BEGIN TERMINATE(P1MIX);
TERMINALMESSAGE(25);
END;
TESTREADY: MOM:=M[LOC+D INX (IF D,REVERSE THEN 2 ELSE NOT 1)],[CF];
IF (M[MOM] AND IOMASK)=0 THEN
WAITORSWAP(M[LOC],[12:6],MOM);
IF TERMSET(P1MIX) THEN GO TO INITIATE;
IF MEMORY[MOM],PRESENT THEN%
MEMORY[C]+MEMORY[MOM]%
ELSE%
BEGIN%
IF MEMORY[MOM],NOTOPEN THEN%
OPEN;%
BEGIN SAVEOPEN(MOM); GO TO TESTREADY END
ELSE BEGIN%
COMMENT READY AND NOT PRESENT INDICATES REEL-SWITCH OR TERMINATE;%
PRTR+M[MOM-3];%
LOC+PRTR[15],[25:5];%
SIZE+PRTR[4],[8:4];%
IF M[MOM],[27:1] THEN%
IF M[MOM],[24:1] THEN%
BEGIN IF SIZE=2 AND NOT PRTR[4],[2:1]%
AND NOT M[MOM],[22:1] THEN%
BEGIN BLASTQ(LOC);%
P(WAITIO(M[MOM-2],0,LOC),DEL);%
P(WAITIO(@1000000340000005,0,LOC),DEL);%
IF M[MOM-2] INX 4],[42:6]=1 THEN%
CLOSE: BEGIN LOC+PRTR[13],[28:10];%
FILECLOSE(MOM&@12[18:33:15]);%
PRTR[13],[28:10]+LOC+1;%
GO TO OPEN;%
END;%
END;%
END ELSE%
BEGIN IF SIZE=2 OR SIZE=7 OR SIZE=8 THEN%
BEGIN IF NOT PRTR[4],[2:1] THEN%
M[MOM-2] INX 4],[42:6]+1;%
GO TO CLOSE;%
END;%
END;%
END;%

```

```

14227100
14227200
14227210
14227220
14227300
14227400
14227500
14227520
14227540
14227560
14227580
14227600
14227620
14227700
14228000
14229000
14230000
14231000
14232000
14233000
14233100
14233200
14233300
14234000
14235000
14236000
14237000
14238000
14239000
14240000
14241000
14242000
14243000
%R9014244000
14245000
14246000
14247000
14248000
14249000
14250000
14251000
14252000
14253000
14254000
14255000
14256000
14257000
14258000
14259000
14260000
14261000
14262000
14263000
14264000
14265000
14266000
14267000
14268000
14269000
14270000

```





\$ SET OMIT = NOT(NEWLOGGING)	14301199
IF MCPROCTEMP#0 THEN	14301200
BEGIN	14301300
MCPROCTEMP+PROCTIME[MIXX]+CLOCK+P(RTR)=MCPROCTEMP;	14301400
IF MCPROCTEMP<0 THEN MCPROCTEMP=0;	14301500
MCPROCTIME[MIXX]+ABS(*P(DUP))+MCPROCTEMP;	14301600
END;	14301700
\$ POP OMIT	14301701
END MAKEPRESENT ;%	14302000
REAL ADDR\$=NT1;%	14342000
PROCEDURE COM5;%	14343000
BEGIN%	14344000
DEFINE HARRYSTA = M[WORKERSTACK+5];%    % STA	14344100
CLEARINPUTANK =	14344200
STATABLE[I]=(*P(DUP))&(CANDEMIX[I]+@240)[1:39:9]	14344250
&0[14:47:1]&1[16:47:1]	14344350
&(NOT (T:=INPUTANK[I]))[13:35:1];	14344400
INPUTANK[I]=NABS(P(DUP,LOD,0,XCH)&P[35:35:1]);	14344450
IF (S:=T:=T,[FF]=2) GTR 0 THEN DO	14344500
BEGIN IF (M[T] AND IOMASK)=0 THEN	14344550
SLEEP([M[T]],IOMASK);               % THEN DISCARD	14344600
FORGETSPACE(T);                   % BUFFERS	14344700
END UNTIL (T:=M[T+2],[CF]=2) LEQ 0 OR T=S;	14344800
#;	14344900
REAL RCW=+0,%	14345000
ERTOG=+2,%	14346000
I =+3,%	14347000
T =+4;%	14348000
INTEGER J=I;%	14349000
ARRAY VECTOR=+5[*],S=+6[*];%	14350000
INTEGER Q=S;	14350100
REAL MSCW=-1;	14350200
REAL TYP=+7;	14351000
INTEGER LINK;	14351100
LABEL RETURNEM,AHEAD;	14351150
INTEGER MOTHER=+8, NEXTMOM=+9, MOMMIX=+10, CATCH=+11;%	14351200
REAL JAR9 = +12;	14351210
\$ SET OMIT = NOT(PACKETS)	14351239
REAL UNITNO=LINK;	14351240
\$ POP OMIT	14351241
ARRAY PRTR=LINK[*];	%R2314351300
\$ SET OMIT = NOT(STATISTICS)	14351309
REAL OBJINFO=+12;	14351310
\$ POP OMIT	14351311
\$ SET OMIT = AUXMEM	14351390
DEFINE STACKSZ=180#;	14351400
\$ SET OMIT = NOT AUXMEM	14351410
DEFINE STACKSZ=200#;	14351420
\$ POP OMIT OMIT	14351430
PRIORITY+PRYOR[P1MIX]*=-1;	14351500
P(GETSPACE(STACKSZ,76,0),STS,.COM5,RCW,0,RDS,0,XCH,CFX,STF);	14353000
P(P&[MSCW][CTF],0,0,0,0,0,0);	14354000
P(0,0,0,0,0,0,0);	14355000
\$ SET OMIT = NOT(STATISTICS)	14355099
P(0);	14355100
CORETIME[P1MIX]=(*P(DUP))+CLOCK+P(RTR)-TIMING[P1MIX];	14355200
\$ POP OMIT	14355201
UVROW[P1MIX]+(VECTOR+UVROW[P1MIX])&	14356300
(GETSPACE(UVSIZE,64,5)+2)[CTC];	14356400
MOVE(UVSIZE,VECTOR,UVROW[P1MIX]);	14356500

VECTOR:=JARROW[P1MIX]&(GETSPACE(30,64,5)+2)[CTC];	14357000
MOVE(30,JARROW[P1MIX],VECTOR);	14357100
IF (MOTHER:=MEMROW[P1MIX],[CF] LSS FENCE) THEN	14357150
FORGETSPACE(JARROW[P1MIX]) ELSE	14357200
IF VECTOR[1] LSS 0 THEN MEMROW[P1MIX]:=MEMROW[0];	14357250
JARROW[P1MIX],[CF]:=VECTOR;	14357300
ERTOG+0&PRT[P1MIX,@25][1:2:46];	14357400
JAR9 := VECTOR[9];	14357500
TYP+VECTOR[2],[8:10];	14358000
IF VECTOR[0]<0 THEN%	14358100
BEGIN CATCH+PRT[P1MIX,@26];	14358150
ERTOG+VECTOR[1]<0 OR ERTOG;	14358200
END;	14358300
IF VECTOR[2]<0 THEN % COBOL	14360100
IF VECTOR[1]>0 THEN % NOT DS=ED	14360200
WHILE PRT[P1MIX,16]>0 DO ASR;%CLEAN OUT AIT	14360300
\$ SET OMIT = NOT(AUXMEM)	14360302
IF VECTOR[1] GTR 0 THEN % NOT DS=ED	14360304
FOR MOMMIX := 6 STEP 5 UNTIL 11 DO	14360310
BEGIN	14360320
Q := NFLAG(PRT[P1MIX,MOMMIX]); % AIT OR OAT ENTRY	14360322
IF Q.[2:1] THEN % PRESENT, GRAB ADDR FROM LINK	14360324
Q := Q & M[Q INX NOT 0][FTC];	14360326
IF Q.[33:3]=7 THEN ARTN(Q,-1); % AUXMEM	14360328
IF VECTOR[2] LSS 0 THEN MOMMIX:=11; % COBOL HAS NO OAT	14360330
END;	14360332
\$ POP OMIT	14360333
SLEEP([OLAYMASK],T:=TWO(P1MIX));	14360334
OLAYMASK := OLAYMASK AND NOT T;	14360336
T := DALOC[P1MIX,0],[CF];	14360338
FOR I:=1 STEP 2 UNTIL T DO	14360340
BEGIN	14360342
FORGETUSERDISK(DALOC[P1MIX,I],-500);	14360360
END;	14360362
J := INTABLEROW[P1MIX] := 0;	14360364
\$ SET OMIT = NOT(AUXMEM)	14360366
S := PRT[P1MIX,4] & ((I:=*P(DUP))+1)[8:38:10];	14360368
FOR I:=1 STEP -1 UNTIL 1 DO	14360370
IF (AUXCODE[P1MIX]+AUXDATA[P1MIX])=0 THEN I:=1 ELSE	14360372
IF (NT1 := S[I])<0 THEN	14360374
IF VECTOR[1] GTR 0 THEN % AVOID CONFUSION IF DS=ED	14360376
BEGIN COMMENT TYPE=TWO (DATA) SEGMENT;	14360378
IF (NT1:=NFLAG(PRT[P1MIX,NT1],[8:10])),[2:1] THEN	14360380
NT1 := NT1 & M[P(DUP) INX NOT 0][FTC];	14360382
IF NT1,[33:3]=7 THEN	14360384
ARTN(NT1,-1);	14360386
END ELSE ELSE	14360388
IF NT1,[5:1] THEN	14360390
BEGIN IF (NT2 := NT1,[FF])>1023 THEN NT2:=M[NT2-1],[FF];	14360392
AUXCODE[P1MIX] := *P(DUP)-NT2,[38:6]-1;	14360394
FORGETAUXILIARYSPACE(NT2, NT1,[CF]);	14360396
END;	14360398
\$ POP OMIT	14360399
IF MOTHER THEN % BELOW FENCE	14360400
BEGIN STASUS[P1MIX]+STABLE;	14360500
CORE,[FF]+CORE,[FF]=SINFO[P1MIX],[FF];	14360600
WAITSTORE(P1MIX);	14361000
WHILE(T+M[I]),[CF] ≠ 0 DO%	14362000
BEGIN%	14363000
IF T>0 THEN%	14364000

```

%
%
IF T,[9:6]=P1MIX THEN% 14365000
%R8114365100
%R8114365200
FORGETSPACE(I INX 2); %R8114365300
I+ T,[CF]% 14366000
END;% 14367000
END ELSE 14367010
BEGIN IF LOGLINE,[33:7]≠0 THEN 14367020
BEGIN CLEAR TANK(LOGLINE,0); 14367021
TANKS[I:=LOGLINE,[40:8]]:=(*P(DUP))&0[CTC]; 14367022
IF WORKING THEN % MAKE SURE HARRY ISN 14367023
IF HARRYSTA=I THEN % DIDDLING OUR LINE 14367024
COMPLEXSLEEP(NOT WORKING OR HARRYSTA≠I); 14367025
CLEAR INPUT TANK; 14367030
TANKOK[I]:=0; 14367032
END; 14367034
SWAP(EQJSTATE,0); 14367036
MEMROW[P1MIX]:=MEMROW[0]; 14367038
T*(I+NOT FENCE INX 1) DIV 1890+2; 14367040
FORGET USER DISK(DISKSTORE[P1MIX],NABS(I DIV 30+T+T+2)); 14367050
END; 14367060
14367100
DAT[P1MIX]+0; 14367200
IF TYP≠0 THEN 14371000
$ SET OMIT = STATISTICS 14371999
FORGETSPACE(DIRECTORYSEARCH(VECTOR[7],IF VECTOR[0]<0 14372000
THEN "DISK " ELSE IF VECTOR[0],[2:1] THEN "CANDE " 14373000
ELSE ABS(VECTOR[1]),13)); 14373100
$ POP OMIT 14373101
$ SET OMIT = NOT(STATISTICS) 14373199
BEGIN 14373200
NT1:=DIRECTORYSEARCH(VECTOR[7],IF VECTOR[0] 14373300
LSS 0 THEN "DISK " ELSE IF VECTOR[0],[2:1] 14373400
THEN "CANDE " ELSE ABS(VECTOR[1]),13),[CF]); 14373500
OBJINFO:=M[NT1+3]&M[NT1+4][24:36:6]; 14373600
FORGETSPACE(NT1); 14373700
END; 14373800
$ POP OMIT 14373801
IF TYP=1 THEN % COMPILE PART OF COMPILE &GO 14374000
BEGIN% 14375000
RETURNEM; SI=[M[SPACE(31)]]&31[8:38:10]; 14376000
DISKWAIT("S,[CF],30,VECTOR[2],[FF]); 14376100
IF ERTOG=0 AND TYP=1 THEN 14376200
BEGIN% 14377000
SLEEP([TOGGLE],SHEETMASK); 14378000
LOCKTOG(SHEETMASK); 14379000
STREAM(A+0:B+P(,SCHEDULEIDS)); 14383100
BEGIN SI+B; 14383200
47(SKIP SB; SKIP DB; TALLY+TALLY+1; 14383300
IF SB THEN BEGIN END 14383400
ELSE JUMP OUT); 14383450
DS+SET; A+TALLY; 14383500
END STREAM; 14383600
T * P; S[3] + O&T[8:38:10]; 14383700
S[25] + CATCH; 14383740
S[6] := VECTOR[6]; 14383750
S[23],[24:24]+(CLOCK+P(RTR))DIV 60; 14383760
DISKWAIT(S,[CF],30, 14383800
VECTOR[2],[FF]); 14383900
I + IF S[18] > SHEETMAX THEN SHEETMAX 14385000

```

```

ELSE S[18]; 14386000
IF SHEET[I],[CF] ≠ 0 THEN 14387000
BEGIN DISKWAIT(-S,[CF],30, 14388000
SHEET[I],[FF]); 14389000
S[29] + VECTOR[2],[FF]; 14391000
DISKWAIT(S,[CF],30, 14392000
SHEET[I],[FF]); 14392500
END ELSE SHEET[I] + VECTOR[2],[FF]; 14394000
SHEET[I],[FF] + VECTOR[2],[FF]; 14395000
UNLOCKTOG(SHEETMASK); 14396000
FORGETSPACE(S INX 0); 14396050
NEEDSELECT:=1; XR93 14396100
GO AHEAD; 14396200
END% 14397000
ELSE BEGIN% 14398000
FORGETESPDISK(VECTOR[2],[18:15]);% 14399000
LINK + S[13]; 14399100
WHILE LINK≠0 DO 14399200
BEGIN DISKWAIT(-S,[CF],30,LINK); 14399300
FORGETESPDISK(LINK); LINK + S[29]; 14399500
END; 14399600
FORGETSPACE(S); 14399700
END 14400000
END ELSE% 14401000
IF TYP=0 THEN 14402000
BEGIN% 14403000
VECTOR[9]:=VECTOR[9],[CF]; 14403900
FOR I+1 STEP 1 UNTIL VECTOR[9] DO% 14404000
IF VECTOR[9+I] ≠ 0 THEN% 14405000
FORGETUSERDISK(VECTOR[9+I],-VECTOR[8]); 14406000
END ELSE 14407000
IF TYP=4 14407100
THEN GO TO RETURNEM; 14407200
IF (T:=VECTOR[6],[CF]) GEQ ESPDISKBOTTOM AND T LSS 14407300
ESPDISKTOP THEN FORGETESPDISK(T); 14407400
AHEAD; 14407500
IF JAR[P1MIX,0] < 0 THEN% 14408000
IF ERTOG ≠ 0 THEN% 14409000
TYP+3; 14410000
T+UVROW[I+P1MIX],[CF]; 14411000
$ SET OMIT = NOT(AUXMEM) 14411099
IF AUXDATA[P1MIX] NEQ 0 THEN 14411100
IF (AUXERRORTOG AND TWO(P1MIX)) = 0 THEN 14411110
AUXILIARYMEMORYCASUALTYRECOVERY 14411120
ELSE AUXDATA[P1MIX]=0; 14411130
$ POP OMIT 14411140
COMMENT SUBTRACT CORE REQUIREMENTS FROM CORE WORD; 14411200
IF VECTOR[2],[3:1] THEN 14411800
BEGIN 14411810
NT1:=GETSPACE(5,73,5)+2; 14411820
M[NT1 ]:= 0 & P1MIX[20:43:5]; 14411840
M[NT1+1]:= VECTOR[5],[1:23]; 14411850
M[NT1+2]:= XCLOCK & VECTOR[2][1:11:17] & 14411860
(VECTOR[1]<0)[18:42:6]; 14411870
M[NT1+3]:= VECTOR[0]; 14411880
M[NT1+4]:= VECTOR[1]; 14411890
LINKUP(14,NT1); 14411900
END; 14411910
VECTOR := VECTOR&SPACE(10)[CTC]; 14412110
MOVE(10,JARROW[P1MIX],VECTOR); 14412120

```

WHILE XCLOCK+P(RTR) GEQ WITCHINGHOUR DO MIDNIGHT;	14412200
MOTHER+IF VECTOR[1]<0 THEN 2 ELSE TYP#3;	14412300
STOPLOG(1,0);	14412350
NEXTMOM+VECTOR[3]+PROCTIME[1];	14412400
CATCH+((VECTOR[2]+USERCODE[1])#ABS(VECTOR[1])	14412600
AND USERCODE[1]#0)+1;	14412610
S := VECTOR&(SPACE(10)-1)[CTC];	14412800
MOVE(10,[S[0]],[S[1]]);	14412900
IF (LOGARRAY[31] AND IOMASK)#0 THEN	%R2714413000
SLEEP([LOGARRAY[31]],IOMASK);	%R2714413010
S[1] + NEXTMOM;	14413100
S[2] + VECTOR[4]+IOTIME[1];	14413200
S[3] + (LC[1]-SC[1])*CHUNKZIZE+CHUNKZIZE;	14413300
% SET OMIT = NOT(NEWLOGGING)	14413399
S[4] + ABS(MCPROCTIME[1]);	14413400
S[5] + MCP IOTIME[1];	14413500
% POP OMIT	14413501
MAKELOG([S[0]],[CF],EOJSTATS);	14414000
IF LOGLINE,[33:7] NEQ 0 THEN	%R2314414010
BEGIN PRTR:=IOQUE&GETAREA(2)[CTC];	%R2314414020
PRTR[0],[FF]:=LOGLINE;	%R2314414030
PRTR[1]:=MOTHER;	%R2314414040
PRTR[2]:=XCLOCK+P(RTR);	%R2314414050
MOVE(9,[S[1]],[PRTR[3]]);	%R2314414060
PRTR[12]:=ERTOG,[1:46];	%R2314414070
PRTR[13]:=VECTOR[0];	%R2314414080
PRTR[14]:=VECTOR[1];	%R2314414090
PRTR[15]:=USERCODE[P1MIX];	%R2314414100
PRTR[16]:=NEXT1[P1MIX];	%R2314414110
PRTR[17]:=NEXT2[P1MIX];	%R2314414120
QUEVENT(PRTR,[CF],CANDYINX);	%R2314414800
END ELSE	%R2314414810
IF NEXT1[P1MIX] NEQ 0 THEN	%R2314414820
ZIPPER(NEXT1[P1MIX],NEXT2[P1MIX]);	%R2314414830
% SET OMIT = NOT(STATISTICS)	14414834
JOBNUM:=JOBNUM-1;	14414835
S[1]:=CORETIME[P1MIX];	14414840
S[2]:=OLAYUSED[P1MIX];	14414845
S[3]:=READYQUETIME[1]&INITIALRQTIME[1][1:25:23];	14414850
S[4]:=OBJINFO&SQ[P1MIX][6:24:18];	14414855
S[5]:=SWAPS[P1MIX]&SWAPOUTS[P1MIX][1:25:23];	14414860
S[6]:=CODEPBITS[P1MIX]&DATAPBITS[P1MIX][1:25:23];	14414865
S[7]:=CODEOLAYS[P1MIX]&DATAOLAYS[P1MIX][1:25:23];	14414870
S[8]:=JOBNUM;	14414875
S[9]:=MORECPBITS[P1MIX]&MOREDPBITS[P1MIX][1:25:23];	14414880
MAKELOG([S[0]],[CF],SYSTATS);	14414885
% POP OMIT	14414886
FORGETSPACE([S[1]]);	%R2314414900
FORMTIME([VECTOR[4]],NEXTMOM);	14415000
LINK:=(NOT VECTOR[0]),[2:1];	%R2014415810
% SET OMIT = NOT(PACKETS)	14415819
TYP,[1:1]+(VECTOR[1]<0) OR (VECTOR[2],[8:10]=3);	14415820
% POP OMIT	14415821
IF STASUS[1] NEQ STABLE THEN	%R4114415850
COMPLEXSLEEP(STASUS[P1MIX]=STABLE);	%R4114415860
STREAM(B+MOTHER,C+CATCH,1,VECTOR);	14415900
BEGIN%	14416000
DS + LIT " "; DI + DI+7; %	14417000
C(DS+LIT "/"; DI+DI+7);	14418000
SI+LOC C; SI+SI+7;	14418100

```

IF SC="1" THEN DS+8 LIT " "; 14418200
DS+LIT"="; SI+LOC I; DS+2 DEC; 14419000
I+DI; DI+DI-2; DS+FILL; DI+I; 14419100
DS+ 5 LIT ",PST="; 14419200
DI+DI+8; 14419500
CI + CI+B;% 14420000
GO TO E;% 14421000
GO TO OK;% 14422000
DS+7 LIT " DS=ED "; 14423000
GO TO X;% 14424000
OK;% 14425000
DS+5 LIT " EOJ "; 14426000
GO TO X;% 14427000
E: DS+11 LIT " SYNTAX ERR "; 14428000
X: DS+LIT"="; 14429000
END; 14429100
SPOUTIT(VECTOR INX 0,(LINK OR CANDYMESS) AND 14430000
EOJMESS AND (NOT JAR9),[2:1] OR EOJK); 14430010
$ SET OMIT = NOT(PACKETS) 14430019
UNITNO:=PSEUDOMIX[P1MIX]; 14430020
P1MIX:=PSEUDOMIX[P1MIX]:=0; 14430030
IF UNITNO#0 THEN 14430040
DRAIN0(UNITNO,(TYP,[CF]#1),TYP,[1:1] 14430050
&JAR[I,6][1:1:1]); 14430060
$ POP OMIT 14430061
IF (LOGARRAY[31] AND IOMASK)=0 THEN 14430100
SLEEP([LOGARRAY[31]], IOMASK); 14430200
IF TABCNT[I]#0 THEN 14430300
BEGIN CLICK:=CLOCK+900; 14430400
COMPLEXSLEEP(TABCNT[I]=0); 14430500
END; 14430600
FORGETSPACE(JARROW[I]); 14431000
P1MIX+JARROW[I]+PRTROW[I]+0; 14431100
14431300
IF BATCHJOB[I] OR FSROW[I],[CF] LSS FENCE %R9314431500
OR NEEDSELECT THEN %R9314431600
SELECTION; 14432000
NEEDSELECT:=0; %R9314432500
FORGETSPACE(T); %R9314432600
KILL([RCW]);% 14434000
END L5COM;% 14435000
PROCEDURE ZIPPER(W1,W2); VALUE W1,W2; REAL W1,W2;% 14531000
BEGIN REAL T,I; 14532000
TI=GETSPACE(12,64,5)+4; 14533000
14534000
IF (1+USERCODE[P1MIX])=ABS(NOT 0) THEN I+ 0; 14534500
STREAM(K+@14,A+[W1],C+I,B+T); 14535000
BEGIN 14536000
SI+LOC K; SI+SI+7; DS+ CHR; 14537000
DS:= 5 LIT "USER="; SI:=LOC C; SI:=SI+1; DS:= 7 CHR; 14537100
DS+ 9 LIT ";EXECUTE "; SI+A; SI+SI+1; 14537200
DS+ 7 CHR; DS+ LIT "/" ; SI+SI+1; DS+ 7 CHR; 14538000
DS+ 6 LIT ";END,+"; 37(DS+ LIT " "); 14539000
END; 14540000
I+IF P1MIX=0 OR USERCODE[P1MIX]=MCP THEN 31 ELSE 26; 14541000
$ SET OMIT = NOT(PACKETS) 14541049
IF PSEUDOMIX[P1MIX] NEQ 0 THEN NYLONZIPPER[P1MIX],[2:1]:=0; 14541050
$ POP OMIT 14541131
FORK(PC,CONTROLCARD) OR (O&LOGLINE[32:1:1]XLOGLINE), 14541150
T&I[3:43:5]&P1MIX[18:42:6],-1,CCSTACK,0); 14541170

```



```

$ POP OMIT                                     14560010
      X := [M[R := SPACE(60)]]&60[8:38:10];    14561000
      AL:=IOQUE&(L+1)[CTC];                    14561100
      S:=SCRAMBLE(A,B);                        %10414562000
CHECK:  DISKWAIT(-R,-60,(J:=S));              14563000
      IF P1MIX #0 THEN%                        14564000
      IF THERE THEN%                          14567000
%                                              14567100
%      A FILE ALREADY EXISTS ON DISK WITH THIS NAME. IF WE ARE ALLOWED 14567110
%      TO REMOVE IT, AND IT IS NOT IN USE, WE WRITE THE NEW HEADER 14567120
%      OVER THE OLD ONE AND RETURN THE OLD FILE'S DISK. IF IT IS IN 14567130
%      USE, DIRECTORYSEARCH IS USED TO REMOVE IT WHEN IT IS FREE AND 14567140
%      THEN, HAVING LOST CONTROL OF THE DIRECTORY, WE RETURN TO CHECK 14567150
%      AND START ALL OVER AGAIN.              14567160
%                                              14567170
      BEGIN DISKWAIT(-R,-30,(T:=X[1+2],[CF])); 14568000
$ SET OMIT = NOT SHAREDISK                    14568990
      UNLOCK(S);                              14569000
$ POP OMIT                                     14569010
      IF B # USERCODE[P1MIX] THEN             % NEED TO CHECK SECURITY 14570000
      BEGIN J:=R&T[CTF];                     14571000
$ SET OMIT = SHAREDISK                        14571990
      UNLOCKDIRECTORY;                       14572000
$ POP OMIT                                     14572010
      IF SECURITYCHECK(A,B,USERCODE[P1MIX],J)#7 THEN 14573000
      BEGIN                                   14574000
$ SET OMIT = NOT SHAREDISK                    14574990
      UNLOCK(T);                             14575000
$ POP OMIT                                     14575010
      FILEMESS("INVALID", " USER ",A,B,0,0,0); 14576000
      END;                                    14577000
$ SET OMIT = SHAREDISK                        14577990
      END ELSE                                % OK TO REMOVE FILE 14578000
$ SET OMIT = NOT SHAREDISK                    14578250
      END;                                    14578500
$ POP OMIT OMIT                              14578510
      IF (X[4],[1:3] OR X[4],[12:24] OR X[9],[1:28]) = 0 THEN 14579000
      BEGIN R:=NABS(R);                       % NOT IN USE - WE CAN 14579500
      AT:=X;                                  % DO A QUICK REPLACE 14580000
      GO TO COPY;                             14580500
      END                                     14581000
$ SET OMIT = SHAREDISK                        14581490
      ELSE UNLOCKDIRECTORY                    14581500
$ POP OMIT                                     14581510
      ;                                       14582000
$ SET OMIT = NOT SHAREDISK                    14582490
      UNLOCK(T);                              14582500
$ POP OMIT                                     14582510
%                                              14582900
%      IN ORDER TO PROVIDE CONTINUITY OF FILE CHARACTERIISTICS, PARTS 14582910
%      OF THE OLD HEADER ARE NOW MOVED TO THE NEW HEADER.          14582920
%                                              14582930
%      IF P(DIRECTORYSEARCH(A,B,7),DUP) GE@ 64 THEN 14583000
%      BEGIN AT:=IOQUE&P(XCH)[CTC];          14583500
COPY:  AL[1]:=AT[1];                          14584000
      AL[2]:=AT[2];                          14584500
      AL[3]:=(*P(DUP))&AT[3][2:2:10];        14585000
      IF AL[4],[36:6]=0 THEN                  14585500
      AL[4]:=(*P(DUP))&AT[4][36:36:6];      14586000
      AL[5]:=AT[5];                          14586500

```



```

                AL[6]:=AT[6];
                IF R LSS 0 THEN GO TO FAST;
                FORGETSPACE(AT);
            END ELSE
            IF P=2 THEN FILEMESS("SYSTEM ", "FILE " ,A,B,0,0,0)
                ELSE GO INITIATE;
$ SET OMIT = SHAREDISK
    LOCKDIRECTORY;
$ POP OMIT
    GO TO CHECK;
    END ELSE ELSE T:=S;          % SETS UP FOR P1MIX=0
%
%   THE FILE IS NOT THERE, WE SEARCH FOR A VACANCY, IF ONE IS FOUND
%   Z AND T ARE ITS ADDRESS, IF THERE ISNT ONE, Z IS THE ADDRESS OF
%   THE LAST BLOCK AND T IS SET TO THE ADDRESS OF THE NEW BLOCK,
%
$ SET OMIT = NOT SHAREDISK
    DISKWAIT("([HOLDER],[CF]),=3,DIRECTORYSEG); % CLOBBERS X1
    X1:=M[R1:=SPACE(60)]&60[8:38:10];
$ POP OMIT
    DO BEGIN
        IF (Z1=T)≠J THEN DISKWAIT("R,60,Z);
        FOR I=0 STEP 3 UNTIL 57 DO
            IF (X[I] EQV @14)≠NOT 0 THEN GO TO FOUND;
        END UNTIL (T:=X[2],[FF])=0;
        X[2],[FF]← BYPASS + BYPASS=2;

        IF BYPASS,[CF]≤BYPASS,[FF] THEN DIRECTORYFULL(0);
$ SET OMIT = SHAREDISK
    DISKWAIT(R,60,Z);          % WRITE OUT POINTER TO NEW BLOCK
$ POP OMIT
    T:=BYPASS,[CF];
    X1[0]:=@14; MOVE(59,X1,X1 INX 1);
$ SET OMIT = NOT SHAREDISK
    X:=X1;
$ POP OMIT
    I:=0;
    FOUND:%
                PBCOUNT←PBCOUNT+((((A EQV"PBD ")=NOT 0) OR
                ((A EQV"PUD ")=NOT 0)) AND (B,[CF]=1));
    X[I]←A; X[I+1]←B; X[I+2],[CF]←NEXTSLOT;
$ SET OMIT = NOT SHAREDISK
$ SET OMIT = NOT STATISTICS OR OMIT
    BYPASSBOTTOM:=BYPASS,[CF];
$ POP OMIT
    IF T≠Z THEN          % WRITE NEW BLOCK, OLD IS UPDATED LATER,
$ POP OMIT
    DISKWAIT(R1,60,T);
%
%   UPDATE THE NAME SEGMENT, BUT DONT WRITE IT OUT UNTIL THE NEW
%   HEADER IS WRITTEN,
%
    J←(NEXTSLOT-DIRECTORYTOP-3)&0[44:44:4]+DIRECTORYTOP+19;
    I:=((T:=NEXTSLOT)-J)×2+30;
    DISKWAIT("R1,"30,J);
    NEXTSLOT:=X1[I+1];
    X1[I]:=A; X1[I+1]:=B;
    IF NEXTSLOT=0 THEN          % GOING TO USE EOF RECORD
    IF I=0 THEN          % WRITE NEW EOF RECORD BEFORE
    BEGIN P(X1[28],X1[29]); % DESTROYING CURRENT ONE

```

```

14587000
14587500
14588000
14588500
14589000
14589200
14589490
14589500
14589510
14590000
14590500
14590900
14590910
14590920
14590930
14590940
14590990
14591000
14591100
14591110
14591500
14592000
14593000
14594000
14595000
14596000
14597000
14598000
14598090
14598100
14598110
14598200
14598300
14598390
14598400
14598410
14598500
14599000
14599900
14599910
14600000
14600290
14600299
14600300
14600301
14600400
14600410
14600500
14600900
14600910
14600920
14600930
14601000
14601500
14602000
14602500
14603000
14603100
14603110
14603200

```

```

X1[28]:=0114;          % SAVE NAME, REPLACE WITH "END" FLAG, 14603300
X1[29]:=0;            14603310
NEXTSLOT:=T+30;      14603320
BYPASS,[FF] + J+16;  14603330
DISKWAIT(R1,30,J+16); 14603400
P([X1[29]],+,[X1[28]],+); % RESTORE CLOBBERED NAME 14603600
IF J+16>BYPASS,[CF] THEN DIRECTORYFULL(0); 14603700
END ELSE 14603800
BEGIN X1[I-2]:=0114; X1[I-1]:=0; NEXTSLOT:=T-1 END; 14604000
% 14604900
% NOW WE CAN WRITE EVERYTHING OUT, NOTE THAT IN ORDER TO MINIMIZE 14604910
% THE DAMAGE CAUSED BY AN UNTIMELY HANG, THE MAIN AND (FOR 14604920
% SHAREDISK) THE BYPASS DIRECTORIES ARE CORRECT AT ALL TIMES, 14604930
% 14604940
FAST; %R92 14605000
$ SET OMIT = NOT SHAREDISK 14605490
  IF NOT DELETETOG THEN 14605500
    FOR I:=AL[9],[43;5]+9 STEP -1 UNTIL 10 DO 14606000
      IF AL[I]#0 THEN SCRATCHDIRECTORYDELETE(AL[I],AL[B]); 14606500
$ POP OMIT 14606510
  DISKWAIT(L+1,-30,T); % FILE HEADER 14607000
  IF R GTR 0 THEN 14607500
    BEGIN 14608000
$ SET OMIT = NOT SHAREDISK 14608490
  DISKWAIT([HOLDER],[CF],-3,DIRECTORYSEG);% BYPASS & NXTSLOT 14608500
$ POP OMIT % (CLOBBERS X1) 14608510
  DISKWAIT(R1,-30,J); % NAME SEGMENT 14609000
$ SET OMIT = NOT SHAREDISK 14609990
  FORGETSPACE(R1); 14610000
  DISKWAIT(R,-60,Z); % BYPASS BLOCK 14610500
  IF S#Z THEN UNLOCK(S); 14611000
$ POP OMIT 14611010
  END ELSE % CLEAN UP OLD HEADER 14612000
  BEGIN DISKLOG(A,B,AT); 14613000
    J:=AT[9]+9; 14614000
    FOR I:=10 STEP 1 UNTIL J DO 14615000
      IF AT[I]#0 THEN FORGETUSERDISK(AT[I],AT[B]); 14616000
    END; 14617000
$ SET OMIT = SHAREDISK 14617990
  UNLOCKDIRECTORY; 14618000
$ POP OMIT 14618010
  EUF:=T; 14619000
  BOMBOUT;% 14620000
  FORGETSPACE(R); 14621000
  END ENTERUSERFILE ;% 14622000
PROCEDURE COM1; COMMENT ALGOL I/O COMMUNICATE;% 14623000
BEGIN REAL CODE=-4,TANK=-5,ROW=-6;% 14624000
  REAL MID=-8,FID=-7; 14624050
  REAL STA=-6, RESULT=-7, B, T, F, S, TIMEOUT=-7; 14624100
  NAME PHYL=-5, A; 14624200
  ARRAY HEADER=-5[*];% 14625000
  ARRAY FINAL=-6[*];% 14626000
  LABEL OPEN,PARITY,EOF,EOT,DISKSPACE,DISKLOCK;% 14627000
  LABEL ARGH,ECH,PURGELOCK,SEEKDC; 14627400
  LABEL CLOSE,RDATA,SELERR,SPACE,REFILL;% 14628000
  LABEL READLABEL; 14629000
  LABEL IOREQ,ROTATE;% 14630000
  SWITCH FUNCTION+OPEN,PARITY,EOF,EOT,DISKSPACE,SEEKDC,CLOSE, 14631000
    RDATA,SELERR,SPACE,REFILL,READLABEL,IOREQ,ROTATE 14632000
  ; 14632900

```

REAL INFO,LOC,USASI;	14633000
REAL I;%	14634000
LABEL MESSAGE, BACK;	14635000
ARRAY FPB[*],FIB[*];%	14636000
REAL TANG=TANK;	14636100
IF CODE=20 THEN GO TO PURGELOCK;	14636900
GO TO FUNCTION[CODE];%	14637000
PARITY:  INFO=" PAR"; GO TO MESSAGE;%	14639000
EOF:      INFO=" EOF"; GO TO MESSAGE;%	14640000
EOT:      INFO=" EOT";%	14641000
MESSAGE:  FPB=PRY[P1MIX,3]; FIB=M[P(,TANK,LOD),[33:15]-3];%	14642000
IF FIB[5],[1:1] THEN INFO:= " INV" OR M;	14642100
IF FIB[4],[8:4]=14 THEN	14642200
BEGIN TERMINATE(P1MIX); TERMINALMESSAGE(61) END;	14642300
STREAM(X=INFO,Z=0,F= IF TANG=0 THEN 0 ELSE	14643000
[FPB[FIB[4],[13:11]]],	14643100
Q=TANG#0,	14643200
J=JARROW[P1MIX],%	14644000
DI=(CODE:=GETSPACE(10,2,5)+2));	14645000
BEGIN DS = LIT "="; SI=LOC X; SI=SI+5; DS=3 CHR;%	14646000
SI=LOC X; IF SC=0 THEN DS=10 LIT " NO LABEL "%	14647000
ELSE IF SC="8" THEN DS=11 LIT	14647100
"WRITE TU 0 "	14647200
ELSE IF SC=@30 THEN DS:=10 LIT	14647300
"ALID USER "	14647400
ELSE DS=10 LIT "ECT ERROR ";%	14648000
Q(X=DI; SI=F; DI=LOC Z;	14649000
IF 8 SC#DC THEN BEGIN SI=F; SI=SI+1; DI=X;%	14650000
DS=7 CHR; DS=LIT" "; X=DI;	14651000
END;%	14652000
DI=X; SI=SI+1; DS= 7 CHR );	14653000
DS=2 LIT "i=";%	14654000
END;%	14655000
	14656000
	14657000
TERMINATE(P1MIX); TERMINALMESSAGE((-CODE));%	14658000
DISKSPACE:OPEN;CLOSE;	XR9014659000
IF HEADER[4] THEN % FILE IS IN DIRECTORY	14659100
FORGETSPACE(DIRECTORYSEARCH	14659200
(MID,FID,"(HEADER,[CF])&ROW[CTF]) ELSE	14659300
HEADER[ROW]=GETUSERDISK(HEADER[8]);	14659400
SEEKDC:  GO TO INITIATE;	14660000
PURGELOCK:  SAVEWORD:=SAVEWORD OR TWO(ROW);  ::	14671000
RDATA:  INFO=" RER"; GO TO MESSAGE;	14673000
SELERR:  INFO := @37000000060622543; GO TO MESSAGE;	14674000
SPACE:  FIB=M[P(,TANK,LOD),[33:15]-3]; LOC=FIB[15],[25:5];%	14675000
BLASTQ(LOC);%	14676000
FPB=[MEMORY[5]]&3[23:46:2]&ROW[22:1:1];%	14677000
ROW=ABS(ROW);%	14678000
WHILE (ROW+ROW=1)≥0 DO INFO=WAITIO(FPB,@40,LOC);%	14679000
GO TO INITIATE;  ::	14680000
REFILL:  FIB=M[(TANK+P(,TANK,LOD),[33:15])-3];%	14681000
CODE=FIB[13],[10:9]=1;%	14682000
LOC=FIB[19],[33:15]=FIB[16],[33:15];%	14683000
FPB=MEMORY[FIB[16] INX 0+ROW];%	14684000
INFO=FPB,[18:15];%	14685000
FOR I=1 STEP 1 UNTIL CODE DO%	14686000
BEGIN IOREQUEST(FLAG(FIB[19]&(INFO+LOC)[33:33:15]),%	14687000
FIB[16]&INFO[33:33:15],FPB);%	14688000
MEMORY[TANK]=MEMORY[TANK]&0[2:2:1]&0[19:19:1];%	14689000

```

                                &0[26:26:7]&INFO[33:33:15];%      14690000
      STREAM(CODE,T+MEMORY[TANK],TANK);%                      14691000
      BEGIN SI=TANK; SI=SI+8; DS=CODE WDS;%                    14692000
              SI=LOC T; DS=WDS;%                               14693000
      END;%                                                     14694000
      INFO+MEMORY[INFO+ROW],[18:15];%                          14695000
END;%                                                         14696000
                                GO TO INITIATE;               !! 14697000
READLABEL: FIB=M[(TANK+P(,TANK,LOD),[33:15])=3];%            14698000
           LOC=FIB[15],[25:15];%                               14699000
           BLASTQ(LOC);%                                       14700000
           P(WAITIO((FIB[5],[44:1])*(M[TANK=2],[8:10]-1) INX M[TANK=2])
              &M[TANK][21:21:4],@37700000,LOC),DEL);          14701000
           STREAM(Y:=0;X:=0,X1:=0,X2:=0,Z:=M[TANK=2]);        14702000
           BEGIN DI:=LOC X; DS:=24 LIT "VOL1HDR1HDR2EOF1EOF2EOFV1"; 14702050
                   DI:=LOC X;                                    14702100
                   6(TALLY:=TALLY+1);                            14702150
                   SI:=Z;                                         14702200
                   IF 4 SC=DC THEN                                14702250
                       JUMP OUT TO A);                            14702300
                   TALLY:=0;                                       14702350
           A;                                                     14702400
           Y:=TALLY;                                             14702450
           END;                                                  14702500
           IF (USASI:=P)>0 THEN                                    14702550
               USASITAPE(M[TANK=2],[CF],USASI,3,LOC,FIB[5],[44:1]); 14702600
               P(WAITIO((M[5])&3[23:46:2]&(NOT FIB[5])[22:44:1],
                  @37700000,LOC),DEL);                            14703000
               GO TO INITIATE; !!                                  14703100
IOREQ:   FPB=MEMORY[(IF (INFO+NFLAG(MEMORY[P(TANK,DUP,[M],INX,PRL)]))
                    ,[22:1] THEN 2 ELSE NOT 1) INX INFO];%    14704000
           IOREQUEST(FINAL,INFO,FPB);%                          14705000
           MEMORY[TANK]+MEMORY[TANK]&0[26:26:7]&0[19:47:1];%    14706000
           GO TO INITIATE;%                                       14707000
!!                                               14708000
ROTATE:  TANK+P(,TANK,LOD),[33:15];%                             14709000
           STREAM(T+M[TANK],N+ROW=1,D+TANK);%                  14709300
           BEGIN SI=D; SI=SI+8; DS=N WDS; SI=LOC T; DS=WDS END;% 14710000
IF M[TANK],[3:5]=16 THEN                                     14711000
IF M[TANK],[24:1] THEN                                       14712000
IF (I+P(M[TANK=3],14,COC))#0 THEN                             14712100
    BEGIN                                                    14712200
        PHYL + TANK INX M;                                     14712300
        FOR LOC + ROW=1 STEP =1 UNTIL 0 DO                    14712400
            BEGIN                                             14712500
                IF ( B+M[PHYL[LOC] INX NOT 2])#0 THEN        14712600
                    BEGIN                                       14712750
                        IF (I+I=1) <= 0 THEN                  14712800
                            LOC + =1;                           14712850
                    END;                                         14712900
                END;                                            14712950
            END;                                                14713000
        GO TO INITIATE;%                                       14713100
DISKLOCK;                                                  14714000
END COM1;%                                                  15006000
PROCEDURE DISPLAY(X); VALUE X; REAL X;%                       15019000
    BEGIN REAL T;                                             15020000
        STREAM(X;J+JARROW[P1MIX],P1MIX,%                     15021000
            Y:=T:=SPACE(25));                                  15022000
    END;

```

BEGIN DS ← LIT "X";	15023000
2(SI ← J; SI ← SI+1; DS ← 7 CHR; J ← SI);	15024000
L: SI ← SI-1;	15025000
IF SC = " " THEN	15026000
BEGIN DI ← DI-1; GO TO L END;	15027000
DS ← LIT "/";	15028000
DI ← DI-1; DS ← LIT "=";	15029000
SI ← LOC P1MIX; DS ← 2DEC; P1MIX ← DI; DI ← DI-2;	15030000
DS ← FILL; DI ← P1MIX; DS ← 2LIT " ";	15030500
SI ← X;	15031000
H: 4(40(IF SC=" " THEN JUMP OUT 2 TO HH;	15032000
DS ← CHR)); HH:	15033000
J ← DI; DI ← DI+8; SI ← J;	15034000
S: SI ← SI-1; IF SC = " " THEN GO TO S;	15035000
SI ← SI+1; J ← SI; DI ← J; DS ← LIT "X";	15036000
X ← DI;	15037000
END;	15038000
X ← (((X+P), [CF]) - T) × 8 + X, [30:3] - 1;	15039000
SPOUT(P(X,T));	15040000
END;	15041000
PROCEDURE COM13 ;	15059000
BEGIN	15060000
%  COBOL IO INTERFACE COMMUNICATE	15061000
REAL CODE = 4, REEL = 6 ;	15062000
NAME FLOC = 5 ;	15063000
ARRAY FIB [*];	15064000
REAL T, COB68;	15065000
LABEL L4;	15066000
DEFINE INOUT=FIB[13],[27:1]#, DIREC=FIB[13],[25:1]#;	15067000
SORTFILE=FIB[4],[7:1]#, LABELSOMITTED=FIB[4],[2:1]#;	15071000
COB68 ← (FIB ← *(FLOC)), [8:10] = 22;	15072000
IF CODE=4 THEN GO TO L4;	15073000
INOUT ← CODE#3; DIREC ← CODE=2;	15074000
IF NOT COB68 THEN	15080000
IF FIB[5],[46:2]=3 THEN BEGIN	15080900
FIB[18],[18:15]+FIB[18],[3:15];	15081000
IF CODE=3 THEN	15082000
FIB[18],[3:15]+FIB[18],[33:15]+FIB[18],[3:15]; END;	15082100
NT1:=FLOC INX 3;	15083000
P(0,STF,PRT[P1MIX,8],STS);	%R9015084000
FILEOPEN(1,NT1);	%R9015085000
L4:	%R9015086000
CODE ← IF (CODE+ABS(REEL))=0 THEN 6 ELSE	15093000
(IF CODE=1 THEN 7 ELSE	15094000
(IF CODE=2 THEN 10 ELSE	15095000
(IF CODE=4 THEN @22 ELSE	15096000
(IF CODE=64 THEN @52 ELSE 0))));	%KRUNCH 15097000
IF (T+FIB[4],[8:4])#2 AND T#4 AND T#8 THEN CODE ← 0;	%KRUNCH 15097500
IF T=4 AND CODE=0 THEN CODE ← 10 ;	15098000
FILECLOSE(( FLOC INX 3 ) & CODE[18:33:15]);	15099000
IF CODE=0 OR CODE=10 OR CODE=@22 THEN FIB[5],[42:1] ← 1	15100000
ELSE FIB[5],[40:2] ← (CODE=7) × 2 + 1;	15101000
IF NOT COB68 THEN	15102000
IF FIB[5],[46:2]=3 THEN BEGIN	15102900
FIB[18],[3:15]+FIB[18],[18:15]; FIB[18],[18:15] ← 0 END;	15103000
GO TO INITIATE;	15104000
END COM13;	15105000
PROCEDURE WHATSIT(BUFH); VALUE BUFH; REAL BUFH;	15106000
BEGIN	%DSX 15106500
	15107000

REAL BF,I,LINE;	15107500
ARRAY INF[*];	15108000
REAL MIX, L=MIX, RCW=+0;	15108100
LABEL EXIT,EDD,GOTIT,OK;	15108500
BOOLEAN GETBUFF,NOSD;	15109000
BOOLEAN SUBROUTINE SPOUTLINE;	15109500
BEGIN	15110000
SYSDISKIO(1,LINE,INF);	15110500
P(0);	15111000
IF (INF[0]=STATABLE[L:=ABS(LINE)]),DIALEDUP THEN	15111100
IF INF[1] GTR 0 THEN GO TO OK ELSE	15111200
IF NOT LINETABLE[	15111280
\$ SET OMIT = TWXONLY	15111299
IF L GTR LMAX THEN STATABLE[L],LEENKER ELSE	15111300
\$ POP OMIT	15111301
L],DIRECTLINE THEN	15111320
BEGIN	15111400
INF[1]:="NOBODY ";	15111500
P(DEL,1);	15111600
IF GETBUFF THEN	15112000
BF:=SPACE(5);	15112500
GETBUFF← TRUE;	15113000
STREAM(S:=IF L GTR LMAX THEN 0 ELSE SCHEDLINE[L],	15113500
U:=INF[1], L, BF);	15113700
BEGIN	15114000
SI←LOC U; SI←SI+1; DS←LIT " ";	15114500
DS←7 CHR; DS←4 LIT " ON ";	15115000
U:=DI; DS:=3 DEC;	15115100
S(DS:=8 LIT"="(SCHED)"");	15115600
DS:=LIT"←";	15115700
DI:=U; DS:=2 FILL;	15116000
END;	15116500
SPOUT(BF);	15117000
END;	15117500
SPOUTLINE← P;	15117550
END;	15118000
BF← BUFH,[15:15]-1;	15118500
NOSD:=SYSDISKADR=0;	15118700
IF (MIX:=BUFH,[9:6]) ≠ 0 THEN       % <MIX> WU	15119500
BEGIN I:=1;	15120000
IF NOSD THEN ELSE	15120100
FOR LINE := 1 STEP 1 UNTIL STATIONMAX DO	15120500
IF STATABLE[LINE],MIXNR=MIX THEN	15121000
GO GOTIT;	15121500
I:=0;	15122000
GOTIT:	15122490
STREAM(T←PUTORTAKE(MIX,[USERCODE[MIX]],1,0),V←PUTORTAKE(MIX,	15122500
[JAR[MIX,0]],1,0),W←PUTORTAKE(MIX,[JAR[MIX,1]],1,0),	15123000
X:=MIX, Y:=LINE, Z:=1, S:=IF NOSD OR LINE GTR LMAX	15123500
THEN 0 ELSE SCHEDLINE[LINE], BF);	15123700
BEGIN	15124000
SI←LOC T; SI←SI+1; DS←LIT " ";	15124500
DS← 7 CHR; DS← LIT " "; SI← SI+1; DS← 7 LIT " USING ";	15125000
DS← 7 CHR; DS← LIT "/" ; SI← SI+1; DS← 7 CHR; DS← LIT "=";	15125500
DS←2 DEC; X←DI; DS←LIT"←"; DI←DI-3; DS←FILL; DI←X;	15126000
Z(DS:=4 LIT" ON "; X:=DI; DS:=3 DEC;	15126500
S(DS:=8 LIT"="(SCHED)""); DS:=LIT"←";	15126600
DI:=X; DS:=2 FILL);	15127100
END;	15127500
SPOUT(BF);	15128000

```

GO EXIT;
END;
STREAM(L*0;Z*BUFH);
BEGIN
SI* Z; DI* LOC L;
DD: IF SC=" " THEN BEGIN SI* SI+1; GO DD END;
IF SC>0 THEN IF SC<9 THEN
BEGIN
Z* SI; SI* SI+1;
IF SC>0 THEN IF SC<9 THEN
BEGIN SI* Z; DS* 2 OCT; GO EGRESS END;
SI* Z; DS* OCT; GO EGRESS;
END;
DS* 8 LIT "+0000001";
EGRESS: END;
IF (LINE#P) GTR STATIONMAX OR LINE=0 THEN
BEGIN
SPOUT(BF); GO EXIT;
END;
IF NOSD THEN ELSE
INF:=[M[SPACE(SYSDISKRL)]]&SYSDISKRL[8:38:10];
IF LINE>0 THEN %WU <LINE>
IF NOSD THEN ELSE
BEGIN
LINE* NABS(LINF);
IF NOT SPOUTLINE THEN
BEGIN;
STREAM(L:=ABS(LINE),C:=INF[0],DIALEDUP,B:=BF);
BEGIN
SI*LOC L; DS*LIT" "; DS*3 DEC;
C(DS* 15 LIT " NOT DIALED-UP="; JUMP OUT TO X);
DS* 12 LIT " NOT IN USE=";
X: DI*B; DS*3 FILL;
END;
SPOUT(BF);
END;
GO EDD;
END;
I* 0; %WU
IF NOSD THEN
BEGIN
STREAM(BF);DS:=16 LIT"#NO SYSTEM DISK=";
I:=1;
SPOUT(BF);
END
ELSE
BEGIN
FOR LINE:=0 STEP 1 UNTIL STATIONMAX DO
I* I+SPOUTLINE;
SYSDISKIO(1,STATIONMAX+1,INF);
END;
IF I#0 THEN
BEGIN;
STREAM(BF); DS*9 LIT" NULL WU=";
SPOUT(BF);
END;
EDD:IF NOSD THEN ELSE FORGETSPACE(INF);
EXIT;
END;

```

```

15128500
15129000
15129500
15130000
15130500
15131000
15131500
15132000
15132500
15133000
15133500
15134000
15134500
15135000
15135500
15136000
15136500
15137000
15137500
15137600
15138000
15138500
15139000
15139100
15139500
15140000
15140500
15141000
15141500
15142000
15142500
15143000
15143500
15144000
15144500
15145000
15145500
15145550
15146000
15146500
15146600
15146610
15146620
15146630
15146640
15146650
15146660
15146670
15147000
15147500
15147600
15147700
15148000
15148500
15149000
15150000
15150500
15151000
15151500
15152000

```

```

BOOLEAN PROCEDURE CONQUER(C,N,L,S,G);
VALUE C,N,L,S,G;
REAL C,N,L; ARRAY S[*];%
INTEGER G;
BEGIN ARRAY B=C[*];%
REAL T,I=T;%
LABEL X,Y;
IF PRT[P1MIX,0]#WORDOFEASE THEN GO TO STACKOVERFLOW;
WHILE L<0 DO
IF MEMROW[P1MIX],[CF]≥FENCE THEN
BEGIN NT1+M[MEM[P1MIX,AVAIL]];
NT2+0;
WHILE (NT1+M[NT1]),[FF]#@77777 DO
NT2+NT1,[FF]+NT2;
IF =N×L×1,1<NT2 THEN GO TO Y;
IF CANEXPAND[P1MIX] THEN GO TO Y;
EXPAND[P1MIX]+3;
SWAP(FORCESWAP,1);
IF TERMSET(P1MIX) THEN P(XIT);
END ELSE L+ABS(L);
Y: L:=ABS(L); IF G THEN IF L×N GTR 512 THEN GO TO X;
IF (T + GETSPACE(N×L,2,3)) = 0 THEN%
BEGIN IF NOT G THEN P(O,RTN);
X: IF NOT N THEN
BEGIN G+CONQUER(C,N+N DIV 2,L,N INX S,1);
G+CONQUER(S INX N,N,L,S,1);
P(1,RTN);
P(XIT);%
END;%
T + GETSPACE(L,2,1);%
END;%
B + [M[T+2]]&L[8:38:10]&C[18:33:15];%
N + N-1;%
FOR I + 0 STEP 1 UNTIL N DO%
BEGIN S[I]+P(DUP)&P(B,XCH)[CTF];
B + L INX B;%
END;%
CONQUER+1;
END;%
REAL PROCEDURE BATCHSELECT(F,N,B,L); VALUE F,N;
REAL F,N,B,L;
BEGIN
REAL I,J; LABEL L1,L2;
IF F NEQ 0 THEN % SHEET ENTRY PASSED AS A PARAMETER
IF NOT L THEN % SHEET QUEUE NOT "LOCKED"
BEGIN
SLEEP([TOGGLE],SHEETMASK);
LOCKTOG(SHEETMASK);
L:=1;% "LOCKED" TOGGLE
END;
IF (N:=(64×N-(N NEQ 0)) DIV CHUNKZIZE) GTR CHUNKMAX THEN N:=CHUNKMAX;
N:=(N=0)+N;
FOR I:=N STEP 1 UNTIL CHUNKMAX DO
BEGIN
FOR J:=0 STEP 1 UNTIL N DO
IF BATCHED[I=J] THEN GO TO L1;
B:=B&(64×N+I)[CTF]; % SIGN BIT INDICATES "OK" TO RUN
GO TO L2;
L1: END;
BATCHSELECT:=TRUE; % CANT RUN IT NOW

```

```

15168000
15168100
15169000
15169100
15170000
15171000
15172000
15172100
15172300
15172400
15172500
15172600
15172700
15172710
15172800
15172810
15172900
15173000
%101 15173050
15173100
15173200
15174000
15175000
15175900
15176000
15177000
15177800
15178000
15179000
15180000
15181000
15182000
15183000
15184000
15185000
15186000
15187000
15187500
15188000
15200000
15200100
15200200
15200300
15200400
15200500
15200600
15200700
15200800
15200900
15201000
15201100
15201200
15201300
15201400
15201500
15201600
15201700
15201800
15201900
15202000

```



```

L2: END;
PROCEDURE COREPRINT(Q);
VALUE Q; REAL Q; %Q IS MIX TO BE CU-ED = Q.[1:1] MEANS ALL MIXES
BEGIN
  LABEL TEST, AROUND, SPUTTER;
  REAL LINK, A, N, MIX; ARRAY C[*];
  $ SET OMIT = NOT(PACKETS)
  REAL UNITNO;
  $ POP OMIT
  SUBROUTINE CHECKMEM;
  BEGIN A:=MEMROW[MIX] INX 0;
    NT2:=MEM[MIX,0],[FF];
    WHILE A ≠ NT2 DO
      BEGIN IF NOT (LINK:=M[A]),[1:1] THEN
        BEGIN NT1:=LINK,[CF]=A;
          IF LINK.[2:1] THEN NT1:=O&NT1[CTF];
            C[LINK,[9:6]]:=( *P(DUP) )+NT1;
          END;
        A:=LINK INX 0;
      END; END;
  C:=UVROW&(SPACE(MIXMAX+1))[CTC];
  FOR A:=0 STEP 1 UNTIL MIXMAX DO C[A]:=0;
  MIX:=ABS(Q);
  $ SET OMIT = NOT(PACKETS)
  IF MIX≠0 THEN UNITNO:=PSEUDOMIX[MIX];
  $ POP OMIT
  IF MEMROW[MIX],[CF] < FENCE THEN CHECKMEM ELSE GO TO TEST;
  IF Q,[1:1] THEN
    FOR MIX:=1 STEP 1 UNTIL MIXMAX DO
      IF JARROW[MIX] ≠ 0 THEN
        BEGIN
TEST:   IF (STATUS[MIX] AND STABLE) = 0 THEN
        BEGIN CLICK:=CLOCK + P(RTR) + 180;
          SLEEP([SQ[MIX]],O&STABLE[18:42:6]);
        END;
        IF JARROW[MIX]≠0 THEN
        BEGIN TABCNT[MIX]:=TABCNT[MIX]+Q,[1:1];
          IF ((A:=(N:=SQ[MIX]),[18:6]) AND STABLE) = 0 THEN
            C[MIX],[9:1]:=1 ELSE
            IF N,[30:6] ≠ 0 THEN %JOB RUNS ABOVE THE FENCE
            BEGIN IF A#READYSTATE AND A#RDYRPT
              AND A#WAITSTATE THEN CHECKMEM;
              C[MIX],[9:9]:=N.[30:6]-N.[36:6]+1;
            END;
          END ELSE C[MIX]:=0;
          IF NOT Q,[1:1] THEN GO AROUND;
        END;
        AROUND:
        FOR MIX:=0 STEP 1 UNTIL MIXMAX DO
          IF (Q,[1:1] OR Q=MIX) AND C[MIX]≠0 THEN
            BEGIN LINK:=SPACE(9);
              IF MIX ≠ 0 THEN IF C,[9:1] OR
                (A:=PUTORTAKE(MIX,[PRYOR[MIX]],1 OR M,0))=NOT 0 THEN
                BEGIN STREAM(MIX,LINK);
                  BEGIN SI:=LOC MIX; DS:=4 LIT "MIX "; DS:=2 DEC;
                    DS:=9 LIT " IS HUNG-"; DI:=DI-11; DS:=FILL;
                  END; GO TO SPUTTER;
                END ELSE
                STREAM(X:=0:A,
                  B:=PUTORTAKE(MIX,[JAR[MIX,0]],1,0),

```

```

15202600
15300000
15301000
15302000
15302500
15303000
15303499
15303500
15303501
15304000
15305000
15306000
15307000
15308000
15309000
15310000
15311000
15312000
15313000
15314000
15315000
15316000
15316100
15316199
15316200
15316201
15317000
15317500
15318000
15319000
15319500
15320000
15321000
15322000
15323000
15323500
15324000
15324250
15324500
15325000
15326000
15327000
15328000
15328500
15328750
15329000
15329250
15329500
15330000
15331000
15332000
15333000
15334000
15335000
15336000
15337000
15338000
15339000
15340000
15341000

```

```

D:=PUTORTAKE(MIX,[JAR[MIX,1]],1,0),
MIX,N:=C[MIX],[9:9],S:=LINK);
BEGIN SI:=LOC A; DS:=6 DEC; DI:=DI-6;
DS:=5 FILL; DI:=S; DI:=DI+6;
DS:=LIT " "; 2(SI:=SI+1; DS:=7 CHR;
DS:=LIT "/" ); DI:=DI-1;
DS:=LIT "="; DS:=2 DEC; S:=DI;
DI:=DI-2; DS:=FILL; DI:=S;
N(DS:=8 LIT " USING "; DS:=2 DEC;
S:=DI; DI:=DI-2; DS:=FILL; DI:=S;
DS:=7 LIT " CHUNKS"; JUMP OUT);
X:=DI; DS:=LIT "+";
END ELSE
STREAM(N:=0;LINK);
BEGIN DS:=14 LIT " 0:MCP/DISK= 0"; N:=DI; END;
IF (N:=C[MIX],[FF]) # 0 THEN
STREAM(S:=P;O:=N,R:=C[MIX],[CF]);
BEGIN DI:=S;
SI:=LOC 0; DS:=6 LIT " SAVE="; DS:=5 DEC;
S:=DI; DI:=DI-5; DS:=4 FILL; DI:=S;
SI:=LOC R; DS:=6 LIT " OLAY="; DS:=5 DEC;
DS:=LIT "+"; DI:=DI-6; DS:=4 FILL;
END STREAM;
P(DEL);
SPUTTER; TABCNT[MIX]:=TABCNT[MIX]-Q,[1:1];
SPOUTER(LINK,UNITNO,1);
END;
IF Q,[1:1] THEN %PRINT TOTAL
BEGIN P(C[0]);
FOR MIX:=1 STEP 1 UNTIL MIXMAX DO
IF C[MIX],[9:9]=0 THEN P(C[MIX],ADD);
N:=(N:=P) INX N,[FF];
STREAM(N,A:=A:=SPACE(5));
BEGIN SI:=LOC N;
DS:=31 LIT "TOTAL MEM IN USE BELOW FENCE = ";
DS:=5 DEC; DS:=LIT "+"; DI:=DI-6; DS:=4 FILL;
END;
SPOUT(A);
END;
FORGETSPACE(C INX 0);
END CORE PRINT;
BOOLEAN PROCEDURE PRTGAMES(BUFF,MIX); VALUE BUFF,MIX; REAL BUFF,MIX;
COMMENT PRTGAMES IS THE BUSINESS END OF "IN" OR "OT" MESSAGES;
BEGIN REAL NX,INDEX,DATA,T,J;
$ SET OMIT = NOT(PACKETS)
DEFINE UNITNO = PSEUDOMIX[MIX]#;
$ POP OMIT
LABEL ECH, X;;;
STREAM(BUFF,F+BUFF<0,D+[DATA],I+[INDEX]);
BEGIN SI+BUFF;
L: IF SC=" " THEN BEGIN SI+SI+1; GO L END;
4(IF SC# " " THEN IF SC# "+" THEN IF SC# "=" THEN
BEGIN TALLY+TALLY+1; SI+SI+1 END);
I+TALLY; DI+DI+8; DI+DI-1; SI+SI-1; DS+1 CHR;
FC
M: IF SC=" " THEN BEGIN SI+SI+1; GO M END;
IF SC# "=" THEN BEGIN E:DI+DI-1;DS+LIT"";JUMP OUT END;
SI+SI+1;
N: IF SC=" " THEN BEGIN SI+SI+1; GO N END; TALLY+0;
8(IF SC>="0" THEN BEGIN TALLY+TALLY+1; SI+SI+1 END

```

```

15342000
15343000
15344000
15345000
15346000
15347000
15348000
15349000
15350000
15351000
15352000
15353000
15354000
15355000
15356000
15357000
15358000
15359000
15360000
15361000
15362000
15363000
15364000
15365000
15365500
15366000
15367000
15368000
15369000
15370000
15371000
15372000
15373000
15374000
15375000
15376000
15377000
15378000
15379000
15380000
15381000
15400000
15401000
15402000
15402499
15402500
15402501
15403000
15404000
15405000
15406000
15407000
15408000
15409000
15410000
15411000
15412000
15413000
15414000
15415000

```

```

ELSE JUMP OUT); IF SC# " " THEN IF SC# "*" THEN GO E; 15416000
I←TALLY; DI←D; SI←SI-1; DS←I OCT); 15417000
END; IF (INDEX AND NOT @1070707)≠0 THEN GO ECH; 15418000
IF JARROW[MIX]≠0 THEN GO ECH; 15419000
IF (NX←INDEX.[45:3]&INDEX[42:39:3]&INDEX[39:33:3]&INDEX[38:29:1
    ])≤20 THEN GO ECH; 15420000
T:=PUTORTAKE(MIX,[PRT[MIX,10]],1&1[2:47:1],0); 15421000
IF T≠NOT FALSE THEN GO ECH; 15421400
T:=PUTORTAKE(MIX,[M[T,MOM=3]],1&1[2:47:1],0); 15421600
IF T≠NOT FALSE OR T,[CF] LSS (PRTROW[MIX] INX NX) THEN GO ECH; 15421800
IF BUFF LSS 0 THEN 15422000
BEGIN 15422200
P(M[T:=PUTORTAKE(MIX,[PRT[MIX,NX]],2,0)); 15422400
FORGETSPACE(T); 15422600
IF P(TOP,XCH,DEL) THEN P(PUTORTAKE(MIX,[PRT[MIX,NX]],
0,DATA),DEL) ELSE GO ECH; 15422800
END ELSE 15423000
BEGIN STREAM(J:=J:=PUTORTAKE(MIX,[JAR[MIX,0]],2,0),MIX, 15424000
INDEX,R:=T:=PUTORTAKE(MIX,[PRT[MIX,NX]],2,0), 15425000
D←DATA←BUFF,[15:15]-1); 15426000
BEGIN SI←J; SI←SI+1; DS←LIT" ";% 15427000
DS←7 CHR; DS←LIT"/"; SI←SI+1;% %WF 15428000
DS←7CHR; DS←LIT"="; SI←LOC MIX; DS←2DEC; %WF 15428100
MIX←DI; DI←DI-2; DS←FILL; DI←MIX; 15429000
DS←3LIT"IR"; SI←SI+4; DS←4 CHR; D←DI; DI←DI-4; 15429500
DS←3 FILL; DI←D; DS←LIT"="; SI←R; 15430000
IF SB THEN % DESCRIPTOR;TYPE OCTAL 15431000
16(DS←3 RESET; 3(IF SB THEN DS←SET ELSE DS←
RESET; SKIP SB)) ELSE 15432000
DS←8 DEC; 15433000
DS←LIT"←"; DI←D; DI←DI+1; DS←7 FILL; 15434000
END; 15435000
SPOUTER(DATA,UNITNO,1); 15436000
FORGETSPACE(T); FORGETSPACE(J); 15437000
END; GO X; 15437100
ECH: PRTGAMES←1; 15437500
X: END; 15438000
PROCEDURE SPOUTMCP(BUFF); VALUE BUFF; REAL BUFF; 15439000
BEGIN REAL X; %021=15500000
DISKWAIT(←(X←SPACE(30)),30,MCPNAMESEG); 15501000
STREAM(ML:=MARKLEVEL,PL:=PATCHLEVEL,LL:=LOCALEVEL 15501100
,N←X+20+2×SYSNO,A←BUFF); 15501500
BEGIN DS←LIT" "; SI←N; SI←SI+1; DS←7 CHR; DS←LIT"/"; 15501600
SI←SI+1; DS←7 CHR; DS←6 LIT" MARK "; 15502000
SI:=LOC ML; IF SC GEQ " " THEN; 15502100
8(IF TOGGLE THEN IF SC="0" THEN SI:=SI+1 ELSE DS:=CHR 15502200
ELSE DS:=CHR); DS:=LIT","; 15502300
SI:=LOC PL; IF SC GEQ " " THEN; 15502400
6(IF TOGGLE THEN IF SC="0" THEN SI:=SI+1 ELSE DS:=CHR 15502500
ELSE DS:=CHR); DS:=2CHR; 15502600
SI:=LOC LL; IF SC GEQ " " THEN; 15502700
8(IF TOGGLE THEN IF SC="0" THEN SI:=SI+1 ELSE DS:=CHR 15502800
ELSE DS:=CHR); 15502900
DS:= 9 LIT" INCLUDES"; 15503000
DS:=12 LIT" B487=S FOR "; 15504000
$ SET OMIT = NOT(TWXONLY) 15505000
DS:=15 LIT"TELETYPES ONLY, "; 15505099
$ POP OMIT 15505100
$ SET OMIT = TWXONLY 15505101
DS:=11 LIT"EVERYTHING, "; 15505199
DS:=11 LIT"EVERYTHING, "; 15505200

```



```

BEGIN
REAL SIZE,LOC,INTWORD,WI,I;
LABEL EXIT;
IF INTSIZE=0 THEN
  BEGIN
    STREAM(BUFF); DS+14 LIT "NO INTRINSICS*";
    SPOUT(BUFF);
    GO EXIT;
  END;
COMMENT MAKE WI INTRINSIC PRESENT;
SIZE := (INTWORD:=INTRNSC[INTRNSC[0]]) INX 0;
LOC := SPACE(SIZE);
$ SET OMIT = NOT(AUXMEM)
IF INTWORD.[3:1] THEN % INTRINSICS ON AUXMEM
  DISKWAIT(=LOC,SIZE,=(O&INTWORD[32:21:12]))
ELSE
$ POP OMIT
DISKWAIT(=LOC,SIZE,INTWORD.[6:27]);
DISKWAIT(=(I:=SPACE(30)),30,0);
STREAM(X:=I+13+5*SYSNO,LOK:=LOC,BUFF);
  BEGIN
    SI:=LOK; SI:=SI+8;
    10(SI:=SI+1;
    7(IF SC="*" THEN JUMP OUT 2 TO L1;
    IF SC="@" THEN SI:=SI+1 ELSE DS:=CHR));
L1: SI:=X;DS:=3LIT" (";
    SI:=SI+1; DS:=7 CHR;DS:=LIT"/";
    SI:=SI+1; DS:=7 CHR;DS:=2LIT")*";
    END STREAM;
  FORGETSPACE(LOC); FORGETSPACE(I);
EXIT:
  END WHATINTRNSIC;
$ SET OMIT = NOT(AUXMEM)
REAL PROCEDURE AUXPRINT(Q); VALUE Q; REAL Q;
% PRINTS AMOUNT OF AUXILIARY MEMORY USED
BEGIN
REAL I, IMAX, C, TC, D, TD, PR, FN, SN, BUFF;
$ SET OMIT = NOT(PACKETS)
REAL UNITNO;
$ POP OMIT
LABEL ERROR,FMT;
IF Q.[1:1] THEN IMAX := MIXMAX ELSE
IF (I:=IMAX:=Q.[CF]) NEQ 0 THEN % NOT MCP AND INTRINSICS
IF I GTR MIXMAX THEN
  BEGIN
ERROR: P(1,RTN);
  END ELSE IF JARROW[I]=0 THEN GO TO ERROR;
IF IMAX=0 OR Q.[1:1] THEN % PRINT MCP AND INTRINSIC USAGE
  BEGIN
    PR:=0; FN:="INTRINS"; SN:="MCP "; GO TO FMT;
  END;
$ SET OMIT = NOT(PACKETS)
IF I#0 THEN UNITNO:=PSEUDOMIX[I];
$ POP OMIT
FOR I:=I STEP 1 UNTIL IMAX DO IF JARROW[I] NEQ 0 THEN
  BEGIN
    TABCNT[I]:=TABCNT[I]+1;
    PR := PUTORTAKE(I,[PROR[I]],1,0);
    FN := PUTORTAKE(I,[JAR[I,0]],1,0);
    SN := PUTORTAKE(I,[JAR[I,1]],1,0);
  END;

```

```

15535000
15536000
15537000
15539000
15540000
15541000
15541100
15542000
15543000
15544000
15545000
15546000
15547000
15547100
15547200
15547300
15547301
15548000
15549000
15550000
15551000
15552000
15552100
15552200
15552300
15552400
15552500
15552600
15552700
15552800
15554000
15555000
15604999
15605000
15605100
15605200
15605300
15605349
15605350
15605351
15605400
15605500
15605600
15605700
15605800
15605900
15606000
15606100
15606200
15606300
15606400
15606449
15606450
15606451
15606500
15606600
15606650
15606700
15606800
15606900

```

```

TABCNT[I]:=TABCNT[I]-1;
FMT: TD := TD + (D := AUXDATA[I]*16);
TC := TC + (C := AUXCODE[I]*16);
STREAM(PR, FN, SN, I, D, C, X:=0, BUFF:=BUFF:=SPACE(7));
BEGIN
  SI:=LOC PR; DS:=6DEC; DS:=LIT" ";
  X:=DI; DI:=DI-7; DS:=5FILL; DI:=X;
  2(SI:=SI+1; DS:=7CHR; DS:=LIT"/");
  DI:=DI-1; DS:=LIT"="; DS:=2DEC;
  X:=DI; DI:=DI-2; DS:=FILL; DI:=X;
  DS:=7LIT" DATA="; DS:=5DEC;
  X:=DI; DI:=DI-5; DS:=4FILL; DI:=X;
  DS:=6LIT" CODE="; DS:=5DEC;
  X:=DI; DI:=DI-5; DS:=4FILL; DI:=X; DS:=LIT"+";
END STREAM;
SPOUTER(BUFF, UNITNO, 1);
END I LOOP;
IF Q,[1:1] THEN % PRINTING FULL MIX
BEGIN
  STREAM(N:=TD+TC, BUFF:=BUFF:=SPACE(10));
  BEGIN
    DS:=21LIT" TOTAL AUX MEM USED: ";
    SI:=LOC N; DS:=6DEC; DS:=2LIT".+";
    DI:=DI-8; DS:=5FILL;
  END STREAM;
  SPOUT(BUFF);
END; % IF FULL MIX
END PROCEDURE AUXPRINT;
$ POP OMIT
REAL PROCEDURE KEYINSCAN(KTR, MIX); REAL KTR, MIX;
BEGIN
  REAL TYPE:=+1, TBLADDR;
  % SCANS INPUT BUFFER FROM SPO
  % RETURNS ERROR FLAG IN MIX,[1:3] ...
  % MIX,[1:1]=FLAG FOR EMPTY BUFFER (GROUP MARK ONLY)
  % MIX,[2:1]=FLAG FOR NO INFO AFTER MIX INDEX
  % MIX,[3:1]=FLAG FOR QMARK (CC) INPUT AS FIRST CHARACTER
  % KTR IS INITIALLY THE ADDRESS OF SPO INPUT BUFFER
  % KTR IS ASSIGNED NEXT CHARACTER LOCATION AFTER SCAN
  % TYPE,[CF] IS ASSIGNED TABLE LOCATION (MIXMSG OR INFMSG)
  % TYPE,[1:5] IS ASSIGNED PROCEDURE NUMBER
  % TYPE,[6:6] IS ASSIGNED MIXCODE
  STREAM(MIX:=63, BUFF:=KTR); % SCAN INPUT BUFFER
  BEGIN
    SI:=BUFF;
    DI:=BUFF; DI:=DI-1; DS:=LIT"<"; % BACKSPACE CHARACTER
    B(60(IF SC="+" THEN % END OF INPUT STRING
    BEGIN
      DS:=CHR; JUMP OUT 2 TO L;
    END;
    IF SC="<" THEN % BACK SPACE CHARACTER
    BEGIN
      DI:=DI-1; IF SC NEQ DC THEN DI:=DI-1;
    END
    ELSE DS:=CHR)); % END OF BACKSPACE CHECK
  L: SI:=BUFF; DI:=LOC MIX; % CHECK FOR MIX INDEX
  L1: IF SC=" " THEN
    BEGIN
      SI:=SI+1; GO TO L1;
    END;

```

```

15606950
15607000
15607100
15607200
15607300
15607400
15607500
15607600
15607700
15607800
15607900
15608000
15608100
15608200
15608300
15608400
15608500
15608600
15608700
15608800
15608900
15609000
15609100
15609200
15609300
15609400
15609500
15609600
15609601
16029000
16029100
16029200
16029300
16029400
16029500
16029600
16029700
16029800
16029900
16030000
16030100
16030200
16030300
16030400
16030500
16030600
16030700
16030800
16030900
16031000
16031100
16031200
16031300
16031400
16031500
16031600
16031700
16031800
16031900
16032000

```



```

        SI:=SI+1; GO TO L;
      END;
      BUFF:=SI;
      XT: END STREAM STATEMENT;
        P( [KTR],STD, ,TYPE,STD);
        FORGETSPACE(TBLADDR);
      END % IF NOT QMARK, EMPTY OR ERROR
    ELSE % QMARK, EMPTY OR ERROR
      IF MIX,[3:1] THEN % QMARK
        BEGIN MIX:=63;
          TYPE:=VCC&1[[1:43:5]];
        END
      ELSE TYPE:=0;
    END PROCEDURE KEYINSCAN;
PROCEDURE KEYINO(B,KTRX); VALUE B,KTRX; REAL B,KTRX;

```

```

BEGIN
REAL BUFF, KTR, TYPE, MIX, A, I, J, K;

```

```

LABEL CUTY, RXIT, ERROR, FORGET, EXIT
,AX, IL, QT, OU, WY, RY, DS, RS, SS, DP
,DD, DB, ST, CM, MF, SV, CL, BK

```

```

;
SWITCH S:= ERROR
,AX, IL, IL, QT, OU, WY, RY, DS, DS, RS
,SS, DP, DD, DB, ST, CM, MF, SV, CL, BK
,RXIT, RY, RXIT, RXIT, RXIT

```

```

;
BUFF:=KTRX,[15:15];
MIX :=KTRX,[ 9:6 ];
TYPE:=KTRX,[ 2:7 ];
KTR :=KTRX,[15:33];
GO TO S[TYPE];

```

```

AX:
I := BUFF;
GO TO RXIT;

```

```

IL:
IF(I:=ANVIL(TYPE=2,KTR)) GTR 35 THEN
IF I < 42 THEN GO TO ERROR;
TYPE := VIL;
IF I GTR 35 THEN BUFF:=I;
GO TO RXIT;

```

```

OU:
STREAM(A:="LP" , B:="MT", C:="DK", D:="CP", KTR);
BEGIN
SI := KTR;
DI := LOC A; DI := DI+6;
TALLY:=1; IF SC="+" THEN GO TO XT;
TALLY:=2; IF 2 SC=DC THEN GO TO XT;
TALLY:=3; SI:=SI-2; DI:=DI+14; IF 2 SC=DC THEN GO TO XT;
TALLY:=4; SI:=SI-2; DI:=DI+6; IF 2 SC=DC THEN GO TO XT;
TALLY:=5; SI:=SI-2; DI:=DI+6; IF 2 SC=DC THEN GO TO XT;

```

```

16037200
16037300
16037400
16037500
16037600
16037650
16037700
16037720
16037740
16037760
16037780
16037800
16037850
16037900
16038000
16039000
16040000
16041000
16042000
16043000
16044000
16045000
16046000
16047000
16048000
16049000
16050000
16051000
16052000
16053000
16054000
16055000
16056000
16057000
16058000
16059000
16060000
16061000
16062000
16063000
16064000
16065000
16066000
16067000
16068000
16069000
16070000
16071000
16072000
16073000
16074000
16075000
16076000
16077000
16078000
16079000
16080000
16081000
16082000
16083000

```



```

        TALLY:=0;
XT: A := TALLY;
    END;
    IF(I:=P) = 0 THEN GO TO ERROR;
    GO TO RXIT;
WY:
    IF MIX LSS 63 THEN GO TO RXIT; % <MIX> WY
    A:=0; % THIS PRINTS OUT TOTAL LIST OF WAITING JOBS ON WY
    FOR I:=0 STEP 1 UNTIL MIXMAX DO
        IF REPLY[I] LSS 0 THEN
            BEGIN
                REPLY[A:=I]:=VWY; BRINGBACK(I);
            END;
        IF A NEQ 0 THEN GO TO FORGET;
        M[BUFF:=1]:=FLAG(@0564434360606060);
        GO TO ERROR; % SPOUT MESSAGE
RY:
    IF (I:=FORMESS(KTR,TYPE=VFM)) LSS 0 THEN GO TO FORGET;
    IF I GTR 31 THEN GO TO ERROR ELSE GO TO RXIT;
RS:
    LINEMESSAGES(KTR);
    GO TO EXIT;
SS:
    IF CANDYINX NEQ 0 THEN CALLCANDE(KTR,"SS") ELSE
        LINEMESSAGES(KTR);
    GO TO EXIT;
DP:
    $ SET OMIT = NOT(DEBUGGING OR DUMP)
    STREAM(A:="LP", KTR : B:="MT");
        BEGIN
            SI := KTR;
            DI := LOC A; DI := DI + 6; TALLY:=1;
            IF 2 SC=DC THEN GO TO XT;
            DI := DI + 22; SI := SI - 2; TALLY:=2;
            IF 2 SC=DC THEN GO TO XT;
            TALLY:=0;
XT: A := TALLY; KTR := SI;
        END STREAM STATEMENT;
        IF (A:=P([KTR],STD))=0 THEN GO TO ERROR;
        IF A#1 THEN
            BEGIN
                PRINTCORE(MIX); GO TO FORGET;
            END;
        DUMPCORE(KTR&BUFF[15:33:15]);
    $ POP OMIT
    GO TO EXIT;
DD:
    $ SET OMIT = NOT(DEBUGGING)
    DDT;
    $ POP OMIT
    GO TO FORGET;
DB:
    $ SET OMIT = NOT(DEBUGGING)
    DISKBUG;
    $ POP OMIT
    GO TO FORGET;
ST:
    IF REPLY[MIX] = (VWY&VOK[36:42:6]) OR JARROW[MIX]=0 THEN GO ERROR;
    IF NOTERMSET(MIX) THEN PRTROW[MIX],[PSF]:=2;
    GO FORGET;

```

```

16084000
16085000
16086000
16087000
16088000
16089000
16090000
16091000
16092000
16093000
16094000
16095000
16096000
16097000
16098000
16099000
16100000
16101000
16102000
16103000
16104000
16105000
16106000
16107000
16108000
16109000
16110000
16111000
16112000
16113000
16114000
16115000
16116000
16117000
16118000
16119000
16120000
16121000
16122000
16123000
16124000
16125000
16126000
16127000
16127001
16128000
16129000
16130000
16131000
16131001
16132000
16133000
16134000
16135000
16135001
16136000
16137000
16138000
16139000
16140000

```

CM:		16141000
	CHANGEMCP(KTR);	16142000
	GO TO EXIT;	16143000
MF:		16144000
	FENCEMOVER(KTR, BUFF);	16145000
	GO TO EXIT;	16146000
SV:		16147000
	SAVETHEUNIT(KTR);	16148000
	GO TO FORGET;	16149000
QT:		16150000
	IF MIX LSS 63 THEN % MIX INDEX SPECIFIED	16153000
	BEGIN	16154000
	CUTY: I:=PUTORTAKE(MIX,[JAR[MIX,0]],2,0);	16155000
	J:=(M[I] EQV "PRNPBT ") = NOT 0) AND	16156000
	((M[I+1] EQV "DISK ") = NOT 0);	16157000
	FORGETSPACE(I);	16158000
	IF J THEN	16159000
	BEGIN	16160000
	REPLY[MIX]:=TYPE;	16161000
	STREAM(A:=0, B:=0 ; KTR);	16166000
	BEGIN	16167000
	SI:=KTR;	16168000
	IF SC="+" THEN TALLY:=2 ELSE	16169000
	IF SC="=" THEN TALLY:=3 ELSE GO XT;	16170000
B2:	SI:=SI+1; IF SC=" " THEN GO TO B2;	16171000
	B:=TALLY; TALLY:=0;	16172000
	6(IF SC LSS "0" THEN JUMP OUT; SI:=SI+1; TALLY:=TALLY+1);	16173000
	KTR:=TALLY; DI:=LOC A; SII:=SI-KTR; DS:=KTR OCT;	16174000
XT:	END STREAM STATEMENT;	16175000
	NT2:=P;	16176000
	NT1:=P;	16177000
	PRT[MIX,@25]:=5&NT1[9:24:24]&NT2[1:46:2];	16178000
	GO TO FORGET;	16179000
	END	16180000
	ELSE GO TO ERROR; % NOT PRNPBT	16181000
	END;	16181500
CL:		16182000
	% QT OR CL LINE OR PERIPHERAL UNIT	16183000
	STREAM(W:=0 ; KTR);	16184000
	BEGIN	16185000
	SI:=KTR;	16186000
	IF SC GEQ "0" THEN IF SC LEQ "9" THEN TALLY:=1;	16187000
	W:=TALLY;	16188000
	END STREAM STATEMENT;	16189000
	IF P THEN	16190000
	BEGIN	16191000
	LINECLEAR(KTR); GO FORGET;	16192000
	END;	16193000
	IF (I:=UNITIN(TINU,KTR)) LSS 36 THEN % CHG IF MORE PSEUDORDRS	16194000
	IF (MIX:= RDCTABLE[I],[B:6]) NEQ 0 THEN	16195000
	BEGIN	16195500
	TABCNT[MIX]:=TABCNT[MIX]+1;	16195750
	IF TYPE=4 THEN GO TO CUTY	16196000
	ELSE GO TO DS; % CLEAR UNIT IN USE BY JOB	16197000
	END;	16197500
	% CLEAR UNIT NOT IN USE BY JOB	16198000
\$	SET OMIT = NOT(SHAREDISK)	16199000
	IF I LSS 40 AND I GTR 35 THEN	16200000
	IF TYPE=19 THEN % CL	16201000
	BEGIN	16202000

IF (I:=I-36)=SYSNO OR I GEQ SYSMAX THEN GO ERROR;	16203000
CLEANOUT(I OR MEMORY);	16204000
GO TO FORGET;	16205000
END;	16206000
\$ POP OMIT	16206001
IF TYPE=4 OR (I GTR 29) THEN GO TO ERROR; % QT OR PSEUDO UNIT	16207000
LABELTABLE[I]:=P(DUP,LOD,SSP); % MARK IT NOT IN USE	16208000
MIX:=63; GO TO RY;	16209000
DS:	16210000
IF JARROW[MIX] NEQ 0 THEN	16211000
BEGIN	16212000
TERMINATE(MIX&3[18:33:15]);	16213000
HALT;	16214000
NOPROCESSTOG:=NOPROCESSTOG-1;	16215000
JAR[MIX,6],[1:1]:=((TYPE=9) OR (TYPE=19)); % DS=8, SD=9 CL=19	16216000
GO TO FORGET;	16217000
END;	16218000
GO TO ERROR;	16219000
BK:	16220000
IF (I:= MESSAGEHOLDER,[CF]) NEQ 0 THEN	16221000
BEGIN	16222000
IF (J:= M[I],[FF]) NEQ 0 THEN	16223000
DO BEGIN	16224000
A:=M[J];	16225000
IF (A,[4:5]=0 AND MIX=63) OR (A,[4:5]=MIX AND MIX NEQ 63) THEN	16226000
BEGIN	16227000
M[I]:=P(DUP,LOD)&A[18:18:15];	16228000
NUMESS:=NUMESS-1;	16229000
FORGETSPACE(J+1);	16230000
END	16231000
ELSE I:=J;	16232000
END UNTIL (J:=A,[FF])=0;	16233000
MESSAGEHOLDER,[FF]:=I;	16234000
END;	16235000
IF SPOWORD LSS 0 THEN % GO CLEAN OUT TANK	16236000
BEGIN	16237000
IF STABLE[K:=ABS(SPOWORD)],OUTPUTANKING THEN	16238000
BEGIN	16239000
WHILE NOT (J:=TANKS[K]),[1:1] DO SLEEP([TANKS[K]],-0);	16240000
STABLE[K],OUTPUTANKING:=0;	16241000
IF J,[2:8] NEQ 0 OR TAILOUT = K THEN	16242000
BEGIN	16243000
A:=0;	16244000
DO I:=A UNTIL (A:=TANKS[A],[2:8])=K;	16245000
TANKS[I]:=(*P(DUP))&J[2:2:8];	16246000
IF K=TAILOUT THEN TAILOUT:=I;	16247000
END;	16248000
END;	16249000
TANKS[K]:=NABS(0);	16250000
\$ SET OMIT = TWXONLY	16251000
IF STABLE[K],STATIONTYPE=TC500 THEN	16252000
TNAOG[SEQARRAY[K],[26:6],[1:13]]:=0 ELSE	16253000
BEGIN	16254000
TNAOG[SEQARRAY[K],[26:6]]:=(*P(DUP))&P(0,XCH)[14:14:14];	16255000
STABLE[K],[22:2]:=0;	16256000
END;	16257000
\$ POP OMIT	16257001
END;	16258000
MIX:=63;	16258500
GO TO FORGET;	16259000

```

RXIT:
  REPLY[MIX]:=TYPE & I[18:33:15];
  BRINGBACK(MIX);
  IF I NEQ BUFF THEN
    BEGIN
FORGET:
  STREAM(T:=BUFF-1); DS:=LIT"+";
ERROR:
  SPOUT(BUFF-1);
  END;
EXIT:
  IF (MIX#0) AND (MIX<63) THEN TABCNT[MIX]:=TABCNT[MIX]-1;
  END PROCEDURE KEYINO;
PROCEDURE KEYINI(B,KTRX); VALUE B,KTRX; REAL B,KTRX;
  BEGIN
    REAL BUFF, KTR, TYPE, MIX, A, I, J, K;

    LABEL COUT, ERROR, FORGET, EXIT
      ,DT, WD, TR, WT, TF, WM, CX, CE, CC, OL
      ,PB, BS, SC, RN, LD, RD, ED, SI, CA, SQ
      ,CS, HS

      ;
    SWITCH SI=ERROR
      ,DT, WD, TR, WT, TF, WM, CX, CE, CC
      ,OL, PB, BS, BS, SC, RN, LD, RD, ED, SI
      ,CA, CA, SQ, CS, HS

      ;

    BUFF:=KTRX,[15:15];
    MIX :=KTRX,[ 9:6 ];
    TYPE:=KTRX,[ 2:7 ];
    KTR :=KTRX,[15:33];
    GO TO S[TYPE];
DT:
  SETDATE(KTR);
  GO TO EXIT;
WD:
  GIMEDATE(BUFF-1,1);
  GO TO EXIT;
TR:
  SETIME(KTR);
  GO TO EXIT;
WT:
  TIMEOUT (BUFF-1);
  GO TO EXIT;
TF:
  CHANGEFACTOR(KTR, TYPE=5); % TF=5, SF=6
  GO TO EXIT;
WM:
  SPOUTMCP(BUFF-1);
  GO TO EXIT;
CX:
  IF REMOTE THEN
    BEGIN
      CALLCANDE(KTR, 0);

```

```

16349000
16350000
16351000
16352000
16353000
16354000
16355000
16356000
16357000
16358000
16359000
16359500
16360000
16361000
16362000
16363000
16364000
16365000
16366000
16367000
16368000
16369000
16370000
16371000
16372000
16373000
16374000
16375000
16376000
16377000
16378000
16379000
16380000
16381000
16382000
16383000
16384000
16385000
16386000
16387000
16388000
16389000
16390000
16391000
16392000
16393000
16394000
16395000
16396000
16397000
16398000
16399000
16400000
16401000
16402000
16403000
16404000
16405000
16406000
16407000

```

*these might be  
useful examples*



GO TO EXIT;	16408000
END ELSE GO TO ERROR;	16409000
CE:	16410000
STREAM(KI=KTR:=SPACE(8)+2);	16411000
DS:=45 LIT "CC RUN CANDE/TSHARER;STACK=200;CORE=4000;END*";	16412000
GO TO COUT;	16413000
CC:	16414000
A:=M[BUFF=3],[CF]=BUFF; % WDS IN MESSAGE	16414100
STREAM(BUFF, BL:=A>8, KTR:=(KTR:=SPACE(A+2)+2));	16415000
BEGIN	16416000
SI:=BUFF;	16417000
BL(36(DS:=2LIT " ")); DI:=KTR);	16417100
IF SC NEQ "*" THEN	16418000
BEGIN	16419000
DS:=CHR;	16420000
L: IF SC NEQ "*" THEN	16421000
BEGIN	16422000
IF SC NEQ @14 THEN DS:=CHR ELSE SI:=SI+1;	16423000
GO TO L;	16424000
END;	16425000
END;	16426000
DS:=CHR;	16427000
END;	16428000
COUT:	16429000
M[KTR=4],[9:6]:=0;	16430000
CCARD(KTR&(IF (ABS(B) GTR 1) THEN 30 ELSE 25)[3:43:5]);	16431000
GO TO FORGET;	16432000
OL:	16433000
OUTPUTLABEL(KTR);	16434000
GO TO EXIT;	16435000
PB:	16436000
PRINTBACKUP(KTR);	16437000
GO TO EXIT;	16438000
BS:	16439000
SPOSET(TYPE=13, KTR); % BS=13, US=14	16440000
GO TO EXIT;	16441000
SC:	16442000
STREAM(SPO:=SPOWORD GEQ 0, S:= ABS(SPOWORD), B:= BUFF=1);	16443000
BEGIN	16444000
DS:= 18 LIT " SPO CONSOLES ARE:";	16445000
SPO(DS:=4 LIT "SPO ");	16446000
SI:= LOC S; DS:= 2 DEC; DS:= LIT "*";	16447000
DI:= DI*3; DS:= 2 FILL;	16448000
END STREAM STATEMENT;	16449000
SPOUT(BUFF=1);	16450000
GO TO EXIT;	16451000
RN:	16452000
RUNTHEDECK(KTR);	16453000
GO TO EXIT;	16454000
LD:	16455000
STARTLOADN(KTR);	16456000
GO TO EXIT;	16457000
RD:	16458000
DECKREMOVER(KTR);	16459000
GO TO EXIT;	16460000
ED:	16461000
EXTERNALEND(KTR);	16462000
GO TO EXIT;	16463000
SI:	16464000
% SET OMIT = NOT(STATISTICS)	16465000

STREAM(A:=0 : KTR);	16466000
BEGIN	16467000
SI:=KTR; TALLY:=1;	16468000
IF SC="*" THEN GO TO L2;	16469000
IF SC GEQ "0" THEN	16470000
BEGIN	16471000
SI:=SI+1;	16472000
IF SC GEQ "0" THEN	16473000
BEGIN	16474000
SI:=SI-1; DI:=LOC A; DS:=2 OCT;	16475000
END	16476000
ELSE	16477000
BEGIN	16478000
SI:=SI-1; DI:=LOC A; DS:= OCT;	16479000
END;	16480000
END	16481000
ELSE GO TO L2;	16482000
IF SC NEQ "*" THEN	16483000
BEGIN	16484000
L2: TALLY:=0; A:=TALLY	16485000
END;	16486000
END;	16487000
IF (I:=P)=0 THEN GO TO ERROR;	16488000
INTERVAL:=I*3600;	16489000
STREAM(A:=I, B:=BUFF-1);	16490000
BEGIN	16491000
SI:=LOC A; DS:=23LIT"NEW TIMING INTERVAL IS ";	16492000
DS:=2 DEC; DS:= 9LIT" MINUTES*";	16493000
END;	16494000
COUNTARRAY[29]:=XCLOCK;	16495000
\$ POP OMIT	16495001
GO TO ERROR; % SPOUT MESSAGE	16496000
CA:	16497000
\$ SET OMIT = NOT(AUXMEM)	16498000
CHANGEAUXFILES(KTR,TYPE=21); % WA=21, CA=22	16499000
\$ POP OMIT	16499001
GO TO ERROR; % SPOUT AUX MESSAGE OR ERROR MESSAGE	16500000
SQ:	16504000
STREAM(TYPE:=0;INFO1:="STOPOKN",INFO2:=@2567630000000000,	16504100
KTR);	16504200
BEGIN	16504300
SI:=KTR; DI:=LOC INFO1; DI:=DI+1; TALLY:=1;	16504400
IF 4 SC=DC THEN GO TO EXT;	16504500
SI:=SI-4; TALLY:=TALLY+1;	16504600
IF 2 SC=DC THEN GO TO EXT;	16504700
SI:=SI-2; TALLY:=TALLY+2;	16504800
IF 4 SC=DC THEN GO TO EXT;	16504900
TALLY:=TALLY+4;	16505000
EXT: TYPE:=TALLY;	16505100
END;	16505200
IF P(M[P(,DISKSQUASH)],TOP) THEN IF P(P,[FF] AND P,DUP)#0 THEN	16505300
P(,DISKSQUASH,STD) ELSE GO TO ERROR ELSE IF P(XCH)=8 THEN	16505400
BEGIN	16505500
FORK(P(,DISKSQUASH),KTR,0,192,1);	16505600
GO TO EXIT;	16505700
END ELSE GO TO ERROR;	16505800
GO FORGET;	16505900
HS:	16506000
\$ SET OMIT = NOT SEPTICTANK	16506990
KTR:=KTR;	16507000

```

$ POP OMIT
CS:
$ SET OMIT = NOT SEPTICTANK
  RUNSEPTIC(KTR);
$ POP OMIT
  GO TO EXIT;
FORGET;
  STREAM(TI=BUFF=1); DS:=LIT"@";
ERROR;
  SPOUT(BUFF=1);
EXIT;
  IF (MIX#0) AND (MIX<63) THEN TABCNT[MIX]:=TABCNT[MIX]-1;
  END PROCEDURE KEYIN1;
PROCEDURE KEYIN2(KTRX); VALUE KTRX; REAL KTRX;
% AUXILIARY PROCEDURE TO "KEYIN",
% THIS PROCEDURE IS CALLED AS AN INDEPENDENT RUNNER FROM
% PROCEDURE "KEYIN";

  BEGIN
  REAL BUFF, KTR, TYPE, MIX, A, I, J, K;
  REAL R, R1, R2, R3, R4;
  INTEGER INT1=NT1, INT2=A, INT3=J, INT4=R4;
  ARRAY UT = R3[*]
$ SET OMIT = NOT SHAREDISK
  ,U = R2[*]
$ POP OMIT
  ;
$ SET OMIT = SHAREDISK
  DEFINE U = AVTABLE#;
$ POP OMIT
  REAL HN1 = MIX, HN2 = TYPE;
  NAME SEGDICT = R3;
  REAL SEG=I, ADR=J, LOCN=K, HALTED=R1; % FOR RA REQUEST
$ SET OMIT = NOT(PACKETS)
  REAL UNITNO;
$ POP OMIT
  LABEL PGA, FERGIT, FORGET, ERROR, EXIT
    ,MX, RO, TS, TI, PR, LF, LC, LS, EX, PD
    ,OT, IT, PO, PG, AU, MS, LN, CD, FE, CU
    ,SY, OC, RW, CI, SM, CT, WU, XD, WI, MC
    ,HD,RA,RAEND

  ;

  SWITCH S:= ERROR
    ,MX, RO, RO, TS, TS, TS, TS, TI, PR, LF
    ,LC, LS, EX, PD, OT, OT, IT, PO, PO, PG
    ,AU, MS, LN, CD, FE, CU, SY, OC, RW, CI
    ,SM,CT,CT,CT,WU,XD,XD,WI,MC,CD
    ,HD,RA

  ;

  BUFF:=KTRX,[15:15];
  MIX :=KTRX,[ 9:6 ];
  TYPE:=KTRX,[ 2:7 ];
  KTR :=KTRX,[15:33];

```

```

16507010
16508000
16508990
16509000
16509010
16510000
16604000
16605000
16606000
16607000
16608000
16608500
16609000
16610000
16611000
16612000
16613000
16614000
16615000
16616000
16616100
16616200
16616300
16616400
16616500
16616600
16616700
16616800
16616900
16617000
16617100
16617200
16617300
16617499
16617500
16617501
16618000
16619000
16620000
16621000
16622000
16623000
16624000
16625000
16626000
16627000
16628000
16629000
16630000
16631000
16632000
16633000
16634000
16635000
16636000
16637000
16638000
16639000
16640000
16641000

```

\$ SET OMIT = NOT(PACKETS)	16641099
IF MIX#63 THEN UNITNO:=PSEUDOMIX[MIX];	16641100
\$ POP OMIT	16641101
GO TO S[TYPE];	16642000
MX:	16643000
MIXPRINT(BUFF=1);	16644000
GO TO EXIT;	16645000
RO:	16646000
CHANGEOPTION(KTR, TYPE=2); % RO=2, SO=3	16647000
GO TO EXIT;	16648000
TS:	16649000
IF (TYPE=6 OR TYPE=7) AND (MIX=63) THEN % ES OR XS SCHEDULE TASK	16650000
SCHEDLOOK(KTR,TYPE=9) % ES=6, XS=9	16651000
ELSE SHEETDIDDLER(KTR,TYPE,MIX); % TS=4, PS=5, ES=6, XS=7	16652000
MIX:=63;	16652500
GO TO EXIT;	16653000
TI:	16654000
TIMEUSED(BUFF=1,MIX);	16655000
GO TO EXIT;	16656000
PR:	16657000
CHANGEPRIORITY(KTR,MIX);	16658000
GO TO EXIT;	16659000
LF:	16660000
I:=3; GO TO PD;	16661000
LC:	16662000
I:=2; GO TO PD;	16663000
LS:	16664000
I:=4; GO TO PD;	16665000
EX:	16666000
I:=1; KTR:= -KTR;	16667000
PD:	16668000
PRINTDIRECTORY(KTR&I[9:42:6]);	16669000
GO TO EXIT;	16670000
OT:	16671000
IF TYPE=16 THEN KTR:= -KTR; % OT=15, IN=16	16672000
IF PRGAMES(KTR,MIX) THEN GO TO ERROR ELSE	16673000
IF KTR LSS 0 THEN GO FORGET ELSE GO EXIT;	16674000
IT:	16675000
IF NOT (I:=PUTORTAKE(MIX,[JAR[MIX,2]],1,0)),[4:1] THEN GO TO ERROR;	16676000
P(PUTORTAKE(MIX,[JAR[MIX,2]],0,I&1[5:47:1]),DEL);	16677000
GO TO FORGET;	16678000
PO:	16679000
TYPOP(KTR,TYPE=19); % TO=18, PO=19	16680000
GO TO EXIT;	16681000
PG:	16682000
STREAM(Y:=KTR);	16683000
BEGIN	16684000
SI:=Y;	16685000
LAI IF SC NEQ "*" THEN	16686000
BEGIN	16687000
SI:=SI+1; DI:=DI+1; GO TO LAI	16688000
END	16689000
ELSE DS:=4LIT"++++";	16690000
END;	16691000
PGA: STREAM(Y:=0, KTR: AI=A:=SPACE(12)+1);	16692000
BEGIN	16693000
SI:=KTR;	16694000
LI IF SC="*" THEN	16695000
BEGIN	16696000
SI:=SI+1; GO TO LI	16697000



END;	16698000
IF SC="+" THEN TALLY := 1 ELSE	16699000
IF SC="0" THEN TALLY := 1 ELSE	16700000
BEGIN	16701000
DS:=3CHR;	16702000
IF SC="-" THEN	16703000
BEGIN	16704000
DS:=CHR;	16705000
LL: IF SC=" " THEN	16706000
BEGIN	16707000
SI:=SI+1; GO TO LL;	16708000
END;	16709000
5(IF SC GEQ 0 THEN DS:=CHR ELSE JUMP OUT);	16710000
END;	16711000
DS:=LIT"+"; KTR:=SI;	16712000
END;	16713000
Y:= TALLY;	16714000
END STREAM STATEMENT;	16715000
IF P([KTR],STD) THEN	16716000
BEGIN	16717000
FORGETSPACE(A=1); GO TO FORGET;	16718000
END;	16719000
A:=A&A[15:33:15];	16720000
TAPEPURGE(A);	16721000
GO TO PGA;	16722000
AU:	16723000
\$ SET OMIT = NOT(AUXMEM)	16724000
IF AUXPRINT(IF MIX=63 THEN =0 ELSE MIX) THEN GO TO ERROR;	16725000
\$ POP OMIT	16725001
GO TO FORGET;	16726000
MS:	16727000
\$ SET OMIT = NOT(AUXMEM OR MONITOR)	16728000
STREAM(TYPE:=0 : KTR);	16729000
BEGIN	16730000
SI:=KTR; TALLY:=2;	16731000
IF SC="R" THEN TALLY:=1 ELSE IF SC="S" THEN TALLY:=0;	16732000
L2: TYPE:=TALLY;	16733000
END STREAM STATEMENT;	16734000
IF (TYPE=P) GTR 1 THEN GO TO ERROR;	16735000
IF TYPE=0 AND CTABLE[4],[1:1] THEN GO TO ERROR; % MONITOR SET	16736000
IF TYPE=1 AND NOT(CTABLE[4],[1:1]) THEN GO TO ERROR;	16737000
SETMONITORFILE(TYPE);	16738000
\$ POP OMIT	16738001
GO TO FORGET;	16739000
LN:	16740000
STREAM(A:=0 : KTR);	16741000
BEGIN	16742000
SI:=KTR; DI:=LOC A; DI:=DI+6;	16743000
DS:=2 CHR;	16744000
END STREAM STATEMENT;	16745000
IF (I=P)="DK" THEN LOGDISK ELSE	16746000
IF I,[36:6]=@37 THEN LOGOUT ELSE	16747000
IF I="ML" THEN FORK(P,LOGOUTMAINT),0,0,128,0) ELSE	16748000
GO TO ERROR;	16749000
GO TO FORGET;	16750000
ED:	16751000
TABLEOFCONTENTS(KTR,TYPE=40);	16752000
GO TO FORGET;	16753000
FE:	16754000
I:= GETSPACE(35,9,5)+2;	16755000

STREAM(KTR;DI:=I+2);	16756000
BEGIN	16757000
SI:=KTR;	16758000
4(63(IF SC NEQ "*" THEN DS:=CHR ELSE JUMP OUT 2 TO LL));	16759000
LL: DS:=LIT"*"; DI:=DI-1; KTR:=DI;	16760000
END STREAM STATEMENT;	16761000
K:= P INX 0;	16762000
M[I]:= (K-I) DIV 5;	16763000
STREAM(DATE, AI:=I+1);	16764000
BEGIN	16765000
SI:=LOC DATE; DS:=8OCT;	16766000
END STREAM STATEMENT;	16767000
LINKUP(19,I);	16768000
GO TO FORGET;	16769000
CU:	16770000
COREPRINT(IF MIX LSS 63 THEN MIX ELSE NABS(0));	16771000
GO TO FORGET;	16772000
SY:	16773000
\$ SET OMIT = NOT(STATISTICS)	16774000
SAVESTATISTICS;	16775000
\$ POP OMIT	16775001
GO TO FORGET;	16776000
OC:	16777000
STREAM(WI:=1, KTR ; BI:=BUFF);	16778000
BEGIN	16779000
SI:=KTR;	16780000
2(36(IF SC="*" THEN	16781000
BEGIN	16782000
WI:=TALLY; DS:=CHR; JUMP OUT 2 TO XT;	16783000
END ELSE DS:=CHR));	16784000
SI:=SI-1; DI:=DI-1; DS:=LIT"*";	16785000
XT: KTR:=SI;	16786000
END STREAM STATEMENT;	16787000
P([KTR],STD, ,1,STD);	16788000
MAKELOG(BUFF=1,OCM);	16789000
IF I THEN GO TO OC ELSE GO TO FORGET;	16790000
RW:	16791000
REWINDANDLOCK(KTR);	16792000
GO TO EXIT;	16793000
RI:	16794000
CHANGEINTRINSICFILE(KTR);	16795000
GO TO EXIT;	16796000
SM:	16797000
K:=KTR&(I:=MIX*(MIX NEQ 63))[9:42:6];	16798000
WHATSGOINGON(K);	16799000
GO TO EXIT;	16800000
CT:	16801000
TIMERELAXER(KTR,TYPE,MIX); % CT=32, XT=33, TL=34	16802000
GO TO EXIT;	16803000
WU:	16804000
I:=KTR&(MIX*(MIX NEQ 63))[9:42:6];	16805000
WHATSIT(I);	16806000
GO EXIT;	16807000
XD:	16808000
IF TYPE=37 THEN KTR,[CF]:=0; % XD=36, MR=37	16809000
DKBUSINESS(KTR);	16810000
GO TO EXIT;	16811000
WI:	16812000
WHATINTRNSIC(BUFF=1);	16813000
SPOUT(BUFF=1);	16813100

GO TO EXIT;	16814000
MC:	16815000
NAMEID(I,KTR); NAMEID(J,KTR); NAMEID(J,KTR);	16816000
IF J.[6:6]="*" THEN GO TO ERROR;	16817000
IF (A:=DIRECTORYSEARCH(I,"J,4")) GEQ 64 THEN	16818000
BEGIN	16819000
IF J NEQ "DISK " THEN	16820000
IF(K:=DIRECTORYSEARCH(I,"DISK ",5)) NEQ 0 THEN	16821000
BEGIN	16822000
P(DIRECTORYSEARCH("I,J,14),DEL);	16823000
FORGETSPACE(A);	16824000
FORGETSPACE(K);	16825000
LBMESS(I,J,"9,29,0,0,1);	16826000
GO FERGIT;	16827000
END	16828000
ELSE	16829000
BEGIN	16830000
M[A INX 4]:=(P(DUP))&2[1:46:2]&1[8:47:1];	16831000
A:=A&EUF("I,"DISK ",A INX 0-1)[18:33:15];	16832000
FORGETSPACE(DIRECTORYSEARCH(I,J,8));	16834000
END ELSE M[A INX 4]:=(P(DUP))&2[1:46:2]&1[8:47:1];	16835000
HEADERUNLOCK(I,"DISK ",A);	16836000
LBMESS(I,J,54,I,"DISK ",0,1);	16837000
END	16838000
ELSE LBMESS(I,J,"9,((A=1)*30)+15,0,0,1);	16839000
FERGIT;	16841600
FORGETSPACE(BUFF=1);	16841800
GO TO EXIT;	16842000
HD:	16843000
STREAM(EU:=1,ERRTOG:=0:EULIT:=@2564000000000000,CX:=0,	16843100
K:=KTR);	16843200
BEGIN	16843300
SI:=K; GO TO L1;	16843400
LO: IF SC=" " THEN BEGIN SI:=SI+1; GO TO LO END; CI:=CX;	16843500
L1: CX:=CI; GO TO LO;	16843600
IF SC="*" THEN GO EXT;	16843700
DI:=LOC EULIT; TALLY:=1;	16843800
IF 2 SC=DC THEN % AN EU SPECIFIED	16843900
BEGIN	16844000
CX:=CI; GO TO LO;	16844100
IF SC GEQ 0 THEN IF SC<12 THEN	16844200
BEGIN	16844300
SI:=SI+1; DI:=LOC EU;	16844400
IF SC GEQ 0 THEN IF SC<12 THEN	16844500
TALLY:=2 ELSE GO TO ERR;	16844600
SI:=SI-1; CX:=TALLY;	16844700
DS:=CX OCT; GO EXT;	16844800
END ;	16844900
END;	16845000
ERR: ERRTOG:=TALLY;	16845100
EXT:	16845200
END;	16845300
IF P THEN GO TO ERROR;	16845400
IF (HN1:=P+1)>0 THEN IF HN1 LEQ NEUP.[FF] THEN	16845500
HN2:=HN1 ELSE GO TO ERROR ELSE	16845600
BEGIN	16845700
HN1:=1;	16845800
HN2:=NEUP.[FF];	16845900
END;	16846000
\$ SET OMIT = NOT SHAREDISK	16846100

FIXARRAY(U,R1,30);	16846200
DISKWAIT(-R1,30,USERDISKBOTTOM);	16846300
\$ POP OMIT	16846400
FOR I:=HN1 STEP 1 UNTIL HN2 DO	16846500
IF NOT (NT2:=U[I]),EUNP THEN % NOT A DUMMY EU	16846600
BEGIN	16846700
INT4:=(INT1:=NT2,STARTWRD) MOD 30;	16846800
INT2:=30-(K1:=(NT2 AND NUMENTM)+R4) MOD 30+K;	16846900
J:=NT1 DIV 30+USERDISKBOTTOM;	16847000
FIXARRAY(UT,R,A);	16847100
\$ SET OMIT = NOT SHAREDISK	16847200
IF J=USERDISKBOTTOM THEN	16847300
BEGIN	16847400
IF A>30 THEN DISKWAIT(-R=30,A=30,J+1);	16847500
MOVE(30,R1,R);	16847600
END ELSE	16847700
\$ POP OMIT	16847800
DISKWAIT(-R,A,J); J:=0;	16847900
FOR NT1:=K=2 STEP -1 UNTIL R4 DO INT3:=J+UT[NT1],[3:19];	16848000
STREAM(A:=I-1,B:=IF U[I],SPEED=1 THEN "F" ELSE "S",	16848100
C:=U[I],[38:10]-1,D:=J,E:=U[I],[1:20],	16848200
F:=A:=SPACE(10));	16848300
BEGIN	16848400
SI:=LOC A; DS:=4 LIT" EU "; DS:=2 DEC;	16848500
A:=DI; DI:=DI-2; DS:=FILL; DI:=A;	16848600
DS:=LIT"("; SI:=SI+7; DS:=CHR;	16848700
DS:=10 LIT"), NO, AV="; DS:=3 DEC;	16848800
A:=DI; DI:=DI-3; DS:=2 FILL; DI:=A;	16848900
DS:=11 LIT", TOTAL AV="; DS:=6 DEC;	16849000
A:=DI; DI:=DI-6; DS:=5 FILL; DI:=A;	16849100
DS:=14 LIT" SEGS, MAX AV="; DS:=6 DEC;	16849200
A:=DI; DI:=DI-6; DS:=5 FILL; DI:=A;	16849300
DS:=6 LIT" SEGS*";	16849400
END;	16849500
FORGETSPACE(R);	16849550
SPOUT(A);	16849600
END; % ELSE IF HN1=HN2 THEN GO TO ERROR;	16849700
\$ SET OMIT = NOT SHAREDISK	16849800
FORGETSPACE(R1);	16849900
\$ POP OMIT	16850000
HN1:=KTRX,[9:6]; % SET "MIX" BACK TO ORIGINAL VALUE	16850100
GO TO FORGET;	16850200
RA:	16850300
IF MEMROW[MIX],[CF] LSS FENCE THEN A:=RUNNING ELSE	16850400
IF (A:=STATUS[MIX]) NEQ RUNNING THEN GO TO RAEND;	16850500
IF NOT HALTED THEN	16850600
IF MIX=P2MIX THEN	16850700
BEGIN	16850800
HALT; HALTED := TRUE; GO TO RA;	16850900
END;	16851000
SEGDICT := PRT[MIX,4];	16851100
IF P( M[LOCN:=PRT[MIX,8],[CF]], TOP, XCH, DEL ) THEN SEGI:=ADR:=0	16851200
ELSE	16851300
DO BEGIN	16851400
IF P(M[LOCN], TOP, XCH, 0, INX, .ADR, STD) THEN % OVERLAID RCW	16851500
BEGIN	16851600
IF NOT M[LOCN],[33:1] THEN % NOT TYPE 13 INTRINSIC	16851700
BEGIN	16851800
SEGI:=ADR; % SEGNO IN RCW	16851900
R:=0; % ADJUST FOR SUBTRACTION BELOW	16852000

```

        ADR:=M[M[LOCN],MOM],[CF];  % REL,ADR,IN MSCW      16852100
    END      16852200
ELSE SEG := (-1);      16852300
END      16852400
ELSE      16852500
    BEGIN % PRESENT RCW, CHECK THE LINKS      16852600
    R:=IF ADR GTR FENCE THEN MEM[MIX,MLINK1],[CF] ELSE 0;      16852700
    WHILE (SEG:=M[R],[CF]) LSS ADR DO      16852800
    IF SEG GTR R THEN R:=SEG ELSE PUNT([PUNTER[4]]);      16852900
    SEG:=IF M[R],[3:6]=1 THEN M[R+1],[CF] ELSE 0;      16853000
    IF P(PRTROW[MIX],0,INX,DUP) GTR R AND P(XCH) LSS M[R],[CF] THEN      16853100
    R4 := "PRT ";      16853200
    R:=R+2;      16853300
    END;      16853400
    IF PRT[MIX,8],[CF] NEQ LOCN OR M[LOCN]=1,MSFF THEN % MARKED      16853500
    DO LOCN:=M[LOCN],MOM UNTIL NOT M[LOCN],MSFF; % GET LAST MSCW      16853600
    LOCN:=M[LOCN],MOM; % POINT LOCN TO NEXT RCW,JUST IN CASE,      16853700
    END      16853800
UNTIL      16853900
(IF SEG NEQ 0 THEN IF SEG = (-1) THEN 0      16854000
ELSE (SEGDICT[0] LSS SEG OR NOT SEGDICT[SEG],PBIT)      16854100
ELSE P(M[R=2],[3:6], DUP) NEQ 7 AND P(XCH) NEQ 13)      16854200
OR LOCN=0;      16854300
ADR := ADR-R;      16854400
RAEND;      16854500
    STREAM(MIX, NAM:=[JAR[MIX,0]], T:=0, SEG, ADR,      16854600
    SYL:=M[PRT[MIX,8]],[10:2], TOG1:=(R4 NEQ 0), R4,      16854700
    TOG2:=((SEG LEQ 0) OR (A NEQ RUNNING)), D:=BUFF-1);      16854800
    BEGIN      16854900
    DS:=LIT" ";      16855000
    SI:=NAM; 2(SI:=SI+1; DS:=7CHR; DS:=LIT"/"); DI:=DI-1;      16855100
    DS:=2LIT" "; SI:=LOC MIX; DS:=2DEC;      16855200
    TOG1(SI:=LOC R4; SI:=SI+1; DS:=LIT" "; DS:=7CHR; JUMP OUT TO XXIT);      16855300
    TOG2(DS:=14LIT" NOT AVAILABLE"; JUMP OUT TO XXIT);      16855400
    DS:=5LIT" SEG="; SI:=LOC SEG; DS:=4DEC;      16855500
    T:=DI; DI:=DI-4; DS:=3FILL; DI:=T;      16855600
    DS:=5LIT" ADR="; DS:=4DEC;      16855700
    T:=DI; DI:=DI-4; DS:=3FILL; DI:=T;      16855800
    DS:=LIT"!"; SI:=SI+7; DS:=CHR;      16855900
XXIT: DS:=LIT"~";      16856000
    END STREAM STATEMENT;      16856100
    IF HALTED THEN NOPROCESSTOG := NOPROCESSTOG -1;      16856200
    GO TO ERROR;      16856300
FORGET;      16944000
    STREAM(T:=BUFF-1); DS:=LIT"~";      16945000
    $ SET OMIT = NOT(PACKETS)      16945099
    UNITNO:=0;      16945100
    $ POP OMIT      16945101
ERROR:      16946000
    SPOUTER(BUFF-1,UNITNO,1);      16947000
EXIT:      16948000
    IF (MIX#0) AND (MIX<63) THEN TABCNT[MIX]:=TABCNT[MIX]-1;      16948500
    KILL([KTRX] INX NOT 1);      16949000
    END PROCEDURE KEYIN2;      16950000
REAL PROCEDURE KEYIN(B); VALUE B; REAL B;      16951000
% THIS PROCEDURE FUNCTIONS AS A DRIVER FOR AUXILIARY PROCEDURES      16951100
% "KEYINO","KEYIN1" AND "KEYIN2", PROCEDURES "KEYINO" AND "KEYIN1"      16951200
% ARE CALLED DIRECTLY, AND PROCEDURE "KEYIN2" IS FORKED AS AN      16951300
% INDEPENDENT RUNNER,      16951400
    BEGIN      16951500

```

REAL BUFF, KTR, TYPE, MIX, A, I, J, K,	16951600
MIXCODE=A, KTRX=A, PROCED=K;	16951700
\$ SET OMIT = NOT(PACKETS)	16951799
DEFINE UNITNO = PSEUDOMIX[MIX]#;	16951800
\$ POP OMIT	16951801
LABEL SWITCHIT, START, ERROR, EXIT, FORGET, TBLERR;	16951900
	16952000
IF B=0 THEN % WAIT TO GET EXCLUSIVE CONTROL OF KEYIN STACK	16952100
BEGIN	16952200
IF KEYBOARDCOUNTER NEQ 0 THEN COMPLEXSLEEP(KEYBOARDCOUNTER=0);	16952300
KEYBOARDCOUNTER:=KEYBOARDCOUNTER&1[17:47:1];	16952400
IF SPOWORD LSS 0 THEN % GET INPUT FROM BACK UP SPO	16952500
BEGIN	16952600
STREAM(MIX:=MIX:[KTR],[CF]); % SEND A QUESTION MARK	16952700
BEGIN	16952800
DS:=LIT"@"; DS:=LIT MARK;	16952900
END;	16953000
TWXOUT(MIX,2,-0,ABS(SPOWORD));	16953100
IF KEYBOARDCOUNTER,[FF]=0 THEN % WAIT FOR INPUT	16953200
SLEEP([KEYBOARDCOUNTER],@7777700000);	16953300
B:=0&KEYBOARDCOUNTER[FTF];	16953400
KEYBOARDCOUNTER:=KEYBOARDCOUNTER&M[B,[FF]=2][CTF];	16953500
END	16953600
ELSE KEYBOARDCOUNTER:=KEYBOARDCOUNTER INX 1;	16953700
END;	16953800
START;	16953900
IF ABS(B) GTR 1 THEN BUFF:=B.[18:15] ELSE	16954000
BEGIN	16954100
BUFF:=SPACE(60)+1;	16954200
P(WAITIO(BUFF&1[24:47:1],0,25),DEL);	16954300
END;	16954400
KTR:=BUFF;	16954500
IF (PROCED:=(TYPE:=KEYINSCAN(KTR,MIX)),[1:5])=7 THEN GO TO TBLERR;	16954600
KTR := KTR&BUFF[15:33:15];	16954650
MIXCODE := TYPE,[10:2]x(MIX#63);	16954700
TYPE := TYPE,[CF];	16954750
IF TYPE=0 OR MIX,[1:2]#0 THEN % EMPTY OR ERROR	16954800
BEGIN	16954900
IF MIX,[1:1] THEN % EMPTY BUFFER	16955000
BEGIN	16955100
KEYIN:=TRUE; GO TO FORGET;	16955200
END	16955700
ELSE GO TO ERROR; % TYPE=0 OR MIX,[2:1]	16955800
END;	16955900
IF MIXCODE=1 OR MIXCODE=2 THEN % MIX INDEX REQUIRED	16956000
BEGIN	16956100
IF MIX GTR MIXMAX THEN GO TO ERROR;	16956200
IF JARROW[MIX]=0 OR PRTRROW[MIX]=0 THEN GO TO ERROR;	16956300
IF MIXCODE=1 THEN % JOB SHOULD BE WAITING FOR THIS INPUT	16956400
BEGIN	16956500
J:=REPLY[MIX];	16956600
WHILE J LSS 0 DO	16956700
BEGIN	16956800
IF J,[42:6]=TYPE THEN GO TO SWITCHIT;	16956900
J:=-J,[6:36]; % SHIFT RIGHT	16957000
END;	16957100
IF TYPE=VWY THEN % "WY", NOT WAITING FOR IT	16957200
BEGIN	16957300
M[BUFF-1]:=FLAG(="WY NOT"&MIX[6:42:6]);	16957400
M[BUFF] :=0&(@1437)[1:37:11];	16957500

```

        END;
        GO TO ERROR;
        END; % IF MIXCODE = 1 OR 2
SWITCHIT;
        TABCNT[MIX]:=TABCNT[MIX]+1;
        $ SET OMIT = NOT(PACKETS)
        IF PSEUDOMIX[MIX]#0 THEN
        BEGIN
        STREAM(I:=0; BUFF);
        BEGIN SI:=BUFF;
        L1: IF SC="*" THEN GO L2; SI:=SI+1; GO L1;
        L2: I:=SI;
        END;
        I:= P.[CF]-BUFF+1; % NWDS
        MOVE(1,BUFF,J:=SPACE(I));
        SPOUTER(J,UNITNO,64);
        END;
        $ POP OMIT
        END; % IF MIX INDEX REQUIRED
        KTRX:=KTR & MIX[9:42:6] & TYPE[2:41:7];
        IF PROCED=2 THEN FORK(NT1:=P(,KEYIN2),KTRX,0,128,0) ELSE
        IF PROCED=1 THEN KEYIN1(B,KTRX) ELSE KEYINO(B,KTRX);
        GO TO EXIT;
        TBLERR;
        STREAM(KTR,B:=BUFF-1);
        BEGIN
        SI:=KTR; SI:=SI-2; DS:=LIT"*"; DS:=2CHR;
        DS:=21LIT" NOT COMPILED IN MCP*";
        END;
        ERROR;
        SPOUT(BUFF-1);
        KEYIN := TRUE;
        GO TO EXIT;
        FORGET;
        STREAM(T:=BUFF-1); DS:=LIT "*"; SPOUT(BUFF-1);
        EXIT;
        IF ABS(B) LEQ 1 THEN KEYBOARDCOUNTER:=P((NOT 0) INX KEYBOARDCOUNTER);
        IF KEYBOARDCOUNTER&0[17:47:1] GTR 0 THEN
        BEGIN
        IF KEYBOARDCOUNTER.[CF] NEQ 0 THEN B:=B&0[CF] ELSE
        BEGIN
        B:=B&KEYBOARDCOUNTER[FTF];
        KEYBOARDCOUNTER:=KEYBOARDCOUNTER&M[B,[FF]=2][CF];
        END;
        GO TO START;
        END;
        KEYBOARDCOUNTER.[17:1]:=0;
        IF B THEN KILL([B] INX NOT 1);
        END PROCEDURE KEYIN;
        PROCEDURE LBMESS(FN,SN,I1,I2,E,UNITNO,X);
        VALUE FN,SN,I1,I2,E,UNITNO,X;
        REAL FN,SN,I1,I2,E,UNITNO,X;
        *****
        %
        %          PARAMETERS
        %          I1      I2      E          FORM OF MESSAGE
        %          -----
        %          LSS 0      0          , FN/SN I1
        %          LSS 0      GTR 0      0          , FN/SN NOT I1(I2)
        %          LSS 0      GTR 0      NEQ 0       , FN/SN NOT I1(I2), E
        %          GTR 0      0          0          FN/SN I1

```

```

16957600
16957700
16957750
16957800
16957810
16957819
16957820
16957830
16957840
16957850
16957860
16957870
16957880
16957890
16957900
16957910
16957920
16957921
16958000
16958100
16958200
16958300
16958400
16968500
16968600
16968700
16968800
16968900
16969000
16969100
16969200
16969300
16969400
16969500
16969600
16969700
16969800
16969900
16970000
16970100
16970200
16970300
16970400
16970500
16970600
16970700
16970800
16970900
16971000
17000000
17000200
17000400
17000405
17000410
17000420
17000430
17000440
17000450
17000460
17000470

```

*use of  
KEYBOARDCOUNTER  
and UNITNO*

```

%      GTR 0      0      NEQ 0      FN/SN 11, E      17000480
%      GTR 0      GTR 0      FN/SN 11 12      17000490
%      52 OR 54      FN/SN 11 12/E      17000500
%NOTE! IF I1 IS NEITHER 52 NOR 54 THEN I1 AND I2 ARE INDICES INTO TABL 17000510
%      ELSE I2 AND E ARE MFID AND FID,      17000520
%*****      17000530
BEGIN      17000600
REAL T,A; ARRAY TABL[*];      17000800
IF LOGLINE,[33:7]=0 OR CANDYMESS THEN      17002200
BEGIN      17002400
TABL:=[M[SPACE(A:=MESSAGETABLE[4],[8:10])]] &      17002600
      MESSAGETABLE[4][8:8:10];      17002800
DISKWAIT(=(TABL,[CF]),A,MESSAGETABLE[4],[22:26]);      17003000
STREAM(A:=[FN],I:=I1 LSS 0,TBL1:=[TABL[ABS(I1)],E,      17003200
      L1:=I1 LSS 0 AND I2 NEQ 0,J:=I1=52 OR I1=54,      17003300
      B:=IF P(DUP) THEN [I2] ELSE [TABL[I2]],T:=T:=SPACE(10));      17003450
BEGIN I(DS:=LIT","); DS:=LIT" "; SI:=A;      17003500
IF SC="+" THEN BEGIN DS:=LIT"="; SI:=SI+8; END      17003550
      ELSE BEGIN SI:=SI+1; DS:=7CHR; END; DS:=LIT"/";      17003600
IF SC="+" THEN BEGIN DS:=LIT"="; SI:=SI+8; END      17003700
      ELSE BEGIN SI:=SI+1; DS:=7CHR; END;      17003750
DS:=LIT" "; L(DS:=4LIT"NOT "); SI:=TBL1;      17003800
63(SI:=SI+1; 7(IF SC="+" THEN JUMP OUT 2 TO L1 ELSE DS:=CHR));      17003850
L1:      SI:=B;      17003900
      J(IF SC="+" THEN BEGIN DS:=LIT"="; SI:=SI+8; END      17003950
      ELSE BEGIN SI:=SI+1; DS:=7CHR; END; DS:=LIT"/";      17004000
      IF SC="+" THEN BEGIN DS:=LIT"="; SI:=SI+8; END      17004050
      ELSE BEGIN SI:=SI+1; DS:=7CHR; END; JUMP OUT TO L3);      17004100
L(DS:=LIT"(");      17004150
63(SI:=SI+1; 7(IF SC="+" THEN JUMP OUT 2 TO L2 ELSE DS:=CHR));      17004200
L2:      L(DS:=LIT")"); SI:=LOC E; SI:=SI+5;      17004250
      IF SC NEQ "0" THEN BEGIN DS:=2LIT", "; DS:=3CHR; END;      17004300
L3:      DS:=LIT"+";      17004600
      END; %STREAM      17005200
      SPOUTER(T,UNITNO,X);      17005400
      FORGETSPACE(TABL,[CF]);      17005600
      END;      17006800
END; %LIBMSG      17007000
PROCEDURE STOPM;      17900000
BEGIN INTEGER PROTY; LABEL AROUND; REAL B;      %S*17901000
      PROTY=PRYOR[P1MIX];      %S*17902000
      PRIORITY=PRYOR[P1MIX]+1023;      17903000
      IF NOTERMSET(P1MIX) THEN PRTRROW[P1MIX],[PSF]+0;      17903500
AROUND: STREAM(J:=JARROW[P1MIX],P1MIX,BI:=B:=SPACE(10));      17904000
      BEGIN DS=13LIT"#OPRTR ST=ED "; SI=J; SI=SI+1; DS=7 CHR;      %ST17905000
      SI=SI+1; DS=LIT"/"; DS=7 CHR; DS=LIT"=";      %ST17906000
      SI=LOC P1MIX; DS=2DEC; DS=LIT"+"; DI=DI-3; DS=FILL;      17907000
      END; SPOUT(B);      %ST17908000
      IF OUTWAIT(FALSE) THEN GO AROUND;      17909000
      PRIORITY=PRYOR[P1MIX]+PROTY;      17910000
END;      %S*17916000
PROCEDURE FILEHOLD(A,B,TOG,LOC,HOLD);      18000000
      VALUE LOC,HOLD;      18001000
      REAL A,B,TOG,LOC,HOLD;      18002000
BEGIN      18003000
      REAL SZ,Y,T;      18004000
      $ SET OMIT = NOT SHAREDISK      18004490
      REAL HOLDER,NEXTSLOT,BYPASS;      % HOLDER MUST BE AT T+1      18004500
      $ POP OMIT      18004510
      ARRAY HOLDLIST[*];      18005000

```



LABEL SLEPE;	18006000
DEFINE DSED=(TERMSET(P1MIX))#;	18007000
IF HOLD THEN	18008000
BEGIN	18009000
IF TOG THEN TOG←TOG+1 ELSE	18010000
BEGIN % MAKE AN ENTRY IN THE HOLDLIST	18011000
\$ SET OMIT = NOT SHAREDISK	18011490
DISKWAIT(=[HOLDER],[CF]),-3,DIRECTORYSEG); % CLOBBERS T	18011500
\$ POP OMIT	18011510
IF (SZ=[Y]=[HOLDER],[FF])+1) GTR HOLDMAX THEN	18012000
BYBY("HOLD LIST OVERFLOW+",19);	18013000
HOLDLIST:=[M[SPACE(SZ)]]&SZ[8:38:10];	18014000
IF Y≠0 THEN	18014100
DISKWAIT(=[HOLDLIST INX 0],Y,HOLDER,[CF]);	18015000
HOLDER,[FF]:=SZ;	18016000
HOLDLIST[Y]:=LOC,[FF]&[TOG][CTF]&SYSNO[2:46:2]	18017000
&P1MIX[10:40:8];	18017100
DISKWAIT(HOLDLIST INX 0,SZ,HOLDER,[CF]);	18018000
\$ SET OMIT = NOT SHAREDISK	18018490
DISKWAIT([HOLDER],[CF]),-3,DIRECTORYSEG); % CLOBBERS T	18018500
\$ POP OMIT	18018510
FORGETSPACE(HOLDLIST);	18019000
END;	18019500
IF M[LOC+4],[3:1] THEN	18020000
\$ SET OMIT = NOT SHAREDISK	18020490
UNLOCK(LOC,[FF])	18020500
\$ POP OMIT	18020510
ELSE	18021000
BEGIN M[LOC+4],[3:1]:=1;	18021500
DISKWAIT(LOC,[CF]),-30,LOC,[FF]);	18022000
END;	18022500
\$ SET OMIT = SHAREDISK	18022990
UNLOCKDIRECTORY;	18023000
\$ POP OMIT	18023010
IF P1MIX≠0 THEN	18024000
BEGIN T:=VWY&(VIF×A,[3:1])[36:42:6];	18025000
IF TOG=0 THEN	18026000
SLEPE;  FILEMESS("  ",A,B," IN USE",0,0,0);	18027000
REPLY[P1MIX]:=-T;	18028000
IF P(0,RDS)≥FENCE THEN SWAP(WAITSWAP,1) ELSE	%029-18029000
COMPLEXSLEEP(REPLY[P1MIX]≥0 OR DSED OR TOG);	%029-18029500
IF NOT WHYSLEEP(T) THEN GO TO SLEPE;	18030000
END ELSE	18031000
BEGIN LBMESS(ABS(A),B,45,0,"MCP",0,1);	18031500
SLEEP([TOG],1);	18032000
END;	18032500
\$ SET OMIT = SHAREDISK	18032990
LOCKDIRECTORY;	18033000
\$ POP OMIT	18033010
TOG:=TRUE;	18033500
IF P((P1MIX NEQ 0 AND DSED),DUP)	18034000
THEN FILEHOLD(A,B,TOG,LOC,2);	18035000
P(RTN); % 1 ON TOP OF STACK IF DSED	18037000
END;	18045000
\$ SET OMIT = NOT SHAREDISK	18045490
DISKWAIT(=[HOLDER],[CF]),-3,DIRECTORYSEG); % CLOBBERS T	18045500
\$ POP OMIT	18045510
IF (SZ=[HOLDER],[FF])=0 THEN	18046000
\$ SET OMIT = NOT SHAREDISK	18046490
UNLOCK(DIRECTORYSEG)	18046500

```

$ POP OMIT
ELSE
BEGIN IF HOLD=2 THEN DISKWAIT(=LOC,[CF],-30,LOC,[FF]);
HOLDLIST:=M[SPACE(SZ)]&SZ[8:38:10];
DISKWAIT(=HOLDLIST INX 0),SZ,HOLDER,[CF]);
IF TOG THEN FOR T:=0 STEP 1 UNTIL SZ-1 DO
$ SET OMIT = NOT(SHAREDISK)
IF HOLDLIST[T],[2:2]=SYSNO THEN
$ POP OMIT
IF HOLDLIST[T],[FF]=[TOG],[CF] THEN
IF HOLDLIST[T],[10:8]=P1MIX THEN
IF (SZ=SZ-1) # T THEN
BEGIN
MOVE(SZ-T,[HOLDLIST[T+1]],[HOLDLIST[T]]);
T:=SZ;
END;
HOLDER,[FF]:=Y:=SZ;
IF SZ#0 THEN
BEGIN
FOR Y=0 STEP 1 UNTIL SZ-1 DO
IF HOLDLIST[Y],[CF]=LOC,[FF] THEN
BEGIN
$ SET OMIT = NOT(SHAREDISK)
IF HOLDLIST[Y],[2:2]#SYSNO THEN
HOLDLIST[Y]:=P(DUP,LOC,SSN) ELSE
$ POP OMIT
IF (T:=HOLDLIST[Y]),[FF] GEQ FENCE THEN
BRINGBACK(T,[10:8]) ELSE M[T,[FF]]:=1;
Y:=SZ;
END;
DISKWAIT(HOLDLIST INX 0,SZ,HOLDER,[CF]);
END;
$ SET OMIT = NOT SHAREDISK
DISKWAIT([HOLDER],[CF],-3,DIRECTORYSEG); % CLOBBERS T
$ POP OMIT
IF SZ=Y THEN
BEGIN
M[LOC+4],[3:1]:=0;
IF HOLD=2 THEN DISKWAIT(LOC,[CF],-30,LOC,[FF]);
$ SET OMIT = NOT SHAREDISK
END ELSE
BEGIN
IF HOLD=2 THEN UNLOCK(LOC,[FF]);
$ POP OMIT
END;
FORGETSPACE(HOLDLIST);
END;
END; % OF FILEHOLDER
%COMMENT THE DISK FILE HEADER CONTAINS THE FOLLOWING INFORMATION:
%
%HI[0],[0:15] RECORD LENGTH
% ,[15:15] BLOCK LENGTH
% ,[30:12] RECORD/BLOCK
% ,[42:6] SEGMENTS/BLOCK
%HI[1],[6:18] CREATION DATE FOR LOGGING (WHEN ON DISK)
% ,[25:23] CREATION TIME FOR LOGGING (WHEN ON DISK)
% ,[1:47] NUMBER OF LOGICAL RECORDS PER ROW (WHEN IN CORE)
%HI[2],[0:48] =0 FREE FILE
% ,[1:1] =0 SOLE USER, PUBLIC OR PRIVATE FILE
% ,[1:1] =1 SECURITY FILE

```

```

18046510
18047000
18047500
18048000
18049000
18050000
18051000
18052000
18052001
18053000
18053500
18054000
18055000
18056000
18057000
18058000
18059000
18060000
18061000
18062000
18063000
18064000
18065000
18066000
18067000
18067001
18068000
18068100
18069000
18070000
18071000
18072000
18072490
18072500
18072510
18073000
18074000
18075000
18075500
18075990
18076000
18076500
18077000
18077010
18077500
18078000
18079000
18080000
18081000
18082000
18083000
18084000
18085000
18086000
18087000
18088000
18089000
18090000
18091000
18092000

```

%	.[6:42]	PRIMARY USER'S CODE	18093000
%H[3]	.[1:1]	=1 NEW FILE HEADER FORMAT	18094000
%	.[2:10]	SAVE FACTOR (BINARY)	18095000
%	.[12:18]	DATE OF LAST ACCESS (BINARY)	18096000
%	.[30:18]	CREATION DATE (BINARY)	18097000
%H[4]	.[1:1]	=1 FILE IS BEING LOADED OR NAME IS BEING CHANGED	18098000
%	.[2:1]	=1 FILE IS OPENED BY AN EXCLUSIVE USER	18099000
%	.[3:1]	=1 A PROGRAM IS WAITING TO USE THE FILE	18100000
%	.[4:2]	SYSTEM NUMBER OF EXCLUSIVE USER	18101000
%	.[6:1]	USED BY AUTOPRINT TO MARK A PBD FILE	18102000
%	.[7:1]	USED TO MARK PSEUDO DECKS THAT WERE CREATED ON	18103000
%		A TIME-SHARING SYSTEM BY A ZIP WITH FILE-ID	18104000
%	.[8:1]	USED TO MARK SPECIAL COMPILERS	18104100
%	.[9:2]	=2 FILE IS DATA	18105000
%		=3 FILE IS PROGRAM	18106000
%		=0 DON'T KNOW IF DATA OR PROGRAM	18107000
%	.[11:1]	FILE ACCESSED BIT	18108000
%	.[12:4]	SYSTEM FILE TOGGLES	18109000
%	.[16:5]	OPEN COUNT 2 FOR SYSTEM 0 (A)	18110000
%	.[21:5]	OPEN COUNT 2 FOR SYSTEM 1 (B)	18111000
%	.[26:5]	OPEN COUNT 2 FOR SYSTEM 2 (C)	18112000
%	.[31:5]	OPEN COUNT 2 FOR SYSTEM 3 (D)	18113000
%	.[36:6]	=0 TYPE IS UNKNOWN	18114000
%		=1 BASIC	18115000
%		=2 ALGOL	18116000
%		=3 COBOL	18117000
%		=4 FORTRAN	18118000
%		=5 TSPOL	18119000
%		=6 XALGOL	18120000
%		=7 SEQ	18121000
%		=8 DATA	18122000
%		=9 LOCK	18123000
%	.[42:1]	USED TO MARK FILES WHICH CANT BE MOVED	18123100
%	.[43:2]	SENSITIVE DATA = ZEROING BITS	18124000
%	.[45:1]	COLD START FILE	18124100
%	.[46:2]	NOT USED	18124200
%H[5]	.[0:48]	=0 SOLE USER FILE	18125000
%	.[1:1]	=1 PRIVATE FILE	18126000
%		=12 IF H[6]=12 THEN INFO FILE ELSE PUBLIC FILE	18127000
%H[7]		NUMBER OF LOGICAL RECORDS (EOF POINTER)	18128000
%H[8]		NUMBER OF SEGMENTS PER ROW	18129000
%H[9]	.[1:1]	TOGGLE 1 FOR SYSTEM 0 (A)	18130000
%	.[2:1]	TOGGLE 1 FOR SYSTEM 1 (B)	18131000
%	.[3:1]	TOGGLE 1 FOR SYSTEM 2 (C)	18132000
%	.[4:1]	TOGGLE 1 FOR SYSTEM 3 (D)	18133000
%	.[5:1]	TOGGLE 2 FOR SYSTEM 0 (A)	18134000
%	.[6:1]	TOGGLE 2 FOR SYSTEM 1 (B)	18135000
%	.[7:1]	TOGGLE 2 FOR SYSTEM 2 (C)	18136000
%	.[8:1]	TOGGLE 2 FOR SYSTEM 3 (D)	18137000
%	.[9:5]	OPEN COUNT 1 FOR SYSTEM 0 (A)	18138000
%	.[14:5]	OPEN COUNT 1 FOR SYSTEM 1 (B)	18139000
%	.[19:5]	OPEN COUNT 1 FOR SYSTEM 2 (C)	18140000
%	.[24:5]	OPEN COUNT 1 FOR SYSTEM 3 (D)	18141000
%	.[29:14]	NOT USED	18142000
%	.[43:5]	MAXIMUM NUMBER OF ROWS	18143000
%H[10]=H[29]		DISK ADDRESSES OF ROWS (0 IF NOT ASSIGNED)	18144000
%			18145000
%			18146000
%			18147000
%			18148000

THE OPEN COUNTS AND TOGGLES ARE USED IN THE FOLLOWING MANNER:

%	TOGGLE 1	TOGGLE 2	OPEN COUNT 1	OPEN COUNT 2	18149000
%	0	0	INPUT ONLY	INPUT	18150000
%	0	1 (OUTPUT)	NOT USED	INPUT	18151000
%	1	0	SHARED	INPUT	18152000
%	1	1	PROTECT	INPUT	18152100
%					18153000
%	END COMMENT;				18154000
	REAL PROCEDURE DIRECTORYSEARCH(A,B,OPTN);%				18155000
	VALUE A,B,OPTN; REAL A,B,OPTN;%				18156000
%	OPTN= 0	OPENS FOR SHARED USE			18157000
%	OPTN= 1	OPENS FOR INPUT			18158000
%	OPTN= 2	OPENS FOR OUTPUT			18159000
%	OPTN= 3	OPENS FOR WRITELOCK			18160000
%	OPTN= 4	OPENS FOR EXCLUSIVE USE			18161000
%	OPTN= 5	RETURNS FILE HEADER (UNCHANGED)			18162000
%	OPTN= 6	REMOVES FILE FROM DISK UNCONDITIONALLY			18163000
%	OPTN= 7	REMOVES FILE FROM DISK AS SOON AS IT IS NOT IN USE			18164000
%	OPTN= 8	REMOVES FILE HEADER ONLY			18165000
%	OPTN= 9	HEADERUNLOCK--WRITES HEADER POINTED TO BY (F=4),[CF]			18166000
%		BACK OUT ON (F=4),[FF], TURNS OFF INTERLOCK & DOES			18167000
%		FORGETSPACE(F=4),			18168000
%	OPTN=10	CLOSE SHARED			18169000
%	OPTN=11	CLOSE INPUT			18170000
%	OPTN=12	CLOSE OUTPUT			18171000
%	OPTN=13	CLOSE WRITELOCK			18172000
%	OPTN=14	CLOSE EXCLUSIVE			18173000
%	OPTN=15	LOGS THE FILE AND RESETS ITS CREATION DATE AND TIME			18174000
%	OPTN=16	MAKES THE FILE NOT A SYSTEM FILE			18175000
%	OPTN=17	MAKES THE FILE A SYSTEM FILE			18176000
%	OPTN=18	WILL INTERLOCK SYSTEM FILES			18177000
%	OPTN=19	RETURNS FILE HEADER (UNCHANGED AND LOCKED,..IT IS UP TO			18178000
%		THE CALLING ROUTINE TO CLEAN UP)			18178100
%	OPTN=20	UNUSED			18179000
%	OPTN=21	OPENS PROTECT			18179100
%	OPTN=22	CLOSE PROTECT			18179200
%	OPTN>512	FILECLOSE--ADDRESS OF HEADER IN OPTN,[CF]			18180000
%		CLOSE OPTION=10 IS IN OPTN,[FF]			18181000
%	OPTN< 0	RETURNS AN AREA OF USER DISK AND UPDATES CORE COPY			18182000
%		OF FILE HEADER--ADDRESS OF HEADER IS IN OPTN,[CF]--			18183000
%		NUMBER OF THE ROW TO BE FILLED IS IN OPTN,[FF]			18184000
%		IS IN OPTN,[CF]			18185000
%	A,[1:1]	DIRECTORYSEARCH WILL FORGET THE MEMORY SPACE			18186000
%		OCCUPIED BY THE FILE HEADER			18187000
%	A,[2:1]	IS DIALED INTO FH[4],[1:1] WHEN OPTN=4			18188000
%	A,[3:1]	IF A CONFLICT OCCURS, AN "IF" WILL BE ENABLED, IF THE			18189000
%		OPERATOR ENTERS AN "IF", DIRECTORYSEARCH WILL RETURN A			18190000
%		VALUE OF =0, CURRENTLY, THIS IS USED ONLY BY LIBMAIN,			18191000
%	B,[1:1]	DIRECTORYSEARCH WILL RETURN A VALUE OF =0 IF THE			18192000
%		FILE IS IN USE			18193000
%	B,[2:1]	WILL NOT UPDATE DATE OF LAST ACCESS			18194000
%	B,[3:1]	WILL SET FILE ACCESSED BIT FOR CLOSE			18195000
	BEGIN				18196000
	REAL OLDDONE=#4;				18197000
	REAL TEMP,I,T,TOG,J,K,N,F,X;				18198000
	INTEGER S,I1,I2,I3;				18199000
	REAL UNITNO;				18199100
	ARRAY FH[*],NB[*];				18200000
\$	SET OMIT = NOT SHAREDISK				18200490
	INTEGER S1;				18200500
	REAL KLUDGE=S1+1, HOLDER=KLUDGE+1, NEXTSLOT=HOLDER+1,				18200600



```

END;
%*****
SUBROUTINE REMOVER; % CANT BE CALLED FROM OTHER SUBROUTINES,
BEGIN NB[X]←@14;
  DISKWAIT(N,=60,S);
$ SET OMIT = NOT SHAREDISK
  IF S1≠S THEN UNLOCK(S1);
  P(0,0,0,KLUDGE); % MAKE WAY FOR THE UNCLEAN
  DISKWAIT(=[HOLDER],[CF]),=3,DIRECTORYSEG);
$ POP OMIT % BUT SAVE THE RETURN LITERAL,
  DISKWAIT(=N,=30,J);
  NB[I]←@14; NB[I+1]←NEXTSLOT; NEXTSLOT←K;
  DISKWAIT(N,=30,J);
$ SET OMIT = NOT SHAREDISK
  DISKWAIT([HOLDER],[CF]),=3,DIRECTORYSEG);
  UNLOCK(K);
$ POP OMIT
END;
%*****
SUBROUTINE HOLD;
BEGIN
$ SET OMIT = NOT SHAREDISK
  IF OPTN=7 THEN UNLOCK(S1);
  IF B.[1:1] THEN
    BEGIN TEMP:=1; % IN USE
      GO UNLOCKHDR;
    END;
$ SET OMIT = SHAREDISK
  IF B.[1:1] THEN GO DONTWAIT;
$ POP OMIT OMIT
  FILEHOLD(A,B,TOG,TEMP,1);
  IF P THEN % 0 OR 1 IS LEFT ON TOP OF STACK IN FILEHOLD
    BEGIN TEMP:=3; A:=1; GO EXIT; END % DS=ED IN FILEHOLD
  ELSE
    IF P1MIX≠0 THEN
      IF REPLY[P1MIX]=VIF THEN
        BEGIN
          FILEHOLD(A,B,TOG,TEMP,2);
        DONTWAIT;
          TEMP:=1; % IN USE
          GO TO EXIT;
        END;
        IF SEARCH=0 THEN
          BEGIN FILEHOLD(A,B,TOG,TEMP,0);
        NOFILE; TEMP:=0; A:=1; GO EXIT;
        END;
        HEADER;
      END; % OF HOLD
%*****
$ SET OMIT = NOT(PACKETS)
  IF OPTN,[CF] LSS 512 THEN
    BEGIN UNITNO:=OPTN,[24:6]; OPTNI=OPTN INX 0; END;
$ POP OMIT
$ SET OMIT = SHAREDISK
  LOCKDIRECTORY;
$ POP OMIT
  IF OPTN=9 THEN
    BEGIN
      DISKWAIT(=(N:=SPACE(30)),=30,(K:=OLDDONE,[FF]));
      FH:=M[F:=OLDDONE,[CF]]&30[8:38:10];

```

```

18242000
18243000
18244000
18245000
18245500
18245990
18246000
18246500
18247000
18247010
18247500
18248000
18248500
18248990
18249000
18249500
18249510
18250000
18251000
18252000
18253000
18253490
18253500
18253600
18253700
18253800
18253900
18253950
18254000
18254010
18255000
18255100
18255200
18255300
18256000
18258000
18259000
18260000
18260500
18261000
18262000
18263000
18264000
18265000
18266000
18269000
18270000
18271000
18272000
18272199
18272200
18272300
18272301
18272990
18273000
18273010
18274000
18275000
18276000
18276500

```

A:=NABS(A);	18277000
FH[4]:=(P(DUP))&M[N+4][3:3:1];	18277500
GO TO CLOSEXCLUSIVE;	18278000
END;	18279000
NBI=[M[N]=SPACE(60)]&60[8:38:10];	18280000
IF SEARCH=0 THEN	18281000
BEGIN	18282000
A:=0;	18283000
GO TO EXIT;	18284000
END;	18285000
\$ SET OMIT = NOT SHAREDISK	18285099
TEMP:=IF OPTN=21 THEN 32 ELSE 30;	18285100
FH=[M[F+SPACE(TEMP)]]&TEMP[8:38:10];	18285200
\$ POP OMIT	18285201
\$ SET OMIT = SHAREDISK	18285999
FH=[M[F]=SPACE(30)]&30[8:38:10];	18286000
\$ POP OMIT	18286001
HEADER;	18287000
IF OPTN<0 THEN GO GETAROW;	18288000
IF OPTN>512 THEN GO TO @[OPTN,[FF]+10];	18289000
\$ SET OMIT = SHAREDISK	18289999
IF OPTN LSS 5 OR OPTN=17 OR OPTN=7 THEN	18290000
\$ POP OMIT	18290001
\$ SET OMIT = NOT SHAREDISK	18290099
IF OPTN<5 OR OPTN=17 OR OPTN=7 OR OPTN=21 THEN	18290100
\$ POP OMIT	18290101
CHECK;	18291000
BEGIN	18292000
IF FH[4],[44:1] AND OPTN LSS 5 THEN	18292100
BEGIN % TRYING TO OPEN WHILE FILE IS BEING BLANKED	18292200
\$ SET OMIT = NOT SHAREDISK	18292300
UNLOCK(K);	18292400
\$ POP OMIT	18292500
GO NOFILE;	18292600
END;	18292700
\$ SET OMIT = SHAREDISK	18292999
IF NOT OPTN OR OPTN=7 THEN	18293000
\$ POP OMIT	18293001
\$ SET OMIT = NOT SHAREDISK	18293099
IF NOT OPTN OR OPTN=7 OR OPTN=21 THEN	18293100
\$ POP OMIT	18293101
IF FH[4],[12:4]≠0 THEN	18293200
BEGIN % IT IS A SYSTEM FILE	18294000
TEMP:=2; % SYSTEM FILE	18295000
\$ SET OMIT = NOT SHAREDISK	18295490
IF OPTN THEN UNLOCK(S1); % OPTN = 7	18295500
UNLOCKHDR;	18295600
\$ POP OMIT	18295610
GO TO EXIT;	18296000
END;	18297000
SEE;	18298000
IF (FH[4],[2:2] AND (NOT TOG OR 2))≠0 THEN	18299000
BEGIN	18300000
HOLD;	18301000
GO CHECK;	18302000
END;	18303000
END;	18305000
GO TO @[OPTN];	18306000
OPENSHARED;	18307000
IF FH[9],[5:4]≠0 THEN	18308000

IF FH[9],[1:4]≠0 OR FH[9],[9:20]=0 THEN	18309000
BEGIN	18310000
\$ SET OMIT = NOT(SHAREDISK)	18310999
P(SYSNO); ::P(,ONE,+,LOD); T+P;	18311000
FH[9],[ 9:20]+P(DUP),[ 9:20]+T;	18312000
P(SYSNO); ::P(,TOG1,+,LOD); T+P;	18313000
FH[9]+P(DUP,LOD) OR T;	18314000
\$ POP OMIT	18314001
\$ SET OMIT = SHAREDISK	18314099
FH[9],[9:5]=P(DUP),[9:5]+1;	18314100
FH[9],[1:1]=1;	18314200
\$ POP OMIT	18314201
GO TO LWRITE;	18315000
END;	18316000
HOLD;	18317000
GO TO OPENSHARED;	18318000
OPENINPUT;	18319000
\$ SET OMIT = NOT(SHAREDISK)	18319999
P(SYSNO); ::P(,ONE,+,LOD); T+P;	18320000
FH[4],[16:20]+P(DUP),[16:20]+T;	18321000
\$ POP OMIT	18321001
\$ SET OMIT = SHAREDISK	18321099
FH[4],[16:5]=P(DUP),[16:5]+1;	18321100
\$ POP OMIT	18321101
GO TO LWRITE;	18322000
OPENOUTPUT;	18323000
IF FH[9],[5:24]=0 THEN	18324000
BEGIN	18325000
\$ SET OMIT = NOT(SHAREDISK)	18325999
P(SYSNO); ::P(,TOG2,+,LOD); T+P;	18326000
FH[9]+P(DUP,LOD) OR T;	18327000
\$ POP OMIT	18327001
\$ SET OMIT = SHAREDISK	18327099
FH[9],[5:1]=1;	18327100
\$ POP OMIT	18327101
GO TO LWRITE;	18328000
END;	18329000
HOLD;	18330000
GO TO OPENOUTPUT;	18331000
OPENWRITELOCK;	18332000
IF FH[9],[1:8]=0 THEN	18333000
BEGIN	18334000
\$ SET OMIT = NOT(SHAREDISK)	18334999
P(SYSNO); ::P(,ONE,+,LOD); T+P;	18335000
FH[9],[9:20]+P(DUP),[9:20]+T;	18336000
\$ POP OMIT	18336001
\$ SET OMIT = SHAREDISK	18336099
FH[9],[9:5]=P(DUP),[9:5]+1;	18336100
\$ POP OMIT	18336101
GO TO LWRITE;	18337000
END;	18338000
HOLD;	18339000
GO TO OPENWRITELOCK;	18340000
OPENEXCLUSIVE;	18341000
IF FH[9],[5:24]=0 THEN	18342000
IF FH[4],[16:20]=0 THEN	18343000
BEGIN	18344000
COMPLETE: FH[4]=P(DUP,LOD)&SYSNO[4:46:2]&1[2:47:1]&A[1:2:1];	18345000
GO TO LWRITE;	18346000
END;	18347000



HOLD;	18348000
GO TO OPENEXCLUSIVE;	18349000
OPENPROTECT;	18349100
\$ SET OMIT = NOT SHAREDISK	18349149
IF FH[9],[1:4]≠0 AND FH[9],[5:4]≠0	18349150
OR FH[9],[9:20]=0 THEN	18349200
BEGIN	18349250
P(SYSNO); ;P(,TOG1,+,LOD); T+P;	18349300
P(SYSNO); ;P(,TOG2,+,LOD); T+P OR T;	18349350
FH[9]+P(DUP,LOD) OR T;	18349400
PROCNT; P(SYSNO); ;P(,ONE,+,LOD); T+P;	18349450
FH[9],[9:20]+P(DUP),[9:20]+T;	18349500
GO TO LWRITE;	18349550
END;	18349600
HOLD;	18349650
GO TO OPENPROTECT;	18349700
\$ POP OMIT	18349701
\$ SET OMIT = SHAREDISK	18349799
GO TO OPENEXCLUSIVE;	18349800
\$ POP OMIT	18349801
CLOSESHARED;	18350000
\$ SET OMIT = NOT(SHAREDISK)	18350999
P(SYSNO); ;P(,ONE,+,LOD); T+P;	18351000
IF ((I:=FH[9],[9:20]=T) AND T×31)=0 THEN	18352000
BEGIN	18353000
P(SYSNO); ;P(,TOG1,+,LOD,NOT); T+P;	18354000
FH[9]+P(DUP,LOD) AND T;	18355000
END;	18356000
FH[9],[9:20]=I;	18357000
\$ POP OMIT	18357001
\$ SET OMIT = SHAREDISK	18357099
IF (I:=FH[9],[9:5]=1)=0 THEN	18357100
FH[9],[1:1]=0;	18357200
FH[9],[9:5]=I;	18357300
\$ POP OMIT	18357301
GO TO CLOSE;	18358000
CLOSEINPUT;	18359000
\$ SET OMIT = NOT(SHAREDISK)	18359999
P(SYSNO); ;P(,ONE,+,LOD); T+P;	18360000
FH[4],[16:20]+P(DUP),[16:20]=T;	18361000
\$ POP OMIT	18361001
\$ SET OMIT = SHAREDISK	18361099
FH[4],[16:5]=P(DUP),[16:5]=1;	18361100
\$ POP OMIT	18361101
FH[4],[6:1]=0;	18361200
GO TO LW;	18362000
CLOSEOUTPUT;	18363000
\$ SET OMIT = NOT(SHAREDISK)	18363999
P(SYSNO); ;P(,TOG2,+,LOD,NOT); T+P;	18364000
FH[9]+P(DUP,LOD) AND T;	18365000
\$ POP OMIT	18365001
\$ SET OMIT = SHAREDISK	18365099
FH[9],[5:1]=0;	18365100
\$ POP OMIT	18365101
GO TO CLOSE;	18366000
CLOSEWRITELOCK;	18367000
\$ SET OMIT = NOT(SHAREDISK)	18367999
P(SYSNO); ;P(,ONE,+,LOD); T+P;	18368000
FH[9],[9:20]+P(DUP),[9:20]=T;	18369000
\$ POP OMIT	18369001

\$ SET OMIT = SHAREDISK	18369099
FH[9],[9:5]:=P(DUP).[9:5]=1;	18369100
\$ POP OMIT	18369101
GO TO LW;	18370000
CLOSEXCLUSIVE;	18371000
FH[4],[1:2]=0;	18372000
GO TO CLOSE;	18373000
CLOSEPROTECT;	18374000
\$ SET OMIT = NOT SHAREDISK	18374001
P(SYSNO);        :P(,ONE,+,LOD); T=P;	18374050
IF ((I+FH[9],[9:20]=T) AND T*31)=0 THEN	18374100
BEGIN	18374150
P(SYSNO);    :P(,TOG1,+,LOD,NOT); T=P;	18374200
P(SYSNO);    :P(,TOG2,+,LOD,NOT); T=P AND T;	18374250
FH[9]=P(DUP,LOD) AND T;	18374300
END;	18374350
FH[9],[9:20]=1;	18374400
GO TO CLOSE;	18374500
\$ POP OMIT	18374501
\$ SET OMIT = SHAREDISK	18374599
GO TO CLOSEXCLUSIVE;	18374600
\$ POP OMIT	18374601
\$ SET OMIT = NOT SHAREDISK	18374999
ONE    ::: 32768, 1024, 32, 1;	18375000
TOG1   ::: @2000000000000000, @1000000000000000,	18376000
@4000000000000000, @2000000000000000;	18377000
TOG2   ::: @1000000000000000, @4000000000000000,	18378000
@2000000000000000, @1000000000000000;	18379000
\$ POP OMIT	18379001
SYS;	18388000
IF FH[9],[1:8]=0 THEN	18389000
BEGIN	18390000
\$ SET OMIT = NOT(SHAREDISK)	18390999
P(SYSNO);    :P(,STOG,+,LOD); T=P;	18391000
FH[4]=P(DUP,LOD) OR T;	18392000
\$ POP OMIT	18392001
\$ SET OMIT = SHAREDISK	18392099
FH[4],[12:1]=1;	18392100
\$ POP OMIT	18392101
GO TO LWRITE;	18393000
END;	18394000
HOLD;	18395000
GO TO SYS;	18396000
UNSYS;	18397000
\$ SET OMIT = NOT(SHAREDISK)	18397999
P(SYSNO);    :P(,STOG,+,LOD,NOT); T=P;	18398000
FH[4]=P(DUP,LOD) AND T;	18399000
\$ POP OMIT	18399001
\$ SET OMIT = SHAREDISK	18399099
FH[4],[12:1]=0;	18399100
\$ POP OMIT	18399101
GO TO LW;	18400000
LOCKSYS;	18401000
OPTN:=4;	18402000
GO SEE;	18403000
\$ SET OMIT = NOT(SHAREDISK)	18403999
STOG   ::: @400000000000, @200000000000,	18404000
@100000000000, @400000000000;	18405000
\$ POP OMIT	18405001
LOGIT;	18412000

IF FH[4],[12:4]#0 THEN GO UNLOCKHDR;	18413000
IF FH[4],[2:1] THEN	18414000
BEGIN	18415000
HOLD;	18416000
GO LOGIT;	18417000
END;	18418000
DISKLOG(A,B,FH);	18419000
GO TO LW;	18420000
GETAROW:	18421000
IF FH[12]=OPTN,[FF]#0 THEN	18422000
BEGIN	18423000
DISKLOG(A,B,FH);	18425000
IF (FH[12]=GETUSERDISK(FH[8]&3[1:46:2]))#0 THEN	18425100
BEGIN	18425150
\$ SET OMIT = SHAREDISK	18425175
UNLOCKDIRECTORY;	18425200
\$ SET OMIT = NOT SHAREDISK	18425225
UNLOCK(K);	18425250
\$ POP OMIT OMIT	18425275
I1:=GETUSERDISK(-FH[8]);	18425300
\$ SET OMIT = SHAREDISK	18425390
LOCKDIRECTORY;	18425400
\$ POP OMIT	18425410
IF SEARCH=0 THEN	18425500
BEGIN FORGETUSERDISK(I1,FH[8]);	18425600
TEMP:=0; A:=-1;	18425700
GO TO EXIT;	18425800
END;	18425900
HEADER;	18426000
FH[12]:=I1;	18426100
END;	18426200
END;	18427000
FOR I2:=FH[9],[43:5]+9 STEP -1 UNTIL 10 DO	18428000
M[OPTN INX I2]:=FH[I2];	18429000
GO TO LW;	18430000
CLOSE:	18431000
IF B.[3:1] THEN FH[4],[11:1] + 1;	18431050
IF OPTN GTR 511 THEN	18431100
BEGIN	18431200
IF (FH[0] EQV M[OPTN])#NOT 0 THEN	18431300
IF (I1:=((((I1:=(((I2:=FH[7]+1) DIV (I3:=FH[0]),[30:12])	18431400
×I3,[42:6])×30)+(I2 MOD I3,[30:12])	18431500
×(IF (I2=I3,[1:14])=0 THEN 30 ELSE I2)) DIV 30)	18431600
DIV (I3:=M[OPTN]),[42:6])×I3,[30:12]	18431700
+((((I1 DIV 30) MOD I3,[42:6])×30	18431800
+I1 MOD 30+I3,[1:14]-1) DIV I3,[1:14])=1)	18431900
#M[OPTN+7] THEN GO TO LW;	18432000
FH[0]:=M[OPTN];	18432100
FH[4]:=(+P(DUP)) OR (M[OPTN+4] AND 16);	18432150
FH[7]:=M[OPTN+7];	18432200
END;	18432300
GO TO LW;	18432400
L71X	18432500
IF (FH[4] AND #1400777777770000)#0 OR	18432600
FH[9],[1:28]#0 THEN	18433000
BEGIN	18434000
HOLD;	18435000
GO TO L7;	18436000
END;	18437000
L61X	18438000

```

IF FH[4],[43:2] NEQ 0 THEN % TEST FILE SENSITIVE
BEGIN
  STREAM(A,B,T:=T:=GETSPACE(10,64,0)+4);
  BEGIN
    DS:=10LIT"CC REMOVE "; SI:=LOC A; SI:=SI+1; DS:=7CHR;
    DS:=LIT"/"; SI:=LOC B; SI:=SI+1; DS:=7CHR;
    DS:=6LIT";END, ";
  END;
  FH[4],[43:2]:=1;
  CCARD(T8(IF UNITNO NEQ 0 THEN UNITNO ELSE 31)[2:42:6]
    &1[8:47:1]);
$ SET OMIT = NOT(SHAREDISK)
  UNLOCK(S1); OPTN:=0;
$ POP OMIT
  GO COMPLETE;
  END;
  LBMESS(ABS(A),ABS(B),7,0,0,UNITNO,LIBMSG);
  IF NOT FH[4],[1:1] THEN DISKLOG(A,B,FH);
  PBCOUNT:=PBCOUNT-(((A,[6:42] EQV "PBD ")=NOT 0) OR
    ((A,[6:42] EQV "PUD ")=NOT 0)) AND B,[CF]=1);
LB: REMOVER;
  IF P(OPTN NEQ 8),DUP) THEN
  FOR I + 1 STEP 1 UNTIL FH[9] DO%
  IF FH[9+I]#0 THEN FORGETUSERDISK(FH[I+9],FH[8]);%
  IF P THEN GO LW ELSE GO EXIT;
LWRITE:
  IF NOT B,[2:1] THEN
  STREAM(A+[DATE],B+[FH[3]],C+0);
  BEGIN SI=A;DI=LOC C;DS=8 OCT;SI=LOC C;SI=SI+5;
    DI=B;DI=DI+2;DS=3 CHR;
  END;
LW:
  IF FH[4],[3:1] OR TOG THEN FILEHOLD(A,B,TOG,TEMP,0);
$ SET OMIT = NOT SHAREDISK
  IF (OPTN OR 1)#7 THEN
$ POP OMIT
  LWS: DISKWAIT(F,-30,K);
  EX: FH[9]:=(#P(DUP)) AND 31;
  EXIT;%
  IF A,[1:1] OR TEMP<64 AND TEMP>0 THEN FORGETSPACE(F);
$ SET OMIT = SHAREDISK
  UNLOCKDIRECTORY;
$ POP OMIT
LEAVELKD:
UNUSED:
  FORGETSPACE(N);
  DIRECTORYSEARCH+TEMP;
END; % OF DIRECTORYSEARCH
PROCEDURE COMMUNICATE1;
BEGIN REAL R4=-4,R5=-5,R6=-6,R7=-7,R8=-8;
  INTEGER I4=-4,I5=-5,I6=-6;
  ARRAY A4=-4[*],A5=-5[*],A6=-6[*];
  ARRAY A7=-7[*];
  NAME N4=-4,N5=-5,N6=-6;
  LABEL C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,
    C17,C18,C19,C20,C21,C22,C23,C24,C25,C26;
  LABEL C27,C28,C29,C30,C31,C32;
  LABEL C33,C34,C35,C36,C37,C38,C39;
  LABEL C45,C49,C30A,C30B,C49A,INIT,CHANGENAME;
  SWITCH C:=C0,INIT,INIT,INIT,C4,INIT,INIT,INIT,INIT,INIT,INIT,

```

```

18438100
18438110
18438120
18438130
18438140
18438150
18438155
18438160
18438170
18438180
18438190
18438202
18438204
18438206
18438210
18438220
18439100
18441000
18442000
18443000
18444000
18444500
18445000
18446000
18447000
18453500
18454000
18455000
18456000
18457000
18458000
18459000
18460000
18460490
18460500
18460510
18461000
18462000
18463000
18465000
18465990
18466000
18466010
18466100
18466101
18466200
18467000
18468000
18500000
18500100
18500200
18500300
18500400
18500500
18500600
18500700
18500800
%026-18500900
18501000
18501100

```

```

INIT,INIT,INIT,INIT,C15,C16,INIT,INIT,INIT,INIT,
INIT,INIT,INIT,INIT,INIT,INIT,INIT,INIT,INIT,INIT,C30,
INIT,INIT,C33,INIT,INIT,INIT,INIT,INIT,INIT,INIT,INIT,
INIT,INIT,INIT,INIT,INIT,INIT,INIT,INIT,INIT,C49;
$ SET OMIT = NOT SHAREDISK
DEFINE
    TIMELIMITMAX=15#; % MAX WAIT TIME FOR LOCKED ADDRESS
    TIMELIMITMIN=0#; % CAN BE ADJUSTED TO SUIT SITE
$ POP OMIT
    REAL RCW=+0,I,J,T;
    ARRAY A[*],FIB=A[*],FPB[*],H[*];
    BOOLEAN DS;
    GO TO C[PRT[P1MIX,9]];
INIT; GO TO INITIATE;
% COBOL INVALID EOJ
C0: TERMINATE(P1MIX); TERMINALMESSAGE(28);
% GENERALIZED ZIP
C4: IF (I+A4,[8:10])#0 THEN BEGIN
    $ SET OMIT = PACKETS
        T:=GETSPACE(I+4,64,5)+4;
    $ POP OMIT
    $ SET OMIT = NOT(PACKETS)
        M[(T:=GETSPACE(I+5,64,5)+4)-4],[9:6]#=0;
    $ POP OMIT
        IF NOT A4,[2:1] THEN MAKEPRESENT(NFLAG(NOT 3 INX [RCW]));
        J := USERCODE[P1MIX];
        STREAM(C+J,A4,I+1,[36:6],I,Q+0,T);
        BEGIN SI:=A4; SI:=SI-1;
        L: SI:=SI+1; IF SC=" " THEN GO TO L; Q:=SI; DI:=Q;
            IF SC=@14 THEN DS:=LIT" " ELSE DS:=2LIT" "; DI:=T;
            DS:=8LIT"CC USER="; SI:=LOC C; SI:=SI+1; DS:=7 CHR;
            DS:=LIT";" ; SI:=A4;
            I1(DS:=32WDS; DS:=32WDS); DS:= I WDS;
    $ SET OMIT = NOT(PACKETS)
        DS:=8 LIT"+";
    $ POP OMIT
        TALLY:=12; I:=TALLY;
        DI:=Q; SI:=LOC I; SI:=SI+7; DS:=CHR;
        END STREAM;
        J*IF USERCODE[P1MIX]=MCP THEN 31 ELSE 26;
    $ SET OMIT = PACKETS
        CCARD(T&P1MIX[18:42:6]&J[3:43:5]);
    $ SET OMIT = NOT(PACKETS)
        IF PSEUDOMIX[P1MIX] NEQ 0 THEN NYLONZIPPER[P1MIX],[2:1]#=0;
        CCARD(T&P1MIX[18:42:6]&J[3:43:5]);
        IF PSEUDOMIX[P1MIX] NEQ 0 THEN
            IF MEMROW[P1MIX],[CF] GEQ FENCE THEN
                DO SWAP(WAITSWAP,1) UNTIL
                    NYLONZIPPER[P1MIX],[2:1] ELSE
            IF PSEUDOMIX[P1MIX] NEQ 0 THEN
                SLEEP([NYLONZIPPER[P1MIX]],@1000000000000000);
    $ RESET OMIT
        END ELSE
        BEGIN FIB=N4[NOT 2];
            FPB=PRT[P1MIX,3];
            I*IF FIB[4],[12:1] THEN FIB[4],[13:11]
                ELSE (FIB[4],[13:11]=1)*ETRLNG;
            T*FPB[I+3],[43:5];
            IF T=10 OR T=12 OR T=13 OR T=26 THEN
                BEGIN IF FIB[5],[42:1] THEN GO TO CHANGENAME;

```

```

18501200
18501300
18501400
18501500
18501600
18501700
18501800
18501900
18502000
18502100
18502200
18502300
18502600
18505000
18510000
18510100
18520000
18520100
18520200
18520300
18520400
18520500
18520600
18520700
18520800
18520900
18521000
18521100
18521200
18521300
18521400
18521500
18521600
18521700
18521800
18521900
18522000
18522100
18522200
18522300
18522400
18522500
18522600
18522700
18522800
18522820
18522840
18522860
18522880
18522900
18523000
18523100
18523200
18523300
18523400
18523500
18523600
18523700
18523800
18523900

```

H*FIR[14];	18524000
\$ SET OMIT = NOT(PACKETS)	18524100
H[6]:=(P(DUP))&3[2:42:6];	18524200
\$ POP OMIT	18524300
H[5]:=USERCODE[P1MIX];	18524400
\$ SET OMIT = NOT(SHAREDISK)	18524500
H[4],[7:11]:=1;	18524600
\$ POP OMIT	18524700
IF H[4] THEN *	18524800
BEGIN FILECLOSE(N4,[33:15]);	18524900
CHANGENAME: IF (T*DIRECTORYSEARCH(FPB[I],FPB[I+1],4))	18525000
LSS 64	18525050
THEN GO TO INITIATE;	18525100
H+[M[T]]&30[8:38:10];	18525200
\$ SET OMIT = NOT(SHAREDISK)	18525300
H[4]:=(P(DUP))&1[7:47:1] OR 1;	18525400
\$ POP OMIT	18525500
H[5]:=USERCODE[P1MIX];	18525600
\$ SET OMIT = NOT(PACKETS)	18525700
H[6]:=(P(DUP))&3[2:42:6];	18525800
\$ POP OMIT	18525900
ENTERCONTROLDECK(H);	18526000
P(DIRECTORYSEARCH(=FPB[I],FPB[I+1],8),DEL);	18526100
J*H[2]; % SAVED LASTCDNUM	18526300
FORGETSPACE(H);	18526400
END ELSE	18526500
BEGIN FILECLOSE((N4,[33:15])&6[18:33:15]);	18526600
ENTERCONTROLDECK(H);	18526700
J*H[2]; % SAVED LASTCDNUM	18526800
FOR T*10 STEP 1 UNTIL 29 DO H[T]*0;	18526900
FILECLOSE(N4,[33:15]);	18527000
END;	18527100
IF RUNUMBER LEQ 0 THEN	18527200
BEGIN	18527300
STREAM(A+[JAR[P1MIX*0]], B*H*USERCODE[P1MIX],	18527400
F*H*MCP AND H#0, P1MIX, J, T+[SPACE(10)]);	18527500
BEGIN SI*A; DS*LIT"#";	18527600
2(SI*SI+1; DS*7 CHR; DS*LIT"/"); DI*DI+1;	18527700
F(SI*LOC B; SI*SI+1; DI*DI+1; DS*7 CHR);	18527800
SI*LOC P1MIX; DS*LIT"#"; A*DI;	18527900
DS*2 DEC; DS*14 LIT" ZIPPED DECK #";	18528000
SI*LOC J; DS*4 DEC; DS*LIT"#";	18528100
DI*DI+5; DS*3 FILL; DI*A; DS*FILL;	18528200
END;	18528300
SPOUT(T);	18528400
END;	18528500
END;	18528600
END;	18528700
GO TO INITIATE;	18528800
C15: DISPLAY(A4 INX 1);	18530000
GO TO INITIATE;	18530100
* COBOL ACCEPT	18540000
C16: DISPLAY(A4 INX 2); REPLY[P1MIX]*=VWY&VAX[36:42:6];	18540100
IF [MEM[P1MIX,MLINK1]],[CF]#FENCE THEN	18540200
SWAP(WAITSWAP,1) ELSE	18540300
COMPLEXSLEEP((TERMSET(P1MIX) OR REPLY[P1MIX]>0));	18540400
IF TERMSET(P1MIX) THEN GO TO INITIATE;	18540500
IF NOT WHYSLEEP(VWY&VAX[36:42:6]) THEN GO TO C16;	18540600
T*REPLY[P1MIX],[18:15]; REPLY[P1MIX]*0;	18540700
STREAM(T,S*A4 INX 2);	18540800

```

BEGIN SI+T; 18540900
L: IF SC#"X" THEN BEGIN SI+SI+1; GO TO L END; 18541000
SI+SI+1; 2(DS+40 CHR); 18541100
END; 18541200
FORGETSPACE(T-1); GO TO INITIATE; 18541300
% DIRECTORYSEARCH AND UN-FILL FILE ID FOR NORMAL STATE PROGRAMS 18550000
C30: COMMENT SEARCHES DISK DIRECTORY AND RETURNS DATA IN ARRAY, 18550100
[0] IS USER=TYPE OR NOT=PRESENT FLAG 18550200
[1] IS MULTI=FILE ID 18550300
[2] IS FILE ID 18550400
IF NOT PRESENT, [3] => [5] ARE =1 18550500
IF INVALID USER, [3] => [5] ARE 0 18550600
IF PRIMARY, SECONDARY, OR TERTIARY USER: 18550700
[3] IS RECORD LENGTH 18550800
[4] IS BLOCK LENGTH 18550900
[5] IS END OF FILE POINTER 18551000
[6] IS OPEN COUNT 18551100
IF ARRAY SIZE IS GREATER THAN 9: 18551200
[7] = FILETYPE (FROM HEADER) 18551300
[8] = HEADER[3] (CREATION/ACCESS DATE,SAVE FACTOR) 18551400
[9] = HEADER[1] ( LOGGING DATES) 18551500
IF ARRAY SIZE IS GREATER THAN 10: 18551600
[10]= SYSTEM NUMBER (SHAREDISK) 18551700
NOT=PRESENT FLAG IS =1 18551800
INVALID USER FLAG IS 0 18551900
PRIMARY USER FLAG IS 7 (LM, INPUT, AND OUTPUT BITS) 18552000
SECONDARY USER FLAG IS 3 (INPUT AND OUTPUT BITS) 18552100
TERTIARY USER FLAG IS 2 (INPUT BIT ONLY) 18552200
; 18552300
IF A4,[8:10]<7 THEN BEGIN TERMINATE(P1MIX);% 18552400
TERMINALMESSAGE(7); END;% 18552500
IF NOT A4,[2:1] THEN MAKEPRESENT(NFLAG(NOT 3 INX [RCW]));% 18552600
P([M[A4 INX NOT 1]],DUP,DUP,LOD,XCH,CCX,,J,STD,IOR);% 18552700
FIB * N5(NOT 2); FPB * PRT[P1MIX,3]; 18552800
I * IF FIB[4],[12:1] THEN FIB[4],[13:11];% 18552900
ELSE (FIB[4],[13:11]=1)*ETRLNG; 18553000
A4[1] * FPB[1]; A4[2] * FPB[1+1]; 18553100
IF P(FPB[1+3],[43:5],DUP,DUP)=10 %RANDOM 18553200
OR (P(XCH) OR 1)=13 OR P(XCH)=26 THEN %SERIAL,UPDATE,PROTECT 18553300
BEGIN 18553400
IF A4[1]=0 THEN 18553500
BEGIN A4[1]:=A4[2]; 18553600
A4[2]:=USERCODE[P1MIX]; 18553700
END; 18553800
IF (T:=DIRECTORYSEARCH(A4[1],A4[2],5)) # 0 THEN 18553900
BEGIN IF (A4[0]:=SECURITYCHECK(A4[1],A4[2],USERCODE[P1MIX],T)) 18554000
#0 AND M[T+4],[12:4]=0 THEN 18554100
BEGIN A4[3]:=M[T],[1:14]; 18554200
A4[4] * M[T],[15:15]; A4[5] * M[T+7];% 18554300
$ SET OMIT = SHAREDISK 18554400
A4[6]:=M[T+4],[16:5]+M[T+9],[9:5]; 18554500
$ POP OMIT 18554600
$ SET OMIT = NOT(SHAREDISK) 18554700
A4[6]:=0&([1:=M[T+4],[16:20]+M[T+9],[9:20])[43:28:5] & 18554800
I[38:33:5]&I[33:38:5]&I[28:43:5]; 18554900
$ POP OMIT 18555000
IF A4,[8:10] GTR 9 THEN %R2218555100
BEGIN A4[7]:=M[T+4],[36:6];A4[8]:=M[T+3]; %R2218555200
A4[9]:=M[T+1]; %R2218555300
END; %R2218555400

```

```

        IF A4,[8:10] GTR 10 THEN A4[10]:=SYSNO;
        END ELSE A4[3]:=A4[4]:=A4[5]:=A4[6]:=0;
        FORGETSPACE(T);
        GO TO C30B
    END ELSE GO TO C30A;
END ELSE
BEGIN
    T:=-1;
    IF (T:=FINDINPUT(A4[1],A4[2],FPB[1+2],[1:17],
        FPB[1+2],[18:30],FPB[1+3],[1:5],[A4[3]] INX 0,
        T,0,0,0))=NABS(1) THEN GO TO C30A ELSE
    IF T GEQ 0 THEN
    BEGIN
        A4[0]:=4; A4[3]:=([:=RDCTABLE[T]),[14:10]];
        A4[4]:=1,[24:17]; A4[5]:=1,[41:7];
        A4[6]:=TINU[T],[30:18]; IF T<16 THEN
        A4[6]:=(*P(DUP))&PRNTABLE[T][12:30:18]; GO C30B;
    END ELSE
    BEGIN
        A4[0]:=5; A4[3],[1:5]:=ABS(T); GO C30B
    END;
END;
C30A: A4[0]:=A4[3]:=A4[4]:=A4[5]:=A4[6]:=-1;
C30B:
    IF NOT J,[2:1] THEN P([M[J]],PRL);%
    GO TO INITIATE;%
C33: STREAM(R4,A+(R4#0),J+JARROW[P1MIX],P1MIX,%
    T:=T:=SPACE(10));
    BEGIN DS+10 LIT " PAUSE # 0";%
        A(DI+DI-1; SI+LOC R4; SI+SI+2; DS+6 CHR);
        DS+5 LIT " FOR"; SI+J; SI+SI+1; DS+7 CHR;%
        DS+LIT "/"; SI+SI+1; DS+7 CHR; DS+LIT "=";%
        SI+LOC P1MIX; DS+2 DEC; DS+LIT "+"; DI+DI-3; DS+FILL;%
    END;%
    SPOUT(T);%
    IF NOTERMSET(P1MIX) THEN PRTRW[P1MIX],[PSF]+2;
    GO TO INITIATE;% DON'T KEEP COMMUNICATE AROUND NEEDLESSLY
C49:
    $ SET OMIT = NOT SHAREDISK
    IF I4 THEN GO TO C49A;
    IF (M[A6] AND @3000000000)=0 THEN
    SLEEP([M[A6]],@3000000000);
    IF (M[A6] AND IOMASK)#0 THEN GO TO RETURN;
    IF (I:=I5 GTR TIMELIMITMAX) THEN I5:=TIMELIMITMAX;
    IF I5<TIMELIMITMIN THEN I5+TIMELIMITMIN;
    IF (M[A6] AND IOMASK)=0 THEN
    IF (I5+I5*60+CLOCK+P(RTR))>CLOCK+P(RTR) THEN
    COMPLEXSLEEP((T:=(CLOCK+P(RTR)) GTR I5) OR % TIMELIMIT
        (TERMSET(P1MIX)) OR % DSED
        ((M[A6] AND IOMASK)#0)); % IOCOMPLETE
    IF (M[A6] AND IOMASK)#0 THEN GO TO RETURN;
    DSI=T=1 AND I=1;
    M[A6]:=(*P(DUP))&3[19:46:2]; R6+M[M[A6]];
C49A: R4+R5+0;
    FOR I+0 STEP 1 UNTIL (LQAVAIL=1) DO
    IF ((J+LQUE[I]),[8:40] EQV R6,[8:40])=NOT 0 THEN
    IF LOCATQUE[J,[1:7]],[3:5]=P1MIX THEN
    BEGIN
        IF I < (LQAVAIL+LQAVAIL-1) THEN
        STREAM(A+LQAVAIL-I,B+[LQUE[I]]);

```

```

18555500
18555600
18555700
18555800
18555900
18556000
18556100
18556200
18556300
18556400
18556500
18556600
18556700
18556800
18556900
18557000
18557100
18557200
18557300
18557400
18557500
18557600
18557700
18557800
18557900
18558000
18570000
18570100
18570200
18570300
18570400
18570500
18570600
18570700
18570800
18570900
18571000
18590000
18590100
18590200
18590300
18590400
18590500
18590600
18590700
18590800
18590900
18591000
18591100
18591200
18591300
18591350
18591400
18591500
18591600
18591700
18591800
18591900
18592000
18592100

```



```

        BEGIN SI+8; SI+SI+8; DS+A WDS END;
        RETURNIOSPACE(J,[1:7]);
        R4+1;
END ELSE R5+1;
IF NOT (R4 AND R5) THEN
BEGIN
    IF NOT R4 THEN R6+(R6 OR @2060) ELSE % UNLOCK ADDRESS
    R6+(R6 OR @60)&SYSNO[30:46:2]; % CLEAR CONTENTION
    P(WAITIO([R6] INX @100000000,0,18),DEL);
END;
IF DS THEN BEGIN TERMINATE(P1MIX); TERMINALMESSAGE(89); END;
$ POP OMIT
GO INITIATE;
END OF COMMUNICATE1;
PROCEDURE COMMUNICATE0;
BEGIN REAL R4=-4,R5=-5,R6=-6,R7=-7,R8=-8;
    INTEGER I4=-4,I5=-5,I6=-6;
    ARRAY A4=-4[*],A5=-5[*],A6=-6[*];
    ARRAY A7=-7[*];
    NAME N4=-4,N5=-5,N6=-6;
    LABEL C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,
        C17,C18,C19,C20,C21,C22,C23,C24,C25,C26;
    LABEL C27,C28,C29,C30,C31,C32;
    LABEL C33,C34,C35,C36,C37,C38,C39;
    LABEL C48,C3A,C21A,INIT,IT,US,D,TD,PR,IOT,TMR,AD,WD;
    SWITCH S:=IT,US,D,TD,PR,IOT,TMR,AD,WD;
    REAL I,J,T,K;
    ARRAY AIT[*];REAL AITL=AIT; ARRAY A=AIT[*];
    NAME ADDR;
    SWITCH C:=INIT,C1,INIT,C3,INIT,INIT,C6,C7,C8,INIT,INIT,
        INIT,INIT,INIT,INIT,INIT,INIT,C17,INIT,INIT,C20,
        C21,C22,INIT,INIT,C25,C26,INIT,INIT,C29,INIT,
        INIT,INIT,INIT,INIT,INIT,INIT,INIT,C38,C39,INIT,
        INIT,INIT,INIT,INIT,INIT,INIT,INIT,C48,INIT;
    DEFINE TIMELIMITMAX=15#;
    GO TO C[PRT[P1MIX,9]];
INIT: GO TO INITIATE;
% TIME INTRINSIC
C1: IF (I4+I4) GEQ (-2) AND I4 LEQ 6 THEN
    BEGIN GO TO S[I4+2];
    IT: I4+JAR[P1MIX,2],[5:1];
        JAR[P1MIX,2]+(*P(DUP)) & 2[4:46:2];
        GO INITIATE;
    US: R4+USERCODE[P1MIX]; GO TO INITIATE;
    D: I4+DATE; GO TO INITIATE;
    TD: I4+XCLOCK+P(RTR); GO TO INITIATE;
    PR: I4+JAR[P1MIX,3]+PROCTIME[P1MIX]+CLOCK+P(RTR);
        GO TO INITIATE;
    IOT: I4+IOTIME[P1MIX]+JAR[P1MIX,4];
        GO TO INITIATE;
    TMR: I4+P(RTR); GO TO INITIATE;
    AD: I4+ACTDATE; GO TO INITIATE;
    WD: I4+WEEKDAY;
    END;
    GO TO INITIATE;
% RETURN SPECIFIC ARRAY
C3: ARTN(N4[0],1); % RETURN 1 DIM ARRAY
C3A: T:=AITL,[CF]; % REMOVE FROM AIT
    IF NOT(AIT+PRT[P1MIX,6]),[2:1] THEN MAKEPRESENT(T);
    J ← AIT[0]; T ← N4,[CF];

```

```

18592200
18592300
18592400
18592500
18592600
18592700
18592800
18592900
18593000
18593100
18593150
18593200
18593300
18599000
18700000
18700100
18700200
18700300
18700400
18700500
18700600
18700700
18700800
%026-18700900
18701000
18701100
18701200
18701300
18701400
18701500
18701600
18701700
18701800
18701900
18702100
18702200
18705000
18710000
18710100
18710200
18710300
18710400
18710500
18710600
18710700
18710800
18710900
18711000
18711100
18711200
18711300
18711400
18711500
18711600
18711700
18720000
%026-18720100
%026-18720200
18720300
18720400

```

```

FOR I=1 STEP 1 UNTIL J=1 DO
    IF AIT[I],[18:15]=T THEN
        BEGIN MOVE(J=I,[AIT[I+1]],[AIT[I]]); J=0 END;
    IF J=0 OR AIT[J],[FF]=T THEN AIT[0] = *P(DUP)=1;
N4[0]=0;
GO TO INITIATE;
%
C6: WHEN
    IF I4 GTR TIMELIMITMAX THEN
        BEGIN TERMINATE(P1MIX); TERMINALMESSAGE(89); END;
    I4:=60*I4+CLOCK+P(RTR);
    WHILE NOTERMSET(P1MIX) AND CLOCK+P(RTR)<I4 DO
        SLEEP([CLOCK],NOT CLOCK);
    GO TO INITIATE;
C7: IF NOT A5,[2:1] THEN MAKEPRESENT(NFLAG(NOT 0 INX [I4]));
    I=M[A5 INX NOT 0]; J=M[A5 INX NOT 1];
    P([M[A5 INX NOT 1]],[IOR]);
    IF (NT2+(NT1+(I4 INX PRT[P1MIX,4])),[18:15])>NT3+A5,[8:10] THEN
        NT2=NT3;
    DISKWAIT(=A5,[CF],NT2,I4+JAR[P1MIX,NT1],[CF]
        DIV JAR[P1MIX,8]+10]+NT1,[33:15] MOD JAR[P1MIX,8]);
    M[A5 INX NOT 0]+*P(,I);
    IF NOT (*P(,J)),[2:1] THEN P([M[A5 INX NOT 1]],[PRL]);
    GO TO INITIATE;
C8: ZIPPER(R5,R4);
    GO TO INITIATE;
%
C17: COBOL I/O ERROR
    A5=*A5; A=PRT[P1MIX,3]; I="I/O ERR";
    IF A5[5],[1:1] THEN
        BEGIN I="INVALID";J=" USER"; R6:=1 END ELSE
    STREAM(R4,N+[J]); BEGIN SI=LOC R4; DI=DI+1; DS=7 DEC;
        DI=DI-7; DS=5 FILL;
    END;
    FILEMESS(I&R6[1:47:1],J,A[T+A5[4],[13:11]],A[T+1],
        IF R4+(R4=16 OR R4=17 OR R4=82) THEN R8 ELSE 0,
        IF R4 THEN R7 ELSE 0,0);
    GO TO INITIATE;
%
C20: TAPE SWAP FOR TAPE SORT
    SLEEP([N4[3]],[IOMASK]); SLEEP([N4[4]],[IOMASK]);
    FOR I=3 STEP 1 UNTIL 4 DO
        BEGIN N5[I],[33:15]=N4[I];
            M[N4[I] INX NOT 1]+(*P(DUP))&N5[3][14:3:4]&[N5[3]][33:33:15];
        END;
    A=N4[0]; A[5],[39:4]+2; A[16]=0; A[18]+NABS(*P(DUP));
    A=N5[0]; A[5]=0; A[16]+NFLAG(N5[3]); A[18]+ABS(*P(DUP));
    GO TO INITIATE;
%
C21: SORT STORAGE ASSIGNMENT
    A=[M[GETSPACE(R6+R5,2,1)+2]]&R5[8:38:10];
    A[0]+(R5 INX A)&(A)[CTF]&R6[8:38:10];
    N4[0]=A;
    IF NOT CONQUER(0,R5=1,(R6&(NOT J))[1:47:1]),1 INX A,J) THEN
        BEGIN FORGETSPACE(A);
C21A: STREAM(P1MIX,T+R5*R6,A+I+SPACE(7));
        BEGIN DS=LIT "#"; SI=LOC P1MIX;
            DS=2 DEC; DS=13 LIT " NO SORT MEM!";
            DS=5 DEC; DS=9 LIT " WDS RQD=";
        END;
    SPOUT(I);
    REPLY[P1MIX]=VWY&VOK[36:42:6]&VOU[30:42:6];
    IF MEMROW[P1MIX],[CF] GEQ FENCE THEN SWAP(WAITSWAP,1) ELSE
    COMPLEXSLEEP(REPLY[P1MIX]>0 OR TERMSET(P1MIX));

```

```

18720500
18720600
18720700
18720800
18720900
18721000
18730000
18730100
18730110
18730120
18730200
18730300
18730400
18740000
18740100
18740200
18740300
18740400
18740500
18740600
18740700
18740800
18740900
18750000
18750100
18770000
18770100
18770200
18770300
18770400
18770500
18770600
18770700
18770800
18770900
18771000
18780000
18780100
18780200
18780300
18780400
18780500
18780600
18780700
18780800
18780900
18790000
18790100
18790200
18790300
18790400
18790500
18790600
18790700
18790800
18790900
18791000
18791100
18791200
18791300
18791400

```

```

IF TERMSET(P1MIX) THEN GO TO INITIATE;
IF NOT WHYSLEEP(VWY&VOK[36:42:6]&VOU[30:42:6]) THEN GO TO C21A;
J*REPLY[P1MIX],[CF]=VOU;
GO TO C21;
END;
GO TO INITIATE;
% SORT STORAGE RETURN
C22: J*(A+FLAG(N4[0])),[B:10]=1;
FOR I*1 STEP 1 UNTIL J DO
IF T>(T+A[I],[CF]) OR T>K THEN
BEGIN K*M[IT=2],[CF]; FORGETSPACE(T) END;
FORGETSPACE(N4[0]); N4[0]*0;
GO TO INITIATE;
% RETURN OLD COPY OF OWN ARRAY
C25: ARTN(A5,R4);
M[A5,[FF]]*A+PRT[P1MIX,17]&P(,A5,LOD)[18:18:15];
IF A,[2:1] THEN M[A,[CF]=1],[CF]*A5,[FF];
GO TO INITIATE;
% INVALID ARGUMENTS TO ALGOL INTRINSICS
C26: IF (I + R4)≥0 THEN
STREAM(A:=R4, I:=1:=SPACE(10));
BEGIN SI*LOC I; SI*SI-1; DS*LIT "-";
IF SC≥3 THEN DS*4 LIT "MAXN" ELSE
IF SC<2 THEN DS*5 LIT "NEGTV" ELSE DS*4 LIT "ZERO";
DS*8 LIT " ARGMENT ";
CI*CI+A;
GO LOG; GO ROOT; GO LOG; GO EXP; GO SIN;
DS*3 LIT "COS"; GO EXIT;
LOG: DS*2 LIT "LN"; GO EXIT;
ROOT: DS*4 LIT "SQRT"; GO EXIT;
EXP: DS*3 LIT "EXP"; GO EXIT;
SIN: DS*3 LIT "SIN";
EXIT: DS:=2 LIT "i=";
END;
IF I = (-7) THEN % COBOL INVALID INDEX
BEGIN
R4 * R5; R5 * R6;
ERRORFIXER(4); % INVALID INDEX CHECK
END;
TERMINATE(P1MIX); TERMINALMESSAGE(-1);
%
C29: COMMENT THIS COMMUNICATE PROVIDES FOR DS=ING AN OBJECT PROGRAM
AND/OR SPOUTING A MESSAGE ABOUT A PROGRAM,
R4 IS USED TO SPECIFY THE MESSAGE REQUIRED,
R5 SET TO TRUE SPECIFIES P1MIX IS TO BE DS=ED,
T IS THE ADDRESS OF THE MESSAGE(WHICH ENDS WITH A "+"),
REMAINING VARIABLES MAY BE USED AS DESIRED;
T I= SPACE(12);
IF R4 ≤ 2 THEN
BEGIN; % 29-1
STREAM(JIT);
BEGIN % 29-2
DS:=9 LIT "-DEC ERR.";
J * DI;
END; % 29-2
J * P;
IF R4=1 THEN
BEGIN; % 29-3

```

```

18791500
18791600
18791700
18791800
18791900
18792000
18800000
18800100
18800200
18800300
18800400
18800500
18800600
18810000
18810100
18810200
18810300
18810400
%WF 18820000
18820100
18820200
%WF 18820300
%WF 18820400
%WF 18820500
%WF 18820600
%WF 18820700
%WF 18820800
%WF 18820900
%WF 18821000
%WF 18821100
%WF 18821200
%WF 18821300
18821400
18821500
18821600
18821700
18821800
18821900
18822000
%WF 18822100
18822200
18822300
18822400
18830000
18830100
18830200
18830300
18830400
18830500
18830600
18830700
18830800
18830900
18831000
18831100
18831200
18831300
18831400
18831500
18831600

```



```

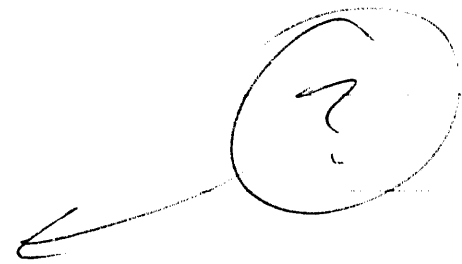
        STREAM(T1+(R6<0),R6+ABS(R6),J);
        BEGIN      % 29-4
        DS+17 LIT "ARRAY DIMENSION=";T1(DS+ 1 LIT "-");
        SI + LOC R6;
        DS + 8 DEC; J +DI;
        DI + DI-8;
        DS + 7 FILL; DI + J;
        DS:=2 LIT "1";
        END;        % 29-4
    END           % 29-3
ELSE
    BEGIN;        % 29-5
    STREAM(R6,J);
    BEGIN      % 29-6
    DS +15 LIT "NO, DISK ROWS=";
    SI + LOC R6;
    DS + 8 DEC; J + DI;
    DI + DI-8;
    DS + 7 FILL; DI + J;
    DS:=2 LIT "1";
    END;        % 29-6
    END;        % 29-5
END;          % 29-1
IF R4=3 THEN
    BEGIN
    ;STREAM(T);
    BEGIN
    DS+ 18 LIT "=MAXN ARGMNT EXP!";
    END;
    END;
IF R4 = 4 THEN STREAM(T); BEGIN
    DS:=37 LIT"ILLEGAL PERFORM = RETURN OR RELEASE!";
END;;
IF R5 THEN
    BEGIN      % 29-7
    TERMINATE(P1MIX);
    TERMINALMESSAGE(=T);
    END      % 29-7
ELSE
    SPOUT(T);
    GO TO INITIATE;
C38: % RETURN STORAGE ( AND AUXMEM ) FOR CODE OR DATA SEGMENT
IF A4,[1:1] THEN % DESCRIPTOR TO CODE SEGMENT
    BEGIN
    AI=PRT[P1MIX,*];
    TI=NFLAG(A4 & (I:=0)[5:47:1]);
    % FIND LAST DESCRIPTOR LINKED INTO THIS CODE SEGMENT
    DO IF T,[5:1] THEN I:=T,[18:15] % SEG,NO, FROM PROG,DESC,
    ELSE IF T,[6:1] THEN I:=T,[07:11] % SEG,NO,(STOPPER BIT ON)
    ELSE
        TI=NFLAG(A[T],[7:11]) % LINK TO NEXT DESC,
    UNTIL I NEQ 0;
    ADDR:= I INX A[4]; % DESCRIPTOR TO SEGMENT DICT, ENTRY
    IF ADDR[0],[3:1] THEN COMPLEXSLEEP((NOT ADDR[0],[3:1])); %INTERLOCK
    KI=(ADDR[0],[4:2] NEQ 0); % AUXMEM FLAGS
    ADDR[0],[3:2]:=2; % SET INTERLOCK,RESET [4:1] (AUXMEM FLAG)
    WAITSTORE(P1MIX); STOREDY(P1MIX):=0;
    IF (I:= (TI=ADDR[0]),[FF]) GTR 1023 THEN
    % "T" IS SEG,DICT,ENTRY,"I" IS CORE ADDRESS IF GTR 1023
    BEGIN % PRESENT SEGMENT
    J:=M[I-1]; % SECOND MEMORY LINK

```

```

18831700
18831800
18831900
18832000
18832100
18832200
18832300
18832400
18832500
18832600
18832700
18832800
18832900
18833000
18833100
18833200
18833300
18833400
18833500
18833600
18833700
18833800
18833900
18834000
18834100
18834200
18834300
18834400
18834500
18834600
18834700
18834800
18834900
18835000
18835100
18835200
18835300
18835400
18835500
18835600
18835700
18840000
18840100
18840200
18840300
18840400
18840500
18840600
18840700
18840800
18840900
18841000
18841100
18841200
18841300
18841400
18841500
18841600
18841700
18841800

```



```

P(OLAY(I=2,P1MIX),DEL);
% OLAY WILL NOT WRITE TO AUXMEM IF NOT [4:1] IN SEG,DICT,
$ SET OMIT = NOT(AUXMEM)
END % IF PRESENT SEGMENT
ELSE IF T,[5:1] THEN % CODE SEGMENT ON AUXMEM
DISKWAIT(=[J] INX 1),0,CODEADDRESS(P1MIX,T));
% "J" IS MEM,LINK FROM AUXMEM WITH SIZE AND ORIG,DISK,ADDR,
IF T,[5:1] THEN % CODE SEGMENT ON AUXMEM (PRESENT OR NOT)
BEGIN
ADDR[0]:=[*P(DUP)]&J[33:3:15]; % RESTORE ORIGINAL DISK ADDR
FORGETAUXILIARYSPACE(J,[FF],T,[CF]);
AUXCODE[P1MIX]:= *P(DUP) - J,[23:6] -1;
$ POP OMIT
END;
% RESET SEG,DICT,INTERLOCK AND RESTORE AUXMEM FLAG IF IT WAS PRESENT
ADDR[0]:=[*P(DUP)]&0[3:47:1]&K[4:47:1]&0[5:47:1];
STOREDY(P1MIX):=1;
GO TO INITIATE;
END; % CODE SEGMENTS
WAITSTORE(P1MIX); STOREDY(P1MIX):=0;
IF (T:=NFLAG(M[J]:=A4,[FF])),[2:1] THEN
% "J" IS CORE ADDRESS OF MOTHER, "T" IS MOTHER DESCRIPTOR
BEGIN % PRESENT SEGMENT
M[T INX NOT 0]:=[*P(DUP)]&([I:=P(DUP),[FF]] OR 1)[CF];
% "I" IS OLAY ADDR, OF SEGMENT FROM SECOND MEMORY LINK
% SET [FF] OF 2ND, MEM, LINK NEQ 0 SO OLAY WILL NOT GET DISK SPACE
K:=M[T INX NOT 1],[2:1]; % SAVE BIT FROM MEM, LINK
M[J],[3:3]:=7; % MARK "READ ONLY, WRITTEN" SO OLAY WILL NOT WRITE SEG.
P(OLAY(T,[CF]-2,P1MIX),DEL); % RELEASE CORE SPACE
END % IF SEGMENT WAS PRESENT
ELSE I:=T,[CF]; % OLAY ADDRESS FROM NON-PRESENT DATA DESC,
STOREDY(P1MIX):=1; % FREE MEMORY TO ALLOW "ARTN" TO BE BROUGHT IN
IF I GTR 511 THEN ARTN( ([I&T[8:8:10]]) -1); % RETN OLAY STORAGE
M[J]:=FLAG(T&0[2:42:6]&K[CTC]); % MARK NOT PRESENT WITH "SAVE" ENTRY
GO TO INITIATE;
C39: % BASIC ARRAY RETURN
ARTN(N4[0],R5); % RETURN R5 DIM ARRAY
GO TO C3A; % TO REMOVE FROM AIT
% MEMORY DUMP OR TRACE FROM THE INTRINSICS
C48: %
$ SET OMIT = NOT(DUMP OR DEBUGGING)
IF I4 NEQ 0 THEN
$ SET OMIT = NOT(DEBUGGING) OR OMIT
GO INITIATE % TRACE IS NOT INCLUDED IN THE TSS MCP,
$ POP OMIT
ELSE DUMPNOW(R5);
$ POP OMIT
GO INITIATE;%
END OF COMMUNICATED;
PROCEDURE SHORTCOMMUNICATES;
BEGIN REAL R4=-4,R5=-5,R6=-6,R7=-7,R8=-8,R9=-9;
ARRAY A4=-4[*],A5=-5[*],A6=-6[*],A7=-7[*],A8=-8[*],A9=-9[*];
REAL I=T,T1=T+1;
LABEL SLOW,INIT,TW;
LABEL C2,C5,C8,C10,C11,C13,C14,C15,C19,C23,C24,C34;
LABEL CM1,CM2,CM3,CM4,CM5,CM6,CM7,CM8,CM9
CM21, CM20
CM19,CM18,CM17,CM16,CM15,CM14,CM13,CM12,CM11,CM10
C35,C36,C37,C40,C41 %

```

```

18841900
18842000
18842100
18842200
18842300
18842400
18842500
18842600
18842700
18842800
18842900
18843000
18843100
18843200
18843300
18843400
18843500
18843600
18843700
18843800
18843900
18844000
18844100
18844200
18844300
18844400
18844500
18844600
18844700
18844800
18844900
18845000
18845100
18845200
18845300
18850000
18850100
18850200
18870000
18870100
18870200
18870300
18870400
18870500
18870600
18870700
18870800
18870900
18872000
19500000
19501000
19502000
19503000
19504000
19505000
19505100
19505150
19505200
19505300
19505499

```

```

SWITCH S+
  CM21, CM20,
  CM19, CM18, CM17, CM16, CM15, CM14, CM13, CM12, CM11, CM10,
  CM9, CM8, CM7, CM6, CM5, CM4, CM3, CM2, CM1,
  SLOW, TW, C2, TW, SLOW, C5, TW, TW, TW, INIT, C10,
  C11, INIT, C13, C14, SLOW, SLOW, TW, INIT, C19, TW, TW,
  TW, C23, C24, TW, TW, INIT, INIT, TW, SLOW, INIT,
  INIT, SLOW, C34, C35, C36, C37, TW, TW, C40, C41, INIT,
  INIT, INIT, INIT, INIT, INIT, TW, SLOW;
DEFINE TIMELIMITMAX=15#;
GO TO S[PRTP1MIX,9]+
  21
  ];
  COMMENT YOU MUST ADD NEW LABELS TO THE FRONT OF THE
  SWITCH AND CHANGE THE LITERAL ABOVE TO CHANGE
  THE NUMBER OF "NEGATIVE" COMMUNICATES;
@NIT: GO TO INITIATE;
SLOW: P(,COMMUNICATE1); GO TO DIFFCOM;
TW: P(,COMMUNICATE0); GO TO DIFFCOM;
% ONLY ON TSS IF THE SLEEP CONDITION IS NOT MET IN 15 SECONDS, DS.
C2: I:=TIMELIMITMAX*60+CLOCK+P(RTR);
  IF NOT(M[A5] AND R4) = NOT(0) THEN
  COMPLEXSLEEP((T:=(CLOCK+P(RTR)) GTR 1) OR % TIMLIMIT
    (NOT(M[A5] AND R4) # NOT(0))); % CONDITION MET
  IF T=1 THEN BEGIN TERMINATE(P1MIX); TERMINALMESSAGE(89); END;
  GO TO RETURN;
C5: P(,COM5); GO TO DIFFCOM;
C10: P(,ASR); GO TO DIFFCOM;
C11: % (SHM)19516000
  IF R4=0 THEN FILEOPEN(0,A5,[CF]); % (SHM)19516050
  IF R4=6 THEN % (SHM)19516100
  BEGIN FILECLOSE(NFLAG(A5)); GO TO INITIATE END; % (SHM)19516150
  IF R4=4 THEN % (SHM)19516200
  BEGIN % (SHM)19516250
  IF A5[4] THEN % FILE IS IN DIRECTORY % (SHM)19516300
  FORGETSPACE(DIRECTORYSEARCH(R8,R7,=(A5,[CF])&R6[CTF])) ELSE 19516350
  BEGIN % (SHM)19516400
  IF (T:=R9,[18:5]) GTR 0 THEN % EU SPECIFIED % (SHM)19516410
  T:=(IF T GTR 20 THEN 0 ELSE -T) ELSE % (SHM)19516420
  IF (T:=R9,[16:2]) GTR 0 THEN % SPEED SPECIFIED % (SHM)19516430
  T:=(IF T GTR 2 THEN 0 ELSE T) ELSE % (SHM)19516440
  T:=0; % NO SPEED OR EU SPECIFIED % (SHM)19516450
  A5[R6]:=PETUSERDISK(A5[8],T); % (SHM)19516460
  END; % (SHM)19516470
  GO TO INITIATE; % (SHM)19516500
  END; % (SHM)19516600
  P(,COM11); GO TO DIFFCOM; % (SHM)19516700
C13: P(,COM13); GO TO DIFFCOM; 19517000
C14: IF NOT R4,[2:1] THEN MAKEPRESENT([R4] INX 0); 19518000
  M[R4 INX NOT 1] + (*P(DUP))&(NOT P(DUP))[2:2:1]; 19519000
  GO TO INITIATE; 19520000
C19: P(,COM19); GO TO DIFFCOM; 19523000
C23: P(,COM23); GO TO DIFFCOM; 19524000
C24: T + A4[R5]; A4[R5] + 0; 19525000
$ SET OMIT = SHAREDISK 19525999
  FORGETUSERDISK(T,A4[8]); 19526000
$ POP OMIT 19526001
$ SET OMIT = NOT(SHAREDISK) 19526499
  FORGETUSERDISK(T,A4[8]&(NOT A4[4])[1:47:1]); 19526500
$ POP OMIT 19526501

```

GO TO INITIATE;	19527000
C34:: IF (T+R4) > 0 THEN STREAM(R4,T+T+SPACE(17));	19528000
BEGIN SI+R4; DS+17 WDS END;	19529000
TERMINATE(P1MIX); TERMINALMESSAGE(=T);	19530000
C35:: IF R4,[FF] LEQ 33 THEN P(,LIBRARYDUMP)	19530500
ELSE IF R4,[FF] LEQ 35 THEN P(,LIBRARYLOAD)	19530505
ELSE P(,LIBRARYZERO);	19530510
T+P(DUP,LOD,RFB);T+T+SPACE(T,[8:10])[CTC];	19530520
\$ SET OMIT = NOT(AUXMEM)	19530522
IF T,[6:1] THEN % STORED ON AUXMEM	19530524
DISKWAIT(=T,[CF],T,[8:10],=(O&T[32:21:12]))	19530526
ELSE	19530528
\$ POP OMIT	19530529
DISKWAIT(=T,[CF],T,[8:10],T,[FF]+MCPBASE);	19530540
M[T1],[CF];=T;	19530560
GO TO DIFFCOM;	19530600
C36: IF (M[A5] AND R4)=0 THEN	19532000
BEGIN	19532100
T := M[M[A5] INX	19532300
(IF P(DUP),[22:1] THEN 2 ELSE NOT 1)),[12:6];	19532400
WAITORSWAP(T, A5,[CF]);	19532500
END;	19532600
GO TO RETURN;	19532700
C37: NEXT1[P1MIX] := R5; % SHM	19532800
NEXT2[P1MIX];=R4;	19532900
GO TO INITIATE;	19532910
C40:: IF R5,[8:10]=1023 THEN	19532930
BEGIN M[R5],[CF];=PRNTABLE[R5,[FF]];GO INITIATE;END ELSE	19532935
IF R5,[CF]=0 THEN	19532940
BEGIN LINKUP(R6,R5:=R5,[FF]);	19532945
SLEEP([M[R5]],@1000000000000000); GO RETURN;	19532950
END ELSE	19532955
IF R5,[15:15]=0 THEN	19532960
BEGIN M[R5];=NOT 0; GO TO INITIATE; END ELSE	19532962
IF R5,[FF]=@77777 THEN BEGIN M[R5];=MOD3IOS;GO INITIATE;END ELSE	19532964
DKBUSINESS(R5);	19532966
GO RETURN;	19532968
C41:: IOREQUEST(R7,R6,FLAG(R5)); GO INITIATE;	19532970
CM2: P(,COMM2); GO TO DIFFCOM;	19533000
CM1: P(,COMM1); GO TO DIFFCOM;	19534000
CM18: GO TO RETURN; %INVALID COMMUNICATE	19539300
CM4: FORGETESPDISK(R5);	19540000
GO TO RETURN;	19541000
CM3: R4+GETESPDISK;	19542000
GO TO RETURN;	19543000
CM5: P(,COMM5);	19544000
GO TO DIFFCOM;	19544100
CM6: R4+GETUSERDISK(R7);	19545000
GO TO RETURN;	19546000
CM7: FORGETUSERDISK(R6,R5);	19547000
GO TO RETURN;	19548000
CM8: DISKWAIT((A8,[CF])&R9[1:47:1],R6,R5);	19549000
M[A8,[CF]-1]+0;	19549500
GO TO RETURN;	19550000
CM9: P(,COMM9); GO TO DIFFCOM;	19551000
CM10:USERCODE[P1MIX]+R5;	19552000
GO TO RETURN;	19553000
CM12:LOGLINE+R8;	19554000
IF FALSE THEN	19554800
CM11:IF NOT DAT[P1MIX],NDSABLE THEN R5:=R5 OR MEMORY;	19554900

```

        TWXOUT(A7,[CF],R6,R5,LOGLINE);
        R5:=BREAK[LOGLINE,[40:8]];
        GO TO RETURN;
CM13:P(,COMM13);
        GO TO DIFFCOM;
CM14:IF (R6:=R6,[40:8]) LEQ LMAX THEN
        IF SCHEDULE[R6] THEN GO TO RETURN;
        SEQARRAY[R6]:=R5;
        STREAM(A:=R5,[21:27],T:=T:=GETAREA(0));
        BEGIN S1:=LOC A; DS:=8 DEC; DS:=LIT LEFTARROW;
              D1:=DI-9; DS:=8 FILL; A:=DI;
              D1:=T; S1:=A; DS:=9 CHR;
        END;
        TWXOUT(T,8,0&1[CTF]&1[2:47:1],R6);
        FORGETAREA(0,T);
        GO TO RETURN;
CM19:R5:=GETSPACE(30,64,5)+2;
        IF NOT A6,[2:1] THEN MAKEPRESENT([R7] INX 1);
        MOVE(30,A6,R5);
        R6+6;
CM15:P(,COMM15);
        GO TO DIFFCOM;
CM16:
        GO TO RETURN;
CM17:P(,COMM17);
        GO TO DIFFCOM;
CM20:IF (T:=DIRECTORYSEARCH(R6,USERCODE[P1MIX],8),[CF]) LSS 64
        THEN GO RETURN;
        M[T+2]:=USERCODE[P1MIX]; M[T+5]:=R5; M[T+6]:=R7;
        M[T+27]:=R8;
        R5:=GETSPACE(30,64,5)+2; %GET SPACE BELOW THE FENCE
        MOVE(30,T,R5);
        FORK(P(,SCHEDULE),R5&1[2:47:1],0,160,0);
        GO TO RETURN;
CM21:IF (T:=DIRECTORYSEARCH(R7,R6,5)) NEQ 0 THEN
        BEGIN
        DISKWAIT(-(A9,[CF]), 30, T.[FF]);
        M[A9,[CF]-1]:=0;
        FORGETSPACE(T);
        END ELSE M[A9,[CF]]:= -1;
        GO TO RETURN;
END SHORTIES;
PROCEDURE FRONTEND(MIX);
        VALUE MIX;
        REAL MIX;
        FORWARD;
% THE FORMAT OF SEGMENT ZERO OF PROGRAMS%
% S[0] = LOCATION OF SEGMENT DICTIONARY%
% S[1] = SIZE OF SEGMENT DICTIONARY%
% S[2] = LOCATION OF PRT%
% S[3] = SIZE OF PRT%
% S[4] = LOCATION OF FILE PARAMETER BLOCK%
% S[5] = SIZE OF FILE PARAMETER BLOCK%
% S[6],[1:1] = 1 FOR NEW FORMAT SEGMENT 0, ELSE 0
% S[6] = STARTING SEGMENT NUMBER%
% S[7],[2:1] = FORTRAN FAULT FLAG
% S[7],[33:15] = NUMBER OF FILES%
% S[7],[18:15] = CORE REQUIREMENT / 64%
% IF S[2] < 0 THEN THE JOB WAS COMPILED BY COBOL%
% S[15] = DISK ADDRESS OF LABEL EQUATION ENTRIES

```

```

19555000
19555100
19556000
19557000
19558000
19559000
19559020
19559050
%11019559100
%11019559120
%11019559140
%11019559160
%11019559180
19559200
%11019559220
19560000
19560100
19560200
19560300
19560400
19561000
19562000
19563000
19564000
19565000
19566000
19567000
19568000
19569000
19569100
19570000
19571000
19572000
19573000
19574000
19574100
19574200
19574300
19574400
19574500
19574600
19900000
%R7519991000
%R7519991100
%R7519991200
%R7519991300
20000000
20001000
20002000
20003000
20004000
20005000
20006000
20006500
20007000
20007100
20008000
20009000
20010000
20010100

```



```

%          PRESENTED WHEN PROGRAM WAS COMPILED AND      20010200
%          APPLICABLE TO ALL EXECUTIONS                 20010300
%          S[16]          = ESTIMATED PROCESSOR TIME    (FROM COMPILATN)20010400
%          S[17]          = ESTIMATED I/O TIME         (FROM COMPILATN)20010500
%          S[18]          = PRIORITY                   (FROM COMPILATN)20010600
%          S[19]          = COMMON VALUE              (FROM COMPILATN)20010700
%          S[20]          = ESTIMATED CORE REQUIREMENTS(FROM COMPILATN)20010800
%          S[21]          = STACK SIZE                 (FROM COMPILATN)20010900

```

```

PROCEDURE SELECTRUN1;
  BEGIN

```

```

    REAL   MSCW          = -2,          20011000
           F             = -1,          20011100
           MYMSCW        = -1,          20011200
           RCW           = +0,          20011300
           I             = +1,          20011400
           T             = +2,          20011500
           L             = +3,          20011600
           DT            = +4,          20011700
           MIX           = +5,          20011800
           HDR           = +6,          20011900
           LEVEL         = +7,          20012000
           MCPJOB        = +8,          20012100
           OLAYDISK      = +9,          20012200
           THISLINK      = +10,         20012300
           NEXTLINK      = +11,         20012400
           PREVLINK      = +12,         20012500
           TYPE          = +13,         20012600
           STACKLOC      = +14,         20012700
           SHEETLOCKED   = +15;        20012800

```

```

    ARRAY  S             = +16[*],      20012900
           SEGO          = +17[*],      20013000
           TRP           = +18[*],      20013100
           LBL           = +19[*],      20013200
           SD            = NT2[*],      20013300
           TSKA          = NT2[*];      20013400

```

```

    REAL  BELOW         = LBL + 1,      20013500
           SWAPDISK      = BELOW + 1,    20013600
           SWAPDISKSIZE = SWAPDISK + 1,  20013700
           UVSPACE       = SWAPDISKSIZE + 1, 20013800
           SVALUE        = UVSPACE,      20013900
           RETURNMSCW    = UVSPACE + 1,   20014000
           RETURNRCW     = RETURNMSCW + 1; 20014100

```

```

    ****          ***NOTE****          20014200
    **** THE VARIABLES DECLARED ABOVE MUST CORRESPOND EXACTLY TO 20014300
    **** THOSE DECLARED IN PROCEDURE SELECTRUN.                   20014400

```

```

    REAL  EUVAL         = RETURNRCW + 1,  20014500
           FBADRS       = EUVAL + 1,     20014600
           FPBVERSION   = FBADRS + 1,    20014700
           FT           = FPBVERSION+ 1,  20014800
           LINDX        = FT + 1,        20014900
           LINK         = LINDX + 1,     20015000
           SENSEVAL     = LINK + 1,      20015100
           SPDVAL       = SENSEVAL + 1,   20015200

```

```

S2          = SPDVAL      + 1#
FB          = S2          + 1#
FPB        = FB          + 1#
REAL FT1    = NT1#
          TYPEDISK      = NT3#

COMMENT THE VALUE OF "TYPE" DETERMINES WHICH PORTIONS OF
THIS PROCEDURE WILL BE EXECUTED. THIS PROCEDURE CAN ALSO
DETERMINE WHICH PORTIONS OF PROCEDURE "SELECTRUN" WILL BE
EXECUTED BY ASSIGNING A NEGATIVE VALUE TO "TYPE" BEFORE
RETURNING TO THAT PROCEDURE.
END OF COMMENT#

DEFINE STARTING      = 1#;
CONTINUEING        = 2#;
QUITTING           = 3#;
RUNING             = 4#;
PASSING            = 5#;
EQUATING           = 6#;

DEFINE XCLOCKTIME =
((NT2=(XCLOCK DIV 3600)) MOD 60 + (NT2 DIV 60)*100 +
0,5 ) DIV 1)#;

DEFINE ACTUALDISKADDRESS(ACTUALDISKADDRESS1) =
((JAR[MIX,((NT4:=ACTUALDISKADDRESS1) DIV (NT3:=JAR[MIX,8]))+10]
+ (NT4 MOD NT3) + 0,5) DIV 1)#;

$ SET OMIT = NOT(PACKETS)
DEFINE UNITNO = S[23],[2:6]#; % ORIGINATING UNIT
$ POP OMIT

LABEL CONTINUE, DLX, EXIT, LEM, RMSG, UNBLK, STOP;

SUBROUTINE DELINK;
% DELINKS THE SHEET ENTRY AND RETURNS SHEET DISK SPACE
BEGIN
STREAM(A:=S[3],[8:10],B:=P(,SCHEDULEIDS));
BEGIN % MARK SCHEDULE SLOT "OPEN"
SKIP A DB; DS:=RESET;
END;
IF F = 0 THEN % SHEET ENTRY NOT PASSED AS PARAMETER
BEGIN
IF NEXTLINK=0 THEN SHEET[LEVEL],[FF]:=PREVLINK;
IF PREVLINK=0 THEN
BEGIN
SHEET[LEVEL],[CF]:=NEXTLINK; GO DLX;
END;
LBL:=[M[SPACE(30)]]&30[8:38:10];
DISKWAIT(=(LBL INX 0), 30, PREVLINK);
LBL[29]:=NEXTLINK;
DISKWAIT( (LBL INX 0), 30, PREVLINK);
FORGETSPDISK(THISLINK);
IF LBL NEQ 0 THEN FORGETSPACE(LBL); LBL:=0;
END; % IF SHEET ENTRY NOT A PARAMETER
END DELINK;

P(MYMSQW, STF);

```

```

20017400
20017500
20017600
20017700
20017800
20017900
20018000
20018100
20018200
20018300
20018400
20018500
20018600
20018700
20018800
20018900
20019000
20019100
20019200
20019300
20019400
20019500
20019600
20019700
20019800
20019900
20020000
20020100
20020110
20020119
20020120
20020121
20020200
20020300
20020400
20020500
20020600
20020700
20020800
20020900
20021000
20021100
20021200
20021300
20021400
20021500
20021600
20021700
20021800
20021900
20022000
20022100
20022200
20022300
20022310
20022400
20022500
20022600
20022700
20022800

```

```

P(0, 0, 0, 0, 0, 0, 0, 0, 0, 0); % FOR VARIABLES LOCAL TO THIS
% PROCEDURE ONLY
IF TYPE=CONTINUEING THEN GO TO CONTINUE;
IF TYPE=STARTING THEN % SEARCH THE SHEET QUEUE TO FIND A CANDIDATE
% FOR SELECTION
BEGIN
FOR LEVEL:=0 STEP 1 UNTIL SHEETMAX DV % FOR ALL "SHEET PRIORITIES"
BEGIN
PREVLINK:=NEXTLINK:=0; % RESET FOR EACH "LEVEL"
% IF THERE IS AN ENTRY IN THE SHEET, SEE IF IT WILL FIT
IF(THISLINK:=SHEET[LEVEL],[CF]) NEQ 0 THEN GO TO LEM;
CONTINUE;
% "NEXTLINK" OBTAINED FROM "SHEET[29]" BELOW
% IF THERE IS ANOTHER ENTRY AT THIS LEVEL, PROCESS IT NOW
IF(THISLINK:=NEXTLINK) NEQ 0 THEN GO TO LEM;
END;
TYPE := -QUITTING; % END OF SHEET SEARCH
GO TO EXIT;
LEM:
% AT THIS POINT, THERE IS A CANDIDATE FOR SELECTION
IF S = 0 THEN % NO SHEET SPACE OBTAINED YET
BEGIN
S := [M[GETSPACE(31,2,0)+2]]&30[B:38:10];
END;
% . . . . .
% READ SHEET ENTRY INTO CORE AT "S"
% . . . . .
DISKWAIT(=(S INX 0), 30, THISLINK);
NEXTLINK:=S[29]; % NEXT ENTRY IN SHEET QUEUE AT THIS LEVEL
% ***** * * ***** * * ***** *****
% * * * * * * * * * * * * * * * * * * * *
% *** * * ***** ***** * * *****
% * * * * * * * * * * * * * * * * * * * *
% * *** ***** ***** * * * * * * * 0
HDR := GETSPACE(30,0,0)+2;
DISKWAIT(=HDR, 30, S[25]); % READ FILE HEADER INTO CORE AT "HDR"
GO TO EXIT;
END; % IF TYPE = STARTING OR CONTINUEING
IF TYPE=PASSING THEN % PASS THIS ENTRY WITHOUT DELINKING
BEGIN
% ***** ***** ***** *****
% * * * * * * * * * *
% ***** ***** ***** *****
% * * * * * * * * * *
% * * * ***** *****
IF S[3] GTR 0 % S[3],[1:1]=0 FIRST TIME THROUGH
$ SET OMIT = PACKETS
AND SCHEDMSG

```

```

20022900
20023000
20023100
20023200
20023300
20023400
20023500
20024600
20024700
20024800
20024900
20025000
20025100
20025200
20025300
20025400
20025500
20025600
20025700
20025800
20025900
20026000
20026100
20026200
20026300
20026400
20026500
20026600
20026700
20026800
20026900
20027000
20027100
20027200
20027300
20027400
20027500
20027600
20027700
20027800
20027900
20028000
20028100
20028700
20028900
20029000
20029100
20029200
20029300
20029400
20029500
20029600
20029700
20029800
20029900
20030000
20030100
20030200
20030249
20030250

```

```

$ POP OMIT
THEN BEGIN
  S[3]:=NABS(S[3]); % MARK IT SCHEDULED
  IF F=0 THEN % SHEET ENTRY NOT PASSED AS PARAMETER
  % WRITE THE SHEET ENTRY BACK OUT WITH S[3] "MARKED"
  DISKWAIT((S INX 0), 30, THISLINK);
  STREAM(C:=LEVEL, A:=S[*], ID:=S[3],[8:10],
    Q:=XCLOCKTIME, B:=HDR);
  BEGIN
    S1:=LOC C; DS1:=6DEC; DI:=DI-6; DS1:=5FILL; % PRIORITY
    DI:=B; DI:=DI+6; DS1:=LIT"1";
    S1:=A; S1:=S1+1; DS1:=7CHR; % MFID
    S1:=S1+1; DS1:=LIT"/"; DS1:=7CHR; % FID
    DS1:=LIT"="; S1:=LOC ID; DS1:=2DEC; % SCH.NO.
    DS1:=11 LIT" SCHEDULED "; S1:=LOC Q; DS1:=4DEC; % TIME
    DS1:=LIT"=";
  END STREAM;
  SPOUTER(HDR,UNITNO,SCHEDMSG);
  END % IF SCHEDMSG AND FIRST TIME THROUGH
ELSE FORGETSPACE(HDR);
IF F NEQ 0 THEN % SHEET ENTRY PASSED AS A PARAMETER
BEGIN
  DISKWAIT(F,[CF],30,T:=GETESPDISK); % WRITE SHEET ENTRY TO DISK
  FORGETSPACE(S[7]); % CORE ADDRESS OF SEGMENT ZERO IN S[7]
  IF NOT SHEETLOCKED THEN
  BEGIN
    SLEEP(ITOGLE,SHEETMASK);
    LOCKTOG(SHEFTMASK);
    SHEETLOCKED := 1;
  END;
  IF (L:=S[2],[CF]) GTR SHEETMAX THEN L:=SHEETMAX;
  % SHEET[2],[CF] = "SHEET" PRIORITY
  IF SHEET[L],[CF] NEQ 0 THEN % SHEET QUEUE ALREADY EXISTS
  BEGIN % LINK IN THIS ENTRY
    DISKWAIT(*F,[CF],30,1:=SHEET[L],[FF]); % TAIL OF QUEUE
    S[29]:=T; % LINK TO THIS ENTRY
    DISKWAIT(F,[CF],30,1); % REPLACE ENTRY
  END
  ELSE SHEET[L]:=T; % ESTABLISH NEW SHEET QUEUE
  SHEET[L],[FF]:=T; % LINK IN AT END OF QUEUE
  TYPE := "QUITTING";
  GO TO EXIT; % DONT PROCESS ANY MORE ENTRIES
  END;
PREVLINK:=THISLINK;
IF MIX LEQ MIXMAX THEN
BEGIN
  TYPE := "CONTINUEING";
  GO TO CONTINUE;
  END
ELSE
BEGIN
  TYPE := "QUITTING";
  GO TO EXIT;
  END;
END; % IF TYPE = PASSING

IF TYPE = EQUATING THEN
BEGIN

```

```

20030251
20030300
20030400
20030500
20030600
20030700
20030800
20031500
20031600
20031700
20031800
20031900
20032000
20032100
20032200
20032900
20033000
20034000
20034100
20034200
20034300
20034400
20034500
20034600
20034700
20034800
20034900
20035000
20035100
20035200
20035300
20035400
20035500
20035600
20035700
20035800
20035900
20036000
20036100
20036200
20036300
20036400
20036500
20037100
20037200
20037300
20037400
20037500
20037600
20037700
20037800
20037900
20038000
20038100
20038200
20038300
20048000
20048100
20048200
20048300

```

```

% ***** ***** ***** 20048400
% * * * * * 20048500
% **** ***** ***** 20048600
% * * * * * 20048700
% * 0 * 0 ***** 0 20048800
% 20048900
FPB:=GETSPACE(SEG0[5] INX 1,0,0)+2; 20049000
% SEG0[5] = SIZE OF THE FILE PARAMETER BLOCK ON DISK 20049100
% SEG0[4] = RELATIVE DISK ADDRESS OF THE FILE PARAMETER BLOCK 20049200
% SEG0[7] = NUMBER OF FILES IN THE F,P,B. 20049300
% ETRLNG = NUMBER OF WORDS PER FILE USED IN THE F,P,B. 20049400
M[SEG0[5] INX FPB]:=0; % SET TO ZERO TO INSURE THAT STREAM STATEMENT 20049500
% USED TO BUILD "IN-CORE" FPB WILL NOT SKAN 20049600
% PAST THE END OF THE COMPILER GENERATED FPB. 20049700
FBI:=GETSPACE(SEG0[7],[CF]*ETRLNG,0,1)+2; 20049800
% "FB" WILL BE "IN-CORE" FILE PARAMETER BLOCK LOCATION 20049900
DISKWAIT(=FPB, SEG0[5] INX 0, ACTUALDISKADDRESS(SEG0[4],[CF])); 20050000
% 20050100
COMMENT FORMAT OF COMPILER GENERATED FPB: 20050200
CHRS 1 AND 2 = FILE NUMBER (12 BIT BINARY) STARTING WITH 1 20050300
CHR, 3 = FILE TYPE 20050400
CHRS 4 THRU 10 = MFID 20050500
CHRS 11 THRU 17 = FID 20050600
CHR 18 = LENGTH OF INTERNAL FILE NAME (6 BIT BINARY) 20050700
CHRS 19 THRU N = INTERNAL NAME 20050800
FOR VERSION 1 ( VERSION NUMBER IN SEG0[5],[1:8] ) 20050900
NEXT TWO CHARACTERS FOLLOWING INTERNAL NAME CONTAIN: 20051000
[40:1] = SENSITIVE BIT 20051100
[41:2] = DISK SPEED (1=FAST, 2=SLOW, 0=USPECIFIED) 20051200
[43:5] = EU NUMBER + 1 20051300
% 20051400
COMMENT FORMAT OF "IN-CORE" FPB ( 5 WORDS FOR EACH FILE ENTRY ) 20051500
WORD[0],[ 6:42] = MFID 20051600
WORD[1],[ 6:42] = FID 20051700
WORD[2],[ 1:17] = REEL NUMBER (3 BCL DIGITS) 20051800
WORD[2],[18:30] = CREATION DATE (5 BCL DIGITS) 20051900
WORD[3],[ 1:5 ] = CYCLE NUMBER (BINARY) 20052000
WORD[3],[ 6:17] = PRN (PHYSICAL REEL NUMBER) FOR NON-DISK FILES 20052100
WORD[3],[15:1 ] = SENSITIVE BIT (DISK FILES ONLY) 20052200
WORD[3],[16:2 ] = DISK SPEED (DISK FILES ONLY) 20052300
WORD[3],[18:5 ] = EU, NUMBER+1 (DISK FILES ONLY) 20052400
WORD[3],[23:1 ] = IO CODE (INPUT=0,OUTPUT=1) 20052500
WORD[3],[24:12] = NUMBER OF ERRORS 20052600
WORD[3],[36:6 ] = LOGICAL UNIT NUMBER + 1 20052700
WORD[3],[43:5 ] = UNIT TYPE 20052800
END OF COMMENT; 20052900
% 20053000
FPBVERSION:=SEG0[5],[1:8]; % NEWER VRSN,CONTAINS EU,SPD,ETC. 20053100
STREAM(TOG:=(FPBVERSION=1),T1:=0,T2:=0,C:=0,FPB,FB); 20053200
BEGIN 20053300
SI:=FPB; 20053400
LL: IF SC="0" THEN % FIRST DIGIT OF FILE NUMBER 20053500
BEGIN 20053600
SI:=SI+1; IF SC="0" THEN GO TO L2; % END OF FPB 20053700
END ELSE SI:=SI+1; 20053800
SI:=SI+1; T1:=SI; SI:=SI+1; % FILE TYPE LOCATION 20053900
2(DS:=LIT"0"; DS:=7CHR); % MFID,FID 20054000
T2:=DI; DI:=LOC C; DI:=DI+7; DS:=CHR; DI:=T2; %INT,NAME SIZE 20054100
DS:=15LIT"0"; % ZERO OUT REEL,DATE,CYCLE,ETC. 20054200
T2:=SI; SI:=T1; DS:=CHR; SI:=T2; % FILE TYPE 20054300

```

GO TO SK; L1: GO TO LL; L2: GO TO XXIT; SK;	20054400
SI:=SI+C; % SKIP OVER INTERNAL NAME	20054500
TOG(T2:=D1; D1:=D1-6; SKIP 3DB; SKIP 4SB;	20054600
IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB; % SENSITIVE	20054700
2(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB); % SPEED	20054800
5(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB); % EU	20054900
D1:=T2);	20055000
DS:=BLIT*0"; % ZERO OUT FIFTH WORD OF FB	20055100
GO TO L1;	20055200
XXIT: END STREAM STATEMENT;	20055300
IF MCPJOB THEN GO TO STOP; % NO LABEL EQUATION FOR "SYSTEM" JOBS	20055400
*** LABEL EQUATION PROCESSING	20055500
COMMENT LABEL EQUATION RECORD FORMAT;	20055600
WORD[ 0] = MFID ( ZERO, IF NONE GIVEN )	20055700
WORD[ 1] = FID	20055800
WORD[ 2 ],[ 1:17] = REEL NUMBER ( 3 BCL DIGITS )	20055900
.[ 18:30] = CREATION DATE ( 5 BCL DIGITS )	20056000
.[ 42:1 ] = MARKER FOR FILE OPEN ( 1 = CDATE GIVEN )	20056100
WORD[ 3 ],[ 1:5 ] = CYCLE NUMBER	20056200
.[ 15:8 ] = NUMBER OF COPIES *1	20056300
.[ 23:1 ] = PACKETS	20056400
.[ 42:1 ] = "FORMS" REQUESTED	20056500
.[ 43:5 ] = UNIT TYPE	20056600
WORD[ 4 ],[ 0:6 ] = SIZE OF INTERNAL NAME	20056700
.[ 6:42] = FIRST SEVEN CHARACTERS OF INTERNAL NAME	20056800
WORD[ 5] THROUGH WORD[11] = REMAINDER OF INTERNAL NAME	20056900
WORD[12],[ 15:1 ] = SENSITIVE BIT	20057000
.[ 16:2 ] = DISK SPEED (1=FAST,2=SLOW,0=NOT SPECIFIED)	20057100
.[ 18:5 ] = EU NUMBER + 1	20057200
WORD[14] = START OF NEXT LBL,EQN,ENTRY (14 IF NO MORE ENTRIES)	20057300
WORD[29] = LINK TO NEXT ESP SEGMENT FOR LABEL EQUATION	20057400
END OF COMMENT;	20057500
FOR L := 1 STEP 1 UNTIL 2 DO	20057600
BEGIN	20057700
LINK:=IF L THEN S[15] ELSE S[13]; % EQN FROM COMPILE/EXEC.	20057800
% S[15] = RELATIVE DISK ADDRESS IN CODE FILE FOR LABEL	20057900
% EQUATION ENTERED AT COMPILE TIME	20058000
% S[13] = ACTUAL ESP DISK ADDRESS OF LABEL EQUATION ENTERED	20058100
% AT RUN TIME,	20058200
S2 := NOT L; % TRUE, IF LBL,EQN,ENTERED AT RUN TIME	20058300
WHILE LINK NEQ 0 DO % IF LBL,EQN,EXISTS	20058400
BEGIN	20058500
LBL:=[M[GETSPACE(30,0,0)+2]]&30[8:38:10];	20058600
% IF LINK=S[15],READ FROM CODE FILE ELSE READ FROM ESP DISK	20058700
DISKWAIT(=(LBL INX 0), 30,	20058800
IF L THEN ACTUALDISKADDRESS(LINK) ELSE LINK);	20058900
I := 0; % START AT BEGINNING OF SEGMENT	20059000
IF NOT L THEN FORGETESPDISK(LINK);	20059100
LINK := LBL[29]; % NEXT LINK	20059200
IF LBL[0] = 14 THEN GO TO STOP;	20059300
LINDX:=I*14; % INDEX INTO LABEL EQUATION SEGMENT	20059400
STREAM(FN:=0 ; FT:=[FT], ZERO:=0, T2:=0,	20059500
TOG:=(FPBVERSION=1), FPB, FI:=[LBL[LINDX+4]], CI:=0);	20059600
BEGIN	20059700
SI := F; DI:=LOC C; DI:=DI+7 ; DS:=CHR; % LBL,NAM,SIZE	20060800
	20060900
	20061000
	20061100
	20061200
	20061300

SI := FPB;	20061400
L: DI:=LOC FN; DI:=DI+6; DS:=2 CHR; % FILE NUMBER	20061500
DI := LOC ZERO; SI:=SI-2;	20061600
IF 2 SC = DC THEN GO TO XXIT; % FILE NUMBER=0	20061700
DI:=FT; DS:=CHR; SI:=SI+14; % SAVE FILE TYPE FOR CHK BELOW	20061800
DI := F; % SI AT FPB INT,NAM,DI AT LBL,EQN,	20061900
IF SC = DC THEN % SAME STRING SIZE	20062000
BEGIN	20062100
IF C SC=DC THEN GO TO XXIT; % ALL CHARACTERS MATCH	20062200
END	20062300
ELSE	20062400
BEGIN % NOT THE SAME SIZE	20062500
SI:=SI-1; DI:=LOC T2; DI:=DI+7; DS:=CHR;	20062600
SI:=SI+T2; % SKIP OVER FPB ENTRY	20062700
END;	20062800
TOG(SI:=SI+2); % SPEED AND EU CHARACTERS IN FPB(VERSION 1)	20062900
GO TO L;	20063000
XXIT: END;	20063100
	20063200
IF (T:=P) NEQ 0 THEN % VALID LABEL EQUATION	20063300
BEGIN	20063400
FBADRS:=(T-1)*ETRLNG+FB; % ADRS OF FB FILE ENTRY	20063500
% FT IS FILE TYPE FROM FPB OBTAINED ABOVE	20063600
IF (FT1:=LBL[LINDX+3],[43:5]) NEQ @37 THEN FT:=FT1;%NEW TYP	20063700
FT:=FT,[43:5]; % REMOVE "FORMS" BIT	20063800
TYPEDISK = (FT=10) OR (FT=12) OR (FT=13) OR (FT=26);	20063900
STREAM(X:=LBL[LINDX],TOG:=(TYPEDISK AND (FPBVERSION=1)),	20064000
FBADRS);	20064100
BEGIN	20064200
SI:=X; DS:=3WDS; DS:=CHR; % MFID,FID,REEL,DATE,CYCLE	20064300
TOG(SI:=SI+2; SKIP 5SB; DI:=DI+2; SKIP 5DB;	20064400
IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB;	20064500
JUMP OUT TO L); % SAVE EU/SPEED SPECS FOR DISK	20064600
DS:=3CHR;	20064700
L: DS:=3CHR;	20064800
IF SC NEQ "+" THEN % NEW TYPE SPECIFIED	20064900
IF SC NEQ "" THEN DS:=CHR ELSE DS:=SET;	20065000
END STREAM STATEMENT;	20065100
SENSEVAL := (EUVAL := LBL[LINDX+12],[15:8]),[40:1];	20065200
SPDVAL := EUVAL,[41:2];	20065300
EUVAL := EUVAL AND @37;	20065400
IF SPDVAL GTR 0 THEN	20065500
M[FBADRS+3]:=(*P(DUP))&SPDVAL[16:46:2];	20065600
IF SENSEVAL THEN % FILE SENSITIVE	20065700
M[FBADRS+3]:=(*P(DUP))&SENSEVAL[15:47:1];	20065800
IF EUVAL GTR 0 THEN % NEW EU NUMBER REQUESTED IN LBL,EQN,	20065900
M[FBADRS+3]:=(*P(DUP))&EUVAL [18:43:5];	20066000
END; % IF VALID LABEL EQUATION	20066100
IF (I:=I+1) = 1 THEN IF LBL[14] NEQ 14 THEN GO TO UNBLK;	20066200
END; % WHILE LINK NEQ 0	20066300
STOP: END; % FOR L	20066400
FORGETSPACE(FPB);	20066500
IF LBL NEQ 0 THEN	20066510
BEGIN	20066515
FORGETSPACE(LBL); LBL:=0;	20066520
END;	20066525
TRP[3] := [M[FB]] & (SEGO[7],[CF]*ETRLNG)[8:38:10];	20066600
END; % IF TYPE = EQUATING	20066700
	20066800
EXIT:	20080000

```
P([RETURNRCW], STS, 0, RDS, 0, XCH, P&P[CTF], SIF);  
END PROCEDURE SELECTRUN1;
```

```
PROCEDURE SELECTRUN2;
```

```
BEGIN
```

```
REAL MSCW = -2,  
F = -1,  
MYMSCW = -1,  
RCW = +0,  
I = +1,  
T = +2,  
L = +3,  
DT = +4,  
MIX = +5,  
HDR = +6,  
LEVEL = +7,  
MCPJOB = +8,  
OLAYDISK = +9,  
THISLINK = +10,  
NEXTLINK = +11,  
PREVLINK = +12,  
TYPE = +13,  
STACKLOC = +14,  
SHEETLOCKED = +15;
```

```
ARRAY S = +16[*],  
SEGO = +17[*],  
TRP = +18[*],  
LBL = +19[*],  
SD = NT2[*],  
TSKA = NT2[*];
```

```
REAL BELOW = LBL + 1,  
SWAPDISK = BELOW + 1,  
SWAPDISKSIZE = SWAPDISK + 1,  
UVSPACE = SWAPDISKSIZE + 1,  
SVALUE = UVSPACE,  
RETURNMSCW = UVSPACE + 1,  
RETURNRCW = RETURNMSCW + 1;
```

```
***  
***NOTE***  
*** THE VARIABLES DECLARED ABOVE MUST CORRESPOND EXACTLY TO  
*** THOSE DECLARED IN PROCEDURE SELECTRUN.
```

```
LABEL DLX, BMSG, NG, EXIT;
```

```
DEFINE XCLOCKTIME =  
(((NT2*(XCLOCK DIV 3600)) MOD 60 + (NT2 DIV 60)*100 +  
0.5 ) DIV 1);
```

```
DEFINE ACTUALDISKADDRESS(ACTUALDISKADDRESS1) =  
((JAR[MIX], ((NT4:=ACTUALDISKADDRESS1) DIV (NT3:=JAR[MIX,8]))+10)  
+ (NT4 MOD NT3) + 0.5) DIV 1);
```

```
$ SET OMIT = NOT(PACKETS)  
DEFINE UNITNO = S[23],[2:6]; % ORIGINATING UNIT  
$ POP OMIT
```

```
DEFINE DALOCSIZE = 7;
```

```
20080100  
20080200  
20080300  
20080400  
20080500  
20080600  
20080700  
20080800  
20080900  
20081000  
20081100  
20081200  
20081300  
20081400  
20081500  
20081600  
20081700  
20081800  
20081900  
20082000  
20082100  
20082200  
20082300  
20082400  
20082500  
20082600  
20082700  
20082800  
20082900  
20083000  
20083100  
20083200  
20083300  
20084500  
20084600  
20084700  
20084800  
20084900  
20085000  
20085100  
20085300  
20085400  
20085500  
20085600  
20085700  
20085800  
20085900  
20086000  
20086100  
20086200  
20086300  
20086400  
20086500  
20086600  
20086700  
20086799  
20086800  
20086801  
20086810  
20086900
```



```

% VALUES ASSOCIATED WITH "TYPE" :
DEFINE STARTING      = 1#,
CONTINUEING         = 2#,
QUITTING            = 3#,
RUNING              = 4#,
PASSING             = 5#,
EQUATING           = 6#;

SUBROUTINE DELINK;
% DELINKS THE SHEET ENTRY AND RETURNS SHEET DISK SPACE
BEGIN
STREAM(A:=S[3],[8:10],B:=P(.SCHEDULEIDS));
  BEGIN % MARK SCHEDULE SLOT "OPEN"
  SKIP A DB; DS:=RESET;
  END;
IF F = 0 THEN % SHEET ENTRY NOT PASSED AS PARAMETER
BEGIN
IF NEXTLINK=0 THEN SHEET[LEVEL],[FF]:=PREVLINK;
IF PREVLINK=0 THEN
  BEGIN
  SHEET[LEVEL],[CF]:=NEXTLINK; GO DLX;
  END;
LBL:=[M[SPACE(30)]]&30[8:38;10];
DISKWAIT(-(LBL INX 0), 30, PREVLINK);
LBL[29]:=NEXTLINK;
DISKWAIT( (LBL INX 0), 30, PREVLINK);
DLX: FORGETESPDISK(THISLINK);
IF LBL NEQ 0 THEN FORGETSPACE(LBL); LBL:=0;
END; % IF SHEET ENTRY NOT A PARAMETER
END DELINK;

P(MYMSCW, STF);

%   ****   *****   ***   *****   *****   *****   *****
%   *   *   *   *   *   *   *   *   *   *   *   *
%   ****   *   *   *   *   *   *   *   *   *   *   *
%   *   *   *   *   *   *   *   *   *   *   *   *
%   ****   *****   ****   *   *   *   *****   *****   ***** 0

IF LOGLINE LSS 0 THEN % NOT BY CANDE
DISKWAIT(-(T:=SPACE(10)), 10, S[6],[CF]) ELSE
BEGIN % VIA CANDE, BUILD A LOG MESSAGE
IF(T:=S[2],[8:10])=0 OR T=2 OR T=5 THEN
  BEGIN % GO, EXECUTE OR RUN
  STREAM(C:=12, A3:=ABS(S[24]), A1:=S[0], A2:=S[1],
    B := T := SPACE(9));
  BEGIN
  S1:=LOC C; S1:=S1+7; DS:=CHR; % QUESTION MK,
DS:=5LIT"USER="; S1:=S1+1; DS:=7CHR; % USER CODE
DS:=10LIT"; EXECUTE ";
2(S1:=S1+1; DS:=7CHR; DS:=LIT"/"); % JOB NAME
DI:=DI-1; DS:=34LIT" ";
  END;
  END
ELSE
BEGIN % COMPILE PART
STREAM(C:=12, A2:=S[1], A3:=S[24], A1:=S[0], B:=T:=SPACE(9));

```

```

20087400
20087500
20087600
20087700
20087800
20087900
20088000
20088100
20088200
20088300
20088400
20088500
20088600
20088700
20088800
20088900
20089000
20089100
20089200
20089300
20089400
20089500
20089600
20089700
20089800
20089900
20090000
20090100
20090200
20090210
20090300
20090400
20090500
20094500
20094600
20094700
20094800
20094900
20095000
20095100
20095200
20095300
20095500
20095600
20095700
20095800
20095900
20096000
20096100
20096200
20096300
20096400
20096500
20096600
20096700
20096800
20096900
20097000
20097100
20097200

```

```

BEGIN
SI:=LOC C; SI:=SI+7; DS:=CHR;
DS:=7LIT"COMPILE"; DS:=5LIT" ";
2(SI:=SI+1; DS:=7CHR; DS:=LIT"/");
DI:=DI-1; DS:=5LIT" ";
SI:=SI+1; DS:=7CHR;
DS:=14LIT" LIBRARY ";
DS:=18LIT" ";
END;
END;
DISKWAIT(T,[CF],10,JAR[MIX,6]:=S[6]:=GETESPDISK);
END;
MAKELOG(T,[CF]=1, CNTRLCARD);
FORGETSPACE(T);
I:=1;
IF BOJMESS THEN
IF MCPJOB,[1:1] THEN % "SYSTEM" TYPE JOB
IF NOT (AUTOMESS) THEN % SUPPRESS BOJ/EQJ MESSAGE
IF NOT (S[2],[2:1]) THEN % NOT ES=ED
IF S[2],[4:1] THEN % SUPRESS BOJ/EQJ MESSAGE
BEGIN
STREAM(N:=S[0], MIX, T:=T:=GETSPACE(4,0,0)+2);
BEGIN
DS:=6LIT" AUTO=";
SI:=LOC N; SI:=SI+1; DS:=7CHR;
DS:=2LIT" ="; SI:=LOC MIX; DS:=2DEC;
DS:=LIT"←"; DI:=DI-3; DS:=FILL;
END;
SPOUT(T);
I:=0;
END;
STREAM(C:=LOGLINE,[40:8]×(LOGLINE,[33:7] NEQ 0),
PRIORITY:=S[18], DAAT:=DT, DTOG:=NOT(MCPJOB) AND TRUE,
KT:=((NT1:=ABS(S[24])) NEQ S[1] AND NT1 NEQ 0)+2,
A1:=S[0], A2:=S[1], A3:=NT1, MIX,
Q := XCLOCKTIME, SV:=0, B:=T:=SPACE(10));
BEGIN
SI:=LOC C; DS:=4DEC; DS:=LIT" ";
DS:=4DEC; DS:=LIT" ";
DI:=DI-5; DS:=3FILL; DI:=DI-1; DS:=LIT"=";
DI:=B; DS:=8FILL;
DI:=B; DI:=DI+10; SI:=LOC A1;
KT(SI:=SI+1; DS:=7CHR; DS:=LIT"/");
SI:=LOC MIX; DI:=DI-1; DS:=LIT"="; DS:=2DEC;
SV:=DI; DI:=DI-2; DS:=FILL; DI:=SV;
DS:=5LIT" BOJ "; DS:=4DEC;
DTQG(DS:=LIT" "; SI:=LOC DAAT; SI:=SI+2;
3(DS:=2CHR; DS:=LIT"/"); DI:=DI-1); DS:=LIT"←"; % CDATE
END STREAM;
SPOUTER(T,UNJTNO
,((NOT S[0]),[2:1] OR CANDYMESS) AND BOJMESS AND I OR BOJK);
% ***** ***** ***** ***** ***** ***** *****
% * * * * *
% ***** ***** * ** * ***** ***** * *
% * * * * *
% ***** ***** ***** ***** ***** * * *****
IF F NEQ 0 THEN % SHEET ENTRY PASSED AS A PARAMETER

```

```

20097300
20097400
20097500
20097600
20097700
20097800
20097900
20098000
20098100
20098200
20098300
20098400
20098500
20098600
20100600
20100700
20100800
20100900
20101000
20101100
20101200
20101300
20101400
20101500
20101600
20101700
20101800
20101900
20102000
20102100
20102200
20102400
20102500
20102600
20102700
20102800
20102900
20103000
20103100
20103200
20103300
20103400
20103500
20103600
20103700
20103800
20103900
20104000
20104100
20104200
20104300
20108900
20109000
20109100
20109200
20109300
20109400
20109500
20109600
20109700

```

```

BEGIN
  SEGO:=S&S[7][CTC] % SEGMENT ZERO PRESENT AT CORE ADDRESS "S[7]"
END
ELSE
  BEGIN
    SEGO:=[M[SPACE(30)]]&30[8:38:10];
    DISKWAIT(=(SEGO INX 0),30,M[HDR INX 10]); % READ SEGMENT ZERO
  END; % IF SEGMENT ZERO WAS NOT PRESENT
  JAR[MIX,2] := (*P(DUP)) & SEGO[2][1:1:2] &
    SEGO[7][3:2:1];
  % SEGO[2],[1:1] = JOB COMPILED BY COBOL ( NO "OAT" ENTRY )
  % SEGO[2],[2:3], SEGO[7],[2:1] = USED FOR INTER-PROG,COMMUNICATION

% *****
% * * * * *
% *****
% * * * * *
% *****
% * * * * *
% *****

  % S[21] CONTAINS STACK SIZE, SEGO[3] CONTAINS PRT SIZE
  SINFO[MIX],[CF] :=
    (STACKLOC:=GETSPACE(SEGO[3] INX S[21] INX 64, 0, 1))+2;
  % COMPUTE THE ADDRESS FOR THE PRT SUCH THAT PRTADR,[42:6]=0
  T:=(((STACKLOC:=STACKLOC+2)+S[21]) OR 63) + 1; % S[21]=STACKSIZE
  IF ((I:=M[STACKLOC-2],[CF])=(L:=SEGO[3] INX T)) GTR 10 THEN
    BEGIN % RETURN REMAINDER OF PRT SPACE
      WAITSTORE(MIX); STOREDY[MIX]:=0;
      M[L] := I & (STACKLOC-2)[CTF] & MIX[9:42:6]; % NEW LINK
      M[I],[CF] := L; % BACK LINK
      M[STACKLOC-2],[CF] := L; % FORWARD LINK
      STOREDY[MIX] := 1;
      FORGETSPACE(L+2);
    END; % IF PRT SPACE WAS TOO LARGE
  % ZERO OUT STACK TO EASE PROBLEMS OF CONGENITAL DUMP-READERS
  M[STACKLOC] := @3333333333333333;
  MOVE(T=STACKLOC-1,STACKLOC,STACKLOC+1);

% . . . . .
% READ IN PRT FROM DISK
% . . . . .

  DISKWAIT(=T, SEGO[3],[CF], ACTUALDISKADDRESS(SEGO[2],[CF]));
  % SEGO[2] = RELATIVE DISK ADDRESS OF THE PRT IN THE CODE FILE
  % SEGO[3] = SIZE OF THE PRT
  TRP:=PRTR0W[MIX]:=[M[T]]&1023[8:38:10]; % DESCRIPTOR TO THE PRT

% *****
% * * * * *
% *****
% * * * * *
% *****
% * * * * *
% *****

  SEGD[MIX] :=
    TRP[4]:=[M[T:=GETSPACE(SEGO[1],[CF],1,1)+2]];
    DISKWAIT(=T, SEGO[1],[CF], ACTUALDISKADDRESS(SEGO[0],[CF]));
    % SEGO[0]= RELATIVE DISK ADDRESS OF SEGMENT DICTIONARY
    % SEGO[1]= SIZE OF THE SEGMENT DICTIONARY
    M[TRP[4]] := SEGO[1],[CF] -1; % SEGDICT[0]=SIZE OF DICTIONARY
  $ SET OMIT = NOT(AUXMEM)
  IF CODEOLAY THEN % MARK ALL CODE SEGMENTS "TO GO TO AUXMEM"

```

```

20109800
20110000
20110100
20110200
20110300
20110400
20110500
20110600
20110700
20111100
20111200
20111300
20115200
20115300
20115400
20115500
20115600
20115700
20115800
20115900
20116100
20116200
20118800
20118900
20119000
20119100
20119300
20119900
20120000
20120100
20120300
20120800
20120900
20121000
20121100
20121200
20121300
20121400
20121500
20121600
20121700
20121800
20121900
20122000
20122100
20122200
20122300
20122400
20122500
20122600
20122700
20122800
20124400
20124900
20125000
20125100
20125200
20125300
20125400
20125500

```

```

        BEGIN
        SDI=TRP[4]&(I:=SEGO[1],[CFJ])[8:38:10]; % END OF SEG.DICT,
        WHILE(I:=I+1) NEQ 0 DO
            IF SD[I],[1:5]=0 THEN SD[I],[4:1]=1;
            % [4:1]= "TO GO TO AUXMEM ON FIRST OVERLAY"
        END;
% POP OMIT % AUXMEM

% ****      ***** *          ***** *****
% * * * * * * * * * *
% * * ***** * * * *
% * * * * * * * * * *
% ****      * * ***** ***** *****

        STREAM(D:=DALOCROW[MIX])=[MIGETSPACE(DALOC SIZE,0,0)+2]] &
        DALOC SIZE[8:38:10]);
        BEGIN
        SII=D; SI:=SI-8; DS:=DALOC SIZE WDS;
        END;
        IF OLAYDISK NEQ 0 THEN % OLAY DISK OBTAINED ABOVE
        BEGIN
        DALOC[MIX,0] := @200002;
        DALOC[MIX,1] := OLAYDISK;
        OLAYDISK := 0;
        END;
        OLAYMASK := TWO(MIX) OR OLAYMASK; % OLAYS NOW ALLOWABLE

% ***** ***** ***** ***** ***** * *
% * * * * * * * * * * * * * * * * * * *
% * * * * * * * * * * * * * * * * * * *
% * * * * * * * * * * * * * * * * * * *
% ***** ***** * * * * * * * * ***** * **

% PLACE "COMMON" VALUE IN FIRST SIMPLE VARIABLE IN THE PRT
NT1 := S[19]; % COMMON VALUE IN SHEET[19]
FOR I:= @25 STEP 1 WHILE NT1 NEQ 0 AND I LSS SEGO[3] DO
    IF TRP[I]=0 THEN % SIMPLE VARIABLE (NOT A DESCRIPTOR)
        BEGIN
        TRP[I]:=NT1;
        NT1:=0;
        END;
    DELINK; % DELINK SHEET ENTRY FROM SHEET QUEUE

EXIT;

P([RETURNRCW], STS, 0, RDS, 0, XCH, P&P[CTF], STF);
END PROCEDURE SELECTRUN2;

% FOR ADDITIONAL INFORMATION CONCERNING THE SHEET, SEE THE
% DOCUMENT AT SEQUENCE NUMBER 20512000

PROCEDURE SELECTRUN(F); VALUE F; REAL F;
BEGIN
    REAL    MSCW      = -2,
           F          = -1,
           MYMSCW     = -1,
           RCW        = +0,
           I          = +1,
           T          = +2,

```

```

20125600
20125700
20125800
20125900
20126000
20126100
20126200
20126600
20126700
20126800
20126900
20127000
20127100
20127200
20127300
20127400
20127500
20127600
20127700
20127800
20127900
20128000
20128100
20128200
20128300
20128400
20128500
20128600
20128700
20128800
20128900
20129000
20129100
20129200
20129300
20129400
20129500
20129600
20129700
20129800
20129900
20130000
20130100
20140000
20140100
20140200
20140300
20140400
20140500
20140600
20140700
20140800
20140900
20141000
20141100
20141200
20141300
20141400
20141500
20141600

```

```

L           = +3,
DT          = +4,
MIX         = +5,
HDR         = +6,
LEVEL      = +7,
MCPJOB     = +8,
OLAYDISK   = +9,
THISLINK   = +10,
NEXTLINK   = +11,
PREVLINK   = +12,
TYPE       = +13,
STACKLOC   = +14,
SHEETLOCKED = +15;

ARRAY  S           = +16[*],
      SEGO        = +17[*],
      TRP         = +18[*],
      LBL         = +19[*],
      SD          = NT2[*],
      TSKA        = NT2[*];

REAL  BELOW       = LBL + 1,
      SWAPDISK    = BELOW + 1,
      SWAPDISKSIZE = SWAPDISK + 1,
      UVSPACE     = SWAPDISKSIZE + 1,
      SVALUE      = UVSPACE,
      RETURNMSCW  = UVSPACE + 1,
      RETURNRCW   = RETURNMSCW + 1;

%%%%          ***NOTE***
%%%% RETURNMSCW AND RETURNRCW ***MUST*** BE THE LAST TWO
%%%% VARIABLES DECLARED IN THIS PROCEDURE.

DEFINE XCLOCKTIME =
  (((NT2:=(XCLOCK DIV 3600)) MOD 60 + (NT2 DIV 60)*100 +
   0.5 ) DIV 1)#;

$ SET OMIT = NOT(PACKETS)
  DEFINE UNITNO = S[23],[2:6]#; % ORIGINATING UNIT
$ POP OMIT

LABEL START, CONTINUE, LOAD, PASS, WINDUP, QUIT;
LABEL JARSPACE, TRYAGAIN;

SWITCH SW := QUIT, START, CONTINUE, QUIT, QUIT, PASS;

COMMENT THE VALUE OF "TYPE" MAY DETERMINE WHICH PORTIONS OF
PROCEDURES "SELECTRUN1" AND/OR "SELECTRUN2" WILL BE EXECUTED,
PROCEDURE "SELECTRUN1" AND "SELECTRUN2" MAY, IN TURN, SPECIFY
THE BRANCH POINT IN THIS PROCEDURE.
THE FOLLOWING DEFINES ARE USED TO SPECIFY THE BRANCH POINT
IN SWITCH "SW".
END OF COMMENT;

DEFINE STARTING      = 1#,
      CONTINUEING    = 2#,
      QUITTING      = 3#,
      RUNNING        = 4#,
      PASSING        = 5#,
      EQUATING       = 6#;

```

```

20141700
20141800
20141900
20142000
20142100
20142200
20142300
20142400
20142500
20142600
20142700
20142800
20142900
20143000
20143100
20143200
20143300
20143400
20143500
20143600
20143700
20144900
20145000
20145100
20145200
20145300
20145400
20145500
20145700
20145800
20145900
20146000
20146100
20146200
20146300
20146400
20146410
20146419
20146420
20146421
20146500
20146600
20146700
20146800
20146900
20147000
20147100
20147200
20147300
20147400
20147500
20147600
20147700
20147800
20147900
20148000
20148100
20148200
20148300
20148400

```

```

P(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
RCW := RCW & P(,SELECTRUN,LOD)[CTC];

TYPE := STARTING;

START:

P1MIX := 0;
IF F NEQ 0 THEN % SHEET ENTRY PASSED AS A PARAMETER
BEGIN
S := IOQUE & F[CTC]; % SHEET ENTRY
HDR := F,[FF]; % CORE ADDRESS OF OBJECT FILE HEADER
END
ELSE
BEGIN
IF TYPE=STARTING AND NOT SHEETLOCKED THEN
BEGIN
SLEEP([TOGGLE],SHEETMASK);
LOCKTOG(SHEETMASK);
SHEETLOCKED := 1;
END;
P([SVALUE], STS);
SELECTRUN;
IF TYPE LSS 0 THEN
GO TO SW[TYPE:=ABS(TYPE)];
END;

CONTINUE:

P1MIX := 0;
IF (MCPJOB :=
((S[1] EQV "DISK ")=(NOT 0)) AND
(((S[0] EQV "PRNPBT ")=(NOT 0)) OR
((S[0] EQV "LIBMAIN")=(NOT 0)) OR
((S[0] EQV "LDCNTRL")=(NOT 0)))) THEN
IF (((S[0] EQV "PRNPBT ")=(NOT 0)) AND AUTOPRINT) OR
(((S[0] EQV "LDCNTRL")=(NOT 0)) AND CDONLY) THEN
MCPJOB,[1:1] := 1;

% NOTE: A NEGATIVE SIGN FOR MCPJOB IMPLIES THAT THIS JOB
% SHOULD BE STARTED REGARDLESS OF THE AVAILABILITY OF
% SYSTEM CORE.

IF INTSIZE=0 THEN % NO INTRINSICS FILE
IF NOT(MCPJOB) THEN % NOT "SYSTEM" PROGRAM
BEGIN
STREAM(NT3:=NT3:=SPACE(4));
DS:=24 LIT"#NO INTRINSICS FILE...";
SPOUT(NT3);
SLEEP([INTSIZE],@1777);
END;

% ***** * * * ***** ***** **** *** ***** * *
% * * * * * * * * * * * * * * * * * * * * * * * * * * *
% ***** * * * ***** ***** * * * ***** ***
% * * * * * * * * * * * * * * * * * * * * * * * * * * *
% ***** ***** * * * ***** **** *** ***** * *

```

```

20148500
20149000
20149200
20149300
20149400
20149500
20149600
20149700
20149800
20149900
20150000
20150100
20150200
20150300
20150400
20150500
20150600
20150700
20150800
20150900
20151000
20151100
20151200
20151300
20151400
20151500
20151600
20151700
20151800
20151900
20152000
20152100
20152200
20152300
20152400
20152500
20152600
20152700
20152800
20152900
20153000
20153100
20153200
20153300
20153400
20153500
20153600
20153700
20153800
20153900
20154000
20154100
20154200
20154400
20154500
20154600
20154700
20154800
20154900
20155000

```



```

% S[2],[1:2] :: [0=NORMAL, 1=NOT USED, 2=XSEED, 3=ES-ED] 20164400
IF (I=CORE,[4:14]/100) GTR 0 THEN % FACTOR GTR 0 20165300
  IF MCPJOB THEN I:=1.10 * I; % TRY AND FORCE IT IN 20165400
IF CORE,[FF] + S[20] GTR CORE,[CF]*I THEN GO TO PASS; 20165500
END % IF "RUN" JOB 20165700
ELSE 20165800
BEGIN % NOT A "RUN" JOB 20165900
IF S[2] LSS 0 OR MCPJOB,[1:1] THEN GO TO JARSPACE; 20166000
IF BACKGROUND THEN % "NOBATCH" OPTION NOT SET 20166100
IF (LOGLINE := S[26]) LEQ 0 % NOT FROM CANDE 20166200
OR (IF LOGLINE,[40:8] GTR LMAX THEN 0 ELSE 20166300
  SCHEDULE[LOGLINE,[40:8]]) THEN % CANDE (TASK) SCHEDULE 20166400
  IF BATCHSELECT(F, S[20], BELOW, SHEETLOCKED) THEN GO TO PASS; 20166500
END; 20166600

% ***** 20166800
% * * * * * 20166900
% * * * * * 20167000
% * * * * * 20167100
% * * * * * 20167200
% ***** 20167300

JARSPACE: 20167400
% FIND A MIX SLOT FOR THIS JOB 20167500
FOR MIX:=1 STEP 1 UNTIL MIXMAX DO 20167600
  IF JAR[MIX,*]=0 THEN GO LOAD; 20167700

% NO FREE SPACE IN JAR: PASS ENTRY WITHOUT DELINKING AND CONTINUE 20167800
NEEDSELECT := 1; % CALL SELECTION NEXT EOJ 20167900
GO TO PASS; 20168000

% ***** 20168100
% * * * * * 20168200
% * * * * * 20168300
% * * * * * 20168400
% ***** 20168500
% * * * * * 20168600
% * * * * * 20168700
% * * * * * 20168800
% * * * * * 20168900
% * * * * * 20169000
% * * * * * 20169100
% ***** 20169200

LOAD: 20169300
JARROW[MIX] := IOQUE & HDR[CTC]; % FILE HEADER BECOMES JAR ROW 20169400
PRTRW[MIX] := 0; 20169500
UVROW[MIX] := UVROW[0] & UVSPACE[CTC] & BELOW[7:1:1]; 20169600
UVSPACE := 0; % MARK IT EMPTY (NO "FORGETSPACE" LATER) 20170100
IF BELOW THEN 20170200
CORE,[FF] := CORE,[FF] + S[20]; % ADD IN THE CORE ESTIMATE 20170300
$ SET OMIT = NOT(PACKETS) 20170400
IF (I:=S[23],[2:6]) GEQ 32 THEN PSEUDOMIX[MIX]:=I; % PSEUDO=RDR JOB 20170600
$ POP OMIT % PACKETS 20170700
JAR[MIX,0] := S[0]; 20170800
JAR[MIX,1] := S[1]; 20170900
JAR[MIX,2]:=S[2]&(IF (NT1:=S[2],[8:10])=5 THEN 2 ELSE NT1)[8:38:10]; 20171000
% IF THIS IS A "RUN" JOB, CHANGE IT TO SAY "EXECUTE" 20171100
% JAR[MIX,2],[8:10] = SHEET[2],[8:10] = 20171200
% 0 = "GO" PART OF COMPILE AND GO 20171300
% 1 = COMPILE AND GO 20171400
% 2 = EXECUTE 20171500
% 3 = COMPILE FOR SYNTAX 20171600
% 4 = COMPILE TO LIBRARY 20171700
% 5 = RUN JOB 20171800
STREAM(A:=JAR[MIX,3],[30:18], D:=[DT]); % CREATION DATE FROM HDR 20171900
20172000
20172100

```



```

BEGIN
  S1:=LOC A; DS:=8DEC;
  END;
GIMEDATE([DT],[CF],-DT); % CONVERT DATE TO "MMDDYY" FORMAT
JAR[MIX,3] := S[16]; % PROCESS TIME LIMIT
JAR[MIX,4] := S[17]; % I/O TIME LIMIT
STREAM(DATE, A:=[1]); % CONVERT DATE TO OCTAL FOR LOGGING
  BEGIN
    S1:=LOC DATE; DS:=8OCT;
    END;
JAR[MIX,5]:=(XCLOCK+P(RTR)) & I[1:25:23]; % DATE AND TIME
JAR[MIX,6] := S[6]&S[23][2:2:6]; % CARD/PSEUDO RDR. UNITNO IN [2:6]
JAR[MIX,7] := S[14]; % ACTUAL MFID OF OBJECT PROGRAM
% JAR[MIX,8] THROUGH JAR[MIX,29] STILL CONTAIN CONTENTS OF
% OBJECT FILE HEADER AS OBTAINED ABOVE
JAR[MIX,9] := M[HDR INX 9],[CF] & MCPJOB[1:47:1] &
  (S[2],[4:1] AND NOT(S[2],[2:1] OR AUTOMESS))[2:47:1];
  % S[2],[4:1]=1 MEANS SUPPRESS BOJ/E0J MESSAGES
  % MARK JAR[9],[1:1]=1 FOR "LIBMAIN","PRNPBT","LDCNTRL"
%% SEE ALSO "SEGMENT ZERO" SECTION IN PROCEDURE "SELECTRUN2" FOR
%% FURTHER ALTERATIONS TO THE JAR.

% *****
% * * * * *
% * *****
% * * * * *
% * * * * *
% * * * * *

IF S[2],[2:1] OR (S[21] LSS 128) THEN S[21]:=128;
% S[2],[2:1]=1 WHEN ES=ED, S[21] CONTAINS STACK SIZE
SINFO[MIX] := 0 & S[20][CTF] & ((CLOCK+P(RTR)) DIV 60)[1:31:17];
SQ[MIX] := -1;
DAT[MIX] := -1;
PRYOR[MIX] := -1;
PROCTIME[MIX]:= -S[16]-CLOCK-P(RTR); % PROCESS TIME LIMIT IN S[16]
$ SET OMIT = NOT(NEWLOGGING)
LOGSTOPPED[MIX] := 0; % LOGGING IS ALLOWABLE
$ POP OMIT % NEWLOGGING
IOTIME[MIX] := *S[17]; % I/O TIME LIMIT IN S[17]

% % % % % % % % % %
% % % % % % % % % %
P1MIX:=MIX; % % % % % % % % % %
% % % % % % % % % %
% % % % % % % % % %

USERCODE[MIX]:=ABS(S[24]); % USERCODE IN S[24]
IF S[2],[8:10]=0 THEN FORGETESPDISK(S[25]); % FORGET OBJ,SKELETON
% S[2],[8:10]=0 FOR "GO" PART OF "COMPILE AND GO"
IF (LOGLINE:=S[26]),[33:7] NEQ 0 THEN % CANDE JOB
  BEGIN
    STABLE[LOGLINE,[40:8]],MIXFLAG I= I I=
    (IF JAR[MIX,0],[1:2] NEQ 0 THEN 32 ELSE 0) + MIX;
    DAT[MIX],NDSABLE:=JAR[MIX,0],[2:1];
    % JAR[MIX,0],[2:1]=JOB NOT DS=ABLE BY USER
    % JAR[MIX,0],[1:1] = COMPILE JOB
    IF I LSS 32 THEN TWXOUT(0,0,1,LOGLINE);
    % SEND CARRIAGE RETURN/LINE FEED TO INDICATE BOJ
  END;
REPORTBACK(BOJW,MIX,0);

```

```

20172200
20172300
20172400
20172500
20172600
20172700
20172800
20172900
20173000
20173100
20173200
20173300
20173500
20174100
20174200
20174300
20174400
20174500
20174600
20176200
20176300
20176400
20176500
20176600
20176700
20176800
20176900
20177000
20178100
20178200
20178400
20178500
20178600
20179100
20179200
20179300
20179400
20179500
20179600
20179700
20179800
20179900
20180000
20180100
20180200
20180300
20180400
20180800
20180900
20181300
20181400
20181500
20181600
20181700
20181800
20181900
20182000
20182100
20182200
20182300

```

IF BELOW THEN	20182400
BEGIN	20182500
FRONTEND(MIX);	20182600
STASUS[MIX]:=SELECTING;	20182700
END	20182800
ELSE	20182900
BEGIN	20183000
DISKSTORE[MIX]:=SWAPDISK;	20183100
SWAPDISK := 0; % MARK IT EMPTY	20183200
INITIALSWAP(S[20],[CF]);	20183300
IF S[20],[2:1] THEN MAXCORE[MIX]:=1; % "CANT EXPAND" BIT	20183400
END;	20183500
HDR := JARROW[MIX],[CF]; % "FRONTEND" MOVES THE JAR ROW	20183600
STREAM(QI=FSROW[MIX]):=[M[GETSPACE(4,0,1)+2]]&4[8:38:10];	20183610
DS:=32LIT"0";	20183620
	20183700
IF ((S[0] EQV "CANDE ")=(NOT 0)) THEN	20183800
IF ((S[1] EQV "TSHARER")=(NOT 0)) THEN	20183900
BEGIN	20184000
I := S[2],[2:1]; T:=S[18];	20184100
STARTCANDY(I, T);	20184200
S[2],[2:1] := I; S[18] := T;	20184300
END;	20184400
% S[2],[2:1]=1 FOR ES=ED JOB, S[18]=PRIORITY	20184500
	20184600
NT1 := IF BELOW THEN 518400000 ELSE 180;	20184700
IOCOUNT[MIX] := -1;	20184800
ELAPSEDLIMIT[MIX] := IOTIME[MIX]+NT1+NT1;	20184900
PROCLIMIT[MIX] := PROCTIME[MIX]+CLOCK+P(RTR)+NT1;	20185000
OLAYCTR[MIX] := UPOLAY(NT1);	20185100
\$ SET OMIT = NOT(AUXMEM)	20185300
AUXCODE[MIX]:=AUXDATA[MIX]:=0; % AMOUNT OF AUXMEM USED FOR THIS JOB	20185400
AUXERRORTOG:=(*P(DUP)) AND NOT(TWO(MIX)); % MASK FOR AUXMEM RECOVERY	20185500
\$ POP OMIT % AUXMEM	20185600
	20185700
TYPE := CONTINUEING;	20185800
	20185900
% SELECTRUN2 IS CONCERNED WITH:	20186000
%  BOJ MESSAGE	20186100
%  SEGMENT ZERO	20186200
%  STACK AND PRT	20186300
%  SEGMENT DICTIONARY	20186400
%  DALOC	20186500
%  COMMON	20186600
	20186700
P([SVALUE],STS);	20186800
SELECTRUN2;	20186900
IF TYPE LSS 0 THEN	20187000
GO TO SW[TYPE:=ABS(TYPE)];	20187100
	20187200
IF (SEGO[7],[CF]=0) THEN % BUILD A DUMMY FILE PARAMETER BLOCK	20187300
TRP[3]:=[M[GETSPACE(1,0,1)+2]] ELSE	20187400
BEGIN	20187500
TYPE := EQUATING; % BUILD FPB AND PROCESS LABEL EQUATION	20187600
P([SVALUE],STS);	20187700
SELECTRUN1;	20187800
IF TYPE,[1:1] THEN GO TO SW[TYPE:=ABS(TYPE)];	20187900
END;	20188000
	20188100
FPBD[MIX] := TRP[3];	20188300

```

% TRP[3] VALUE SET BY SELECTRUN1 FOR NON-MCP TYPE JOB
GO TO WINDUP;

PASS;

TYPE := PASSING;
P([SVALUE],STS);
SELECTRUN1;
IF TYPE LSS 0 THEN
    GO TO SW[TYPE:=ABS(TYPE)]; % SELECTRUN1 DETERMINES BRANCH POINT

WINDUP;

%      ***  **  *   ***  *****      ***  *****  ****
%      *   **  *   *   *           *   *   *   *   *
%      *   *  *   *   *           *   *   *   ****
%      *   *  **  *   *           *   *   *   *   *
%      ***  *  **  ***  * 0      ****  *****  ****

% INITIALIZE OTHER PRT CELLS
TRP[0] := WORDOFFEASE;
TRP[2] := MEMORY;
TRP[10] := TRP&(STACKLOC+1)[18:33:15];
IF JAR[MIX,0] LSS 0 THEN % COMPILE JOB
    BEGIN
        IF(NT1:=JAR[MIX,2],[8:10])=4 THEN % COMPILE TO LIBRARY
            TRP[26]:=S[22] % SAVE FACTOR FOR OBJECT FILE IN SHEET[22]
        ELSE IF NT1=3 THEN % COMPILE FOR SYNTAX ONLY
            BEGIN
                TRP[26]:=-1; % SAVE FACTOR = (-1) % DONT SAVE OBJECT
                JAR[MIX,2],[8:10]:=2; % MARK IT AN "EXECUTE" JOB
            END;
        END; % COMPILE JOBS
    TRP[6]:=FLAG(O&[TRP[6]][18:33:15]&32[8:38:10]);
    IF JAR[MIX,2] GEQ 0 THEN % NOT COBOL
        TRP[11]:=FLAG(O&[TRP[11]][18:33:15]&8[8:38:10]); % "OAT" ENTRY
    % BRING IN STARTING SEGMENT&BUILD CONTROL WORDS FOR INITIATE%
    MAKEPRESENT(TRP INX POLISH(SEG0[6],TRP[4],INX,LOD),[8:10]);
    % SEG0[6] = STARTING SEGMENT NUMBER
    % SEGDICT[SEG0[6]],[8:10] = PRT LOCN, OF DESC, FOR STARTING SEGMENT
    M[STACKLOC+2]:= -FLAG(POLISH(SEG0[6],TRP[4],INX,LOD),[18:15]);
    M[STACKLOC+1]:= -FLAG(O&(TRP)[6:33:9]);
    M[STACKLOC] := @2222222222222222;
    TRP[8] := -FLAG(STACKLOC+2); % INITIATE CONTROL WORD
    IF(NT1:=TRP[4],[18:6]) NEQ 0 THEN
        INTABLEROW[MIX]:=INTABLEROW[NT1]
    ELSE IF NOT(JAR[MIX,9],[1:1]) THEN % NOT SYSTEM TYPE JOB
        BEGIN
            IF BELOW THEN I:=INTSIZE ELSE
                I:=INT13SIZE+INT13START;
            INTABLEROW[MIX]:=[M[GETSPACE(I,1,1)+2]]&I[8:38:10];
            STREAM(A:=I,T:=INTABLEROW[MIX]);
            BEGIN
                S:=T; S:=S-8; DS:=A WDS;
            END;
        END;

FORGETSPACE(SEG0);
IF S[2],[2:1] THEN % S[2],[2:1]=1 WHEN ES=ED, CALL TERMINATE
    BEGIN

```

```

20188800
20188900
20189000
20189100
20189200
20189300
20189400
20189500
20189600
20189700
20189800
20189900
20190000
20190100
20190200
20190300
20190400
20190500
20190600
20190700
20190800
20190900
20191000
20191100
20191200
20191300
20191400
20191500
20191600
20191700
20191800
20191900
20192000
20192100
20192200
20192300
20192400
20192500
20192600
20192700
20192800
20192900
20193000
20193100
20193200
20193300
20193400
20193500
20193700
20193800
20194300
20194400
20194500
20194600
20194700
20194800
20194900
20195000
20195100
20195200

```

```

        TERMINATE(MIX & 35[CTF]);
        END;
$ SET OMIT = NOT(NEWLOGGING)
  STOPLOG(MIX,0); % STOP LOGGING TIME FOR THIS JOB
$ POP OMIT % NEWLOGGING
  SAVEMIX(MIX,LOGLINE);
  PRYOR[MIX] := S[18]; % PRIORITY IN SHEET[18];
  STASUS[MIX]:=RUNNING;
$ SET OMIT = NOT(STATISTICS)
  TIMING[MIX]:=CLOCK+P(RTR);
$ POP OMIT
  IF F=0 THEN % SHEET ENTRY NOT PASSED AS A PARAMETER
    BEGIN
      TYPE := (IF S[2],[1:1] THEN STARTING ELSE CONTINUEING);
      % IF ES=ED THEN RE=START SHEET SEARCH; OTHERWISE,CONTINUE ON
      GO TO START;
    END;

QUIT;

P1MIX := 0;
IF SHEETLOCKED THEN UNLOCKTOG(SHEETMASK);
IF S NEQ 0 THEN FORGETSPACE(S); % SPACE FOR SHEET ENTRY
IF OLAYDISK NEQ 0 THEN FORGETUSERDISK(OLAYDISK,-500);
IF SWAPDISK NEQ 0 THEN FORGETUSERDISK(SWAPDISK,-SWAPDISKSIZE);
IF UVSPACE NEQ 0 THEN FORGETSPACE(UVSPACE);
KILL([F] INX NOT 1);
END SELECTION ROUTINE;
DEFINE%
  COMMA      = 10#,%
  EQUAL      = 11#,%
  PERIO      = 12#,%
  SLASH      = 13#,%
  QUEST      = 14#,%
  POUND      = 15#,%
  SPECI      = 19#,%
  IDENT      = 20#,%
  UNLOCKV    = 22#,% A SWITCH LABEL (FUNC) IN
  USEV       = 23#,% SECURITYMAINT USES THE ORDER OF
  LOCKV      = 24#,% VALUES OF "UNLOCKV" THROUGH "OPEN".
  FREE       = 25#,%
  OPEN       = 26#,%
  PACKET     = 27#,%
  USER       = 28#,%
  RUNV       = 29#,%
  COMPI      = 30#,%
  EXECU      = 31#,%
  COPY       = 32#,%
  UNLOAD     = 33#,%
  ADDV       = 34#,%
  ENTER      = 35#,%
  REMOV      = 36#,%
  CHANG      = 37#,%
  ENDFI     = 39#,%
$ SET OMIT = NOT(PACKETS)
  WAITV     = 40#,%
$ POP OMIT
  DATAV    = 41#,%
  LABEV     = 42#,%
  SETV      = 43#,%

```

```

20196000
20197900
20198000
20198100
20198200
20198300
20198500
20198700
20198800
20198900
20199000
20199200
20199300
20199400
20199500
20199600
20199700
20199800
20210000
20210100
20210200
20210300
20210400
20210600
20210900
20211000
20211100
20211600
20212000
20213000
20214000
20215000
20216000
20217000
20217500
20218000
20219000
20219050
20219060
20219100
20219200
20219300
20219310
20219400
20219500
20220000
20221000
20222000
20223000
20224000
20224500
20225000
20225500
20226000
20226099
20226100
20226101
20226500
20227000
20228000

```

%LP 1

	RESETV	= 44#,%		20228100
	FILEV	=47#,		20228200
	EXPIRED	=48#,		20228300
	ACCESSD	=49#,		20228400
	PROCE	= 50#,%	A STORE NEAR THE END OF PCC	20229000
	IO	= 51#,%	MAKES USE OF THE ORDER AND VALUES	20230000
	PRIOR	= 52#,%	OF "PROCE" THRU "SAVEV",	20231000
	COMMONV	= 53#,%		20232000
	COREV	= 54#,%		20232500
	STACK	= 55#,%		20233000
	SAVEV	= 56#,%	(SAVE #DAYS ON COMPILE TO LIBRARY)	20233500
	ALGOL	= 60#,%		20234000
	FORTTRAN	= 62#,%		20235000
	TSPOL	=63#,		20235050
	BASIC	= 64#,		20235075
	COBOL68	=65#,		20235080
	WITH	= 66#,		20235099
	COBOL	= 67#,		20235100
	LIBRA	= 68#,%		20236000
	SYNTA	= 69#,%		20237000
	FROM	= 70#,%		20238000
	TOV	= 71#,%		20239000
	FORM	= 78#,	%SWITCH D(PCC) "FORM"-"SPECIAL"%	20240000
	NO	= 79#,%		20241000
	DISK	= 80#,%		20242000
	TAPE	= 81#,%		20243000
	PUNCH	= 82#,%		20244000
	PRINT	= 83#,%		20245000
	BACK	= 85#,%		20246000
	SPECIAL	=89#,%		20247000
	EU	=91#,		20247600
	SLOW	=92#,		20247700
	B6500	=93#,		20247800
	FAST	= 94#,		20247900
	COPYN	=95#,		20247910
	MAXV	= 96#,		20247920
	FREEF	=97#,		20247930
	FIXED	= 98#,		20247940
	SENSE	= 100#,		20247950
	LATESTV	= 101#,		20247960
	PAPER	= 84#;%		20248000
COMMENT	RESWDS	CONTAINS RESERVED WORDS FOR CONTROL CARDS;%		20249000
REAL	MSCW	= -2,		20288100
	CARD	= MSCW+1,	MYMSCW = CARD,	20288105
	RCW	= +0,		20288110
	PROCVAL	= RCW+1,	%IN CASE OF TYPED PROCEDURES	20288115
	A	= PROCVAL+1,	T = A,	20288120
	CADDR	= A+1,	SFID = CADDR,	20288125
	CARDLOC	= CADDR+1,		20288130
	CDEX	= CARDLOC+1,	SDEX = CDEX,	20288135
	CLOSET	= CDEX+1,		20288140
	CMPLR	= CLOSET+1,		20288145
	CN	= CMPLR+1,		20288150
	INV	= CN+1,		20288155
	KOUNT	= INV+1,		20288160
	LASTSCAN	= KOUNT+1,		20288165
	LIBNO	= LASTSCAN+1,		20288170
	N1	= LIBNO+1,		20288175
	N2	= N1+1,		20288185
	N3	= N2+1,		20288190

	N4	=	N3+1,	U	=	N4,	20288195
	OPTN	=	N4+1,				20288200
	OPTNN	=	OPTN+1,				20288205
	PADDR	=	OPTNN+1,	SFH	=	PADDR,	20288210
	PDEX	=	PADDR+1,	SMID	=	PDEX,	20288215
	PPCPROCESS	=	PDEX+1,				20288220
	SFD	=	PPCPROCESS+1,				20288225
	SMD	=	SFD+1,				20288230
	SOURCE	=	SMD+1,				20288235
	SPOUTUNIT	=	SOURCE+1,				20288240
	ST	=	SPOUTUNIT+1,				20288245
	T1	=	ST+1,				20288250
	UNITNO	=	T1+1,				20288255
	USERID	=	UNITNO+1;				20288260
ARRAY	ACCUM	=	USERID+1[*],				20288265
	CEQN	=	ACCUM+1[*],				20288270
	CMM	=	CEQN+1[*],				20288275
	DIRECT	=	CMM+1[*],				20288280
	NB	=	DIRECT+1[*],				20288285
	PEQN	=	NB+1[*],				20288290
	PROG	=	PEQN+1[*];				20288295
NAME	ADDR	=	PROG+1;				20288300
BOOLEAN	ABORT	=	ADDR+1,				20288305
	TOG	=	ABORT+1;				20288310
REAL	RETURNMSCW	=	TOG+1,	% THESE LOCALS MUST BE THE LAST			20288315
	RETURNRCW	=	RETURNMSCW+1,	% THREE LOCALS OF CONTROLCARD			20288320
	RETURNVAL	=	RETURNRCW+1;				20288325
							20289000
							20289009
							20289010
							20289020
							20289025
							20289030
							20289035
							20289040
							20289045
							20289050
							20289055
							20289060
							20289065
							20289080
							20289085
							20289090
							20289095
							20289100
							20289105
							20289110
							20289113
							20289115
							20289116
							20289120
							20289125
							20289130
							20289135
							20289140
							20289142
							20289144
							20289146
							20289148
							20289150

```

$ SET OMIT = NOT(PACKETS)
PROCEDURE PRINTTHECOVER(CARD,UNITNO,PS);
VALUE CARD,UNITNO,PS; REAL CARD,UNITNO,PS;
% TO ALTER SIZE OF ONE-AREA PACKET PAGE CHANGE DEFINE AT 02113091
BEGIN LABEL L,TRYAGAIN;
REAL BUF,T,TP,X;
INTEGER I,PAGEADDR; ARRAY HEADER[*];
SUBROUTINE BUILDHEADER;
BEGIN
HEADER:=10QUE & BUF[CTC];
M[BUF]:=0;
MOVE(29,BUF,BUF+1);
STREAM(DATE,H3:=HEADER INX 3);
BEGIN SI:=LOC DATE; DS:=8OCT; % CREATION
DI:=H3; DSI:=2LIT"+#"; % SAVE 10
SI:=H3; SII:=SI+5; DSI:=3CHR; % ACCESSED
END;
HEADER[0]:=00013200132000103; % 90,90,1,3
HEADER[1]:=(XCLOCK+P(RTR)) & HEADER[3][6:30:18];
HEADER[2]:=MCP;
HEADER[4]:=0 & (@1001)[2:38:10];
HEADER[5]:="PACKET ";
HEADER[7]:=(PAGE SIZE DIV 3)-1;
HEADER[8]:=PAGE SIZE;
HEADER[9]:=1;
HEADER[10]:=PAGEADDR;
END BUILDHEADER;
TRYAGAIN;
CIDTABLE[UNITNO=32,6]:=TP:= 001 & NEXTCDNUM(1)[6:24:24];
IF DIRECTORYSEARCH("PBD ",TP,5)≠0 THEN GO TRYAGAIN;
BUF:=GETSPACE(90,64,5)+2;
PAGEADDR:=GETUSERDISK(PAGE SIZE);

```

```

PS:=
IF PS=0 THEN "CRA" ELSE IF PS=1 THEN "CRB" ELSE
IF PS=2 THEN TINU[UNITNO],[30:18] ELSE
IF PS=3 THEN "ZIP" ELSE
" ";
STREAM(CARD,TP:=CIDTABLE[UNITNO-32,2],PS
, NI:=CIDTABLE[UNITNO-32,7]+1,BUF);
BEGIN DS:=8LIT" "; SI:=BUF; 2(DS:=44 WDS);
SI:=LOC N; DI:=LOC N; DS:=8DEC; DI:=LOC N; DS:=8FILL;
SI:=LOC N; SI:=SI+3; DI:=BUF; DI:=DI+12;
DS:=7LIT"INPUT "; DS:=5CHR; DS:=12LIT" CARDS FROM ";
SI:=LOC PS; SI:=SI+5; DS:=3CHR;
DI:=BUF; 4(DI:=DI+34); DS:=8LIT":x≥14000"; BUF:=DI;
SI:=LOC TP; SI:=SI+2; DI:=DI+12;
DS:=8LIT"PACKET "; DS:=4CHR;
DI:=BUF; 4(DI:=DI+34); DS:=8LIT":x≥14000"; BUF:=DI;
SI:=CARD; DS:=9WDS;
DI:=BUF; 4(DI:=DI+34); DS:=8LIT":x≥12000"; BUF:=DI;
54(DS:=LIT"#"); DS:=11LIT" ABORTED "; 55(DS:=LIT"#");
DI:=BUF; 4(DI:=DI+34); DS:=8LIT":x≥12000"; BUF:=DI;
DS:=16LIT"+ABORTEDOPAGE "; DS:=9LIT"OPACKET 0";
SI:=LOC TP; SI:=SI+2; DS:=4CHR; DI:=DI+3;
SI:=CARD; DS:=8WDS; DS:=8LIT"OPACKET ";
40(DS:=LIT"0");
END;
DISKWAIT(BUF,90,PAGEADDR);
STREAM(A:=I:=((NT1:=((XCLOCK+P(RTR)) DIV 3600)) MOD 60
+(NT1 DIV 60)×100),ACTDATE,WEEKDAY,BUF);
BEGIN
3(4(DI:=DI+34); DS:=8LIT":x0×2000"); BUF:=DI;
DS:=8LIT" "; SI:=BUF; DS:=34 WDS; DI:=BUF;
SI:=LOC WEEKDAY; DI:=DI+12; DS:=4LIT"DATE"; DI:=DI+4;
SI:=SI+2; 6(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR);
SI:=LOC ACTDATE; DS:=5LIT"DAY, ";
SI:=SI+2; 2(DS:=2CHR; DS:=LIT"/"); DS:=2 CHR;
DI:=BUF; 4(DI:=DI+34); DS:=8LIT":x≥14000"; BUF:=DI;
SI:=LOC A; DI:=DI+12; DS:=4LIT"TIME"; DI:=DI+4; DS:=4DEC;
DI:=BUF; 4(DI:=DI+34); DS:=8LIT":x≥14000";
END;
DISKWAIT(BUF,90,PAGEADDR+3);
GO TO L;
L: X:=6; M[BUF+17]:=0;
IF (T:=DIRECTORYSEARCH("MESSAGE","OTHE DAY",5))≠0 THEN
BEGIN
FOR I:=0 STEP 1 WHILE (I<6) AND NOT M[BUF+17] DO
BEGIN
DISKWAIT(=BUF,90,M[T+10]+3×I);
DISKWAIT(BUF,90,PAGEADDR+6+3×I);
X:=X+3;
END;
FORGETSPACE(T);
END;
STREAM(ML:=MARKLEVEL,PL:=PATCHLEVEL,LL:=LOCALEVEL
, [L:=M[3],BUF:=BUF+54]);
BEGIN DS:=8LIT" "; SI:=BUF; DS:=34 WDS; DI:=BUF;
DI:=DI+8; DS:=8LIT":x0Q0803";
DI:=DI+12; DS:=18LIT"#NO MESSAGES TODAY";
DI:=BUF; 4(DI:=DI+34); DS:=8LIT":x≥12002"; BUF:=DI;
DI:=DI+8; DS:=31LIT"*** BURROUGHS B5700 TSMCP MARK ";
SI:=LOC ML; IF SC GEQ " " THEN;

```

20289152  
20289153  
20289154  
20289155  
20289159  
20289160  
20289163  
20289165  
20289170  
20289175  
20289180  
20289185  
20289190  
20289195  
20289200  
20289205  
20289210  
20289215  
20289220  
20289225  
20289230  
20289235  
20289240  
20289245  
20289250  
20289255  
20289260  
20289265  
20289270  
20289275  
20289280  
20289285  
20289290  
20289295  
20289300  
20289305  
20289310  
20289315  
20289320  
20289325  
20289330  
20289335  
20289340  
20289345  
20289350  
20289355  
20289360  
20289365  
20289370  
20289375  
20289380  
20289385  
20289390  
20289395  
20289400  
20289405  
20289410  
20289415  
20289420  
20289425

```

      8(IF TOGGLE THEN IF SC="0" THEN SII=SI+1 ELSE DS:=CHR
        ELSE DS:=CHR); DS:=LIT",";
      SII=LOC PL; IF SC GEQ " " THEN;
      6(IF TOGGLE THEN IF SC="0" THEN SII=SI+1 ELSE DS:=CHR
        ELSE DS:=CHR); DS:=2CHR;
      SII=LOC LL; IF SC GEQ " " THEN;
      8(IF TOGGLE THEN IF SC="0" THEN SII=SI+1 ELSE DS:=CHR
        ELSE DS:=CHR); DS:=2LIT" AND INTRINSICS MARK ";
      SII=LOC ML; IF SC GEQ " " THEN;
      8(IF TOGGLE THEN IF SC="0" THEN SII=SI+1 ELSE DS:=CHR
        ELSE DS:=CHR); DS:=LIT",";
      SII=LOC IL; SII=SI+1; IF SC>"0" THEN DS:=CHR ELSE
      SII=SI+1; DS:=2CHR; DS:=4LIT" ***";
      DI:=BUF; 4(DI:=DI+34); DS:=8LIT"IX212001";
END;
DISKWAIT(BUF,90,PAGEADDR+X);
BUILDHEADER;
ENTERUSERFILE("PBD      ",TP,BUF=1);
PSEUDO[UNITNO=32]:=(P(DUP))&
      11[8:38:10]& % PACKETPBD
      (IF T#0 THEN 3 ELSE 2)[18:45:3]& % PACKETREC
      1[21:47:1]& % PACKETFREE
      (PAGEADDR+X)[22:22:26]; % PACKETPAGE
FORGETSPACE(BUF);
END PRINTTHECOVER;
$ POP OMIT
COMMENT  FETCH READS THE NEXT CONTROL CARD , SETS SOURCE TO BEGINNING
        OF CARD , SETS LAST WORD OF CARD TO PERIOD,      ;%
PROCEDURE FETCH(UNITNO,CARDLOC,SOURCE);
VALUE UNITNO,CARDLOC;
REAL UNITNO,CARDLOC,SOURCE ;
BEGIN%
REAL T,E;
E1=@14&UNITNO[45:1:1]; UNITNO:=ABS(UNITNO);
IF (UNITNO OR 1)=31 THEN %DATACOM OR ZIP
M[SOURCE:=CARDLOC]:=@1425452432373737 ELSE
BEGIN
$ SET OMIT = PACKETS
IF UNITNO>32 THEN T+
$ POP OMIT
$ SET OMIT = NOT(PACKETS)
IF UNITNO GEQ 32 THEN
DO UNTIL NOT E,[45:1] OR T:=
$ POP OMIT
      READFROMDISK(CIDROW[UNITNO=32],%
        [M[CARDLOC]]&10[8:38:10]) ELSE%
DO
BEGIN T:=IF UNITNO=30 THEN 0 ELSE
      WAITIO(CARDLOC INX @40000000,E,UNITNO);
      IF UNITNO=30 OR T,[45:1] THEN
      STREAM(CARDLOC);
      BEGIN DS:=LIT MARK;DS:=4 LIT "END." END;
      IF UNITNO=25 THEN
      BEGIN
      STREAM(T+0;CARDLOC);%
      BEGIN SI=CARDLOC;DI=LOC T;DI=DI+6;SI=SI-1;DS=2CHR;SI=SI-1;
      DI=CARDLOC;DI=DI-1;DS=LIT"<";8(60(IF SC="<" THEN
      BEGIN DS=CHR;JUMP OUT 2 TO L END;IF SC="<" THEN
      BEGIN DI=DI-1;IF SC#DC THEN DI=DI-1 END ELSE
      DS=CHR));
20289430
20289435
20289440
20289445
20289450
20289455
20289460
20289465
20289470
20289475
20289480
20289485
20289490
20289495
20289500
20289510
20289520
20289530
20289540
20289550
20289560
20289570
20289580
20289590
20289600
20289601
20290000
20291000
20292000
20292100
20292200
20293000
20294000
20294800
20295000
20295100
20295200
20295999
20296000
20296001
20296099
20296100
20296200
20296201
20297000
20298000
20298100
20298200
20299000
20299020
20299030
20299040
20299110
20299111
20300000
20301000
20301100
20301200
20301300
20301400

```



```

      LI DI=CARDLOC;DI=DI-1;SI=LOC T;SI=SI+6;DS=CHR;
      END;
      MAKELOG((CARDLOC,[CF])=1,SPIN);
      END ELSE P(0);
      END UNTIL P,[42:6]#31;
      M[(SOURCE + CARDLOC)+9]+0&"",[1:43:5];%
      END;
      END ;%
COMMENT THE SCAN ROUTINE IS USED FOR CONTROL CARD SCANNING.%
      SCAN RETURNS THE FOLLOWING RESULTS :%
      4 FOR IDENTIFIERS WHICH ARE NOT RESERVED%
      0 FOR PERIOD%
      1 FOR SLASH%
      2 FOR QUESTION MARK%
      5,.. FOR IDENTIFIERS IN DIRECT.%
      3 FOR OTHER SPECIAL CHARACTERS.%
      13 FOR "PRIORITY" ;%
REAL PROCEDURE SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,
      DIRECT);
      VALUE UNITNO,CARDLOC ;
      REAL UNITNO,CARDLOC,SOURCE, KOUNT,LASTSCAN ;
      ARRAY ACCUM[*],DIRECT[*];
      BEGIN%
      LABEL GOGO, TYPE0,TYPE1,TYPE2;%
      SWITCH TYPE + TYPE0,TYPE1,TYPE2 ;%
      REAL I;%
      LABEL PERPER;%
      GOGO;%
      IF LASTSCAN THEN%
      BEGIN IF LASTSCAN < 0 OR UNITNO = 31 THEN%
      BEGIN I + QUEST; LASTSCAN + 0; GO TO TYPE1 END;
      FETCH(UNITNO,CARDLOC,SOURCE);
      LASTSCAN:=0
      $ SET OMIT = NOT(PACKETS)
      &1[2:47:1];
      $ POP OMIT
      END;%
      I + IDENT;%
      STREAM (J=0,K=0,SOURCE ; ACCUM);%
      BEGIN%
      SI + SOURCE ; DI + ACCUM ; DI=DI+1;%
      LI IF SC = " " THEN BEGIN SI=SI+1; GO L END;%
      IF SC = ALPHA THEN%
      BEGIN%
      IF SC #@14 THEN GO TO L3;%
      DS + CHR ; TALLY + 1;%
      L1: 63(IF SC=ALPHA THEN BEGIN DS=CHR;%
      TALLY=TALLY+1 END ELSE JUMP OUT);%
      K=TALLY; TALLY=0; J=TALLY; DS=8 LIT" ";%
      END%
      ELSE IF SC = "" THEN%
      BEGIN SI + SI+1;%
      30(IF SC="" THEN JUMP OUT;
      DS:=CHR; TALLY:=TALLY+1);
      IF TOGGLE THEN % FOUND CLOSING QUOTE
      BEGIN DS:=8 LIT" "; SI:=SI+1;
      K:=TALLY; TALLY:=1; J:=TALLY;
      END
      ELSE % INVALID STRING
      BEGIN

```

```

20301500
20301600
20301650
20301700
20302000
20303000
20303900
20304000
20305000
20306000
20307000
20308000
20309000
20310000
20311000
20312000
20313000
20314000
20314050
20314100
20314200
20314300
20315000
20316000
20317000
20319000
20320000
20321000
20322000
20323000
20324000
20325000
20325100
20325109
20325110
20325111
20326000
20327000
20328000
20329000
20330000
20331000
20332000
20333000
20334000
20335000
20336000
20337000
20338000
20339000
20340000
20341000
20342000
20342250
20342500
20342750
20343000
20343250
20343500
20343750

```

```

                SI←SI-31; GO L3;
                END;
            END%
        ELSE BEGIN%
L3: %
            TALLY ← 2; J←TALLY; DI←LOC K; DI←DI+7; DS←CHR ;%
            END;%
            SOURCE ← S1;%
            END;%
COMMENT STACK NOW CONTAINS : 0 FOR IDENTIFIER & NO. OF CHRS%
                             1 FOR "ID"      & NO. OF CHRS%
                             2 FOR SPECIAL CHR & ACTUAL CHR  ;%

P([SOURCE],+);
P([KOUNT],+);
GO TO TYPE[POLISH];%
TYPE0: %
    BEGIN
    I←2; WHILE DIRECT[I+I+2]≠0 DO%
        IF (DIRECT[I] EQV ACCUM[0])≠ NOT 0 THEN%
            BEGIN IF DIRECT[I+1]≠QUEST OR (UNITNO=25 OR UNITNO≥30)
                AND CARDLOC,[CF]=SOURCE,[CF] THEN
                    BEGIN I←DIRECT[I+1]; GO TO TYPE1 END END;%
                I ← IDENT ; END;%
            GO TO TYPE1 ;%
TYPE2: %
    IF KOUNT≠"%" THEN ACCUM[0]← " 0" OR KOUNT;
    IF KOUNT="%" OR%
        KOUNT = "," THEN%
        BEGIN LASTSCAN ← 1;%
PERPER: I ← PERIO; GO TO TYPE1;%
        END;%
    IF KOUNT="%" THEN BEGIN IF UNITNO≥32 THEN
        IF CIDTABLE[UNITNO=32,3]≥
            CIDTABLE[UNITNO=32,7] THEN
                BEGIN I←ENDFI; GO TO TYPE1 END;
            IF UNITNO = 31 THEN
                BEGIN I←PERIO; GO TO TYPE1 END;
                FETCH(UNITNO,CARDLOC,SOURCE);
$ SET OMIT = NOT(PACKETS)
    IF UNITNO GEQ 32 AND NOT LASTSCAN,[2:1] THEN
        BEGIN STREAM(CARDLOC, I←I+SPACE(10));
        BEGIN DS←5LIT">";
            SI←CARDLOC; 2(DS+36 CHR); DS←LIT"←";
        END; SPOUTER(I,UNITNO,64);
    END;
$ POP OMIT
                GO TO GOGO;
            END;
    IF KOUNT = ";" THEN%
        BEGIN LASTSCAN ← -1; GO TO PERPER END;%
    I ← IF KOUNT = "/" THEN SLASH ELSE%
        (IF KOUNT = @14 THEN QUEST ELSE%
        (IF KOUNT = "," THEN COMMA ELSE%
        (IF KOUNT="=" THEN EQUAL ELSE %
        (IF KOUNT="#" THEN POUND ELSE SPECI)))));
TYPE1: SCN←I;
        END SCAN ;%
PROCEDURE SEEKNAM(A,B,C,D,E,N); VALUE A,B; REAL A,B,C,D,E,N;
    BEGIN
    LABEL FIND,L;

```

```

20344000
20344250
20345000
20346000
20347000
20348000
20349000
20350000
20351000
20352000
20353000
20354000
20355000
20356000
20357000
20358000
20361000
20362000
20363000
20364000
20364500
20365000
20366000
20367000
20368000
20368100
20369000
20370000
20371000
20372000
20373000
20374000
20374100
20374200
20374300
20374310
20374320
20374400
20374409
20374410
20374415
20374420
20374425
20374430
20374435
20374436
20374500
20374600
20375000
20376000
20377000
20378000
20379000
%LP 1 20380000
%LP 1 20380500
20381000
20382000
20382010
20382020
20382030

```

ARRAY NB[*];	20382040
REAL I,T; INTEGER J;	%10420382050
INTEGER J1,J2,J3,K1,K2;	%10420382052
LABEL RESTART;	%10420382054
IF C=0 THEN	%10420382056
BEGIN N:=SPACE(60)-1;	%10420382058
J1:=J3:=0; K1:=K2:=MODULUS-1;	%10420382060
IF A GEQ 0 THEN J1:=K1:=(A,[6:18]+A,[24:24]) MOD MODULUS;	20382062
IF B GEQ 0 THEN J3:=K2:=(B,[6:18]+B,[24:24]) MOD MODULUS;	20382064
END ELSE	%10420382066
BEGIN I:=(T:=M[N]),[42:6];	%10420382068
J1:=T,[36:6]; J2:=T,[30:6]; J3:=T,[12:6];	%10420382070
K1:=T,[24:6]; K2:=T,[18:6];	%10420382072
END;	%10420382074
NB:=[M[N+1]]&60[8:38:10];	%10420382076
IF C NEQ 0 THEN GO TO RESTART;	%10420382095
FOR J1:=J1 STEP 1 UNTIL K1 DO	%10420382100
FOR J2:=J3 STEP 1 UNTIL K2 DO	%10420382110
BEGIN J:=SCRAMBLE(J1,J2);	%10420382120
DO BEGIN	%10420382130
DISKWAIT(-N-1,60,J);	%10420382140
FOR I:=0 STEP 3 UNTIL 57 DO	%10420382150
BEGIN	%10420382160
IF (T:=NB[I]) NEQ @14 THEN	%10420382165
IF (T EQV A)=NOT 0 OR A<0 THEN	20382170
IF (NB[I+1] EQV B)=NOT 0 OR B<0 THEN GO FIND;	20382180
END;	%10420382190
END UNTIL (J:=NB[2],[FF])=0;	%10420382195
END;	20382200
FORGETSPACE(NB);	%10420382202
IF C=0 THEN N:=0 ELSE C:=0;	20382204
GO TO L;	%10420382206
FIND:	20382210
D:=NB[I];E:=NB[I+1];	20382220
C:=NB[I+2],[CF];	%10420382225
M[N]:=I&J1[36:42:6]&J2[30:42:6]&K1[24:42:6]&K2[18:42:6]&	%10420382226
J3[12:42:6];	%10420382227
L:	20382230
END;       % SEEKNAME	20382240
REAL PROCEDURE PPC	20383000
(ADDR,EQN,X,DEX,TYPE,UNITNO,CARDLOC,SOURCE,ACCUM,LASTSCAN,	20384000
DIRECT));	20384100
VALUE                   TYPE,UNITNO,CARDLOC                   ;	20385000
REAL ADDR,           DEX,TYPE,UNITNO,CARDLOC,SOURCE,       LASTSCAN ;	20386000
ARRAY EQN[*],X[*],ACCUM[*],DIRECT[*];	20386100
BEGIN%	20387000
REAL IOD,KOUNT;	20388000
LABEL EXIT,ERROR,NEXT,LFORM,LNO,LDISK,LTAPE,LPUNCH,LPAPER,%	20389000
ROUND,PROTECT,	20390000
SERIAL,UPDATE,SPO,DSKCHECK,                   % (SHM)	20391000
DOWN,%	20392000
LSPECIAL,LPRINT,LBACK,LCOPY,LFREE;	20393000
SWITCH D * LFORM,LNO,LDISK,LTAPE,LPUNCH,LPRINT,LPAPER,%	20394000
LBACK,SERIAL,UPDATE,SPO,%	20395000
LSPECIAL,ERROR,ERROR,ERROR,ERROR,ERROR,LCOPY,ERROR,	20396000
LFREE,ERROR,PROTECT;	20396010
REAL NOLBL,TPNO ;%	20397000
BOOLEAN FAROUT;	20397050
REAL SUBROUTINE SCAN;	20397100
BEGIN SCAN=SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,	20397200

```

                                DIRECT)
END;
IF TYPE = FILEV THEN%
BEGIN%
  IF ADDR = 0 THEN ADDR=X[13]+GETESPDISK ;%
  IF DEX = 2 THEN%
    BEGIN%
      EQN [29] = GETESPDISK;%
      DISKWAIT(EQN,[CF],30,ADDR);
      ADDR = EQN[29];%
      DEX = 0;%
    END;%
  IF (TYPE=SCAN) < IDENT THEN GO TO ERROR;
  EQN = (14 x DEX) INX EQN ;%
  EQN[12]=0; % ZERO OUT EU/SPEED CELL
  STREAM( KOUNT, ACCUM, Z + [EQN[4]]);%
  BEGIN%
    SI = LOC KOUNT ; SI+SI+7; DI+Z; DS+CHR;%
    SI = ACCUM ; SI+SI+1; DS= KOUNT CHR ;%
  END ;%
  IF X[0]<0 THEN IF KOUNT=4 AND ACCUM[0],[6:24]="CARD"
    THEN FAROUT = TRUE;
  IF SCAN ≠ EQUAL THEN GO TO ERROR;
  IF SCAN < IDENT THEN GO TO ERROR;
  EQN[2] = EQN[3];%
  EQN[0]=0; EQN[1] = ACCUM[0];%
  IF (TYPE+SCAN)= SLASH THEN%
  BEGIN IF SCAN≥IDENT THEN%
    BEGIN EQN[0]=EQN[1]; EQN[1]=ACCUM[0] ;%
    ; END ELSE GO TO ERROR;%
    TYPE = SCAN END;%
  IF TYPE = COMMA THEN%
    BEGIN%
  IF (TYPE+SCAN)≠ IDENT OR KOUNT >3 THEN GO TO ERROR;%
  STREAM ( S + 3-KOUNT,KOUNT,ACCUM, T+[EQN[2]]);%
  BEGIN SI+ACCUM; SI+SI+1; DI+DI+S; DS+KOUNT NUM;%
  END;%
  IF (TYPE+SCAN)= COMMA THEN%
    BEGIN%
  IF (TYPE+SCAN)≠ IDENT OR KOUNT>5 THEN GO TO ERROR;%
  STREAM( S+8-KOUNT,KOUNT,ACCUM, T+[EQN[2]]);%
  BEGIN SI+ACCUM; SI+SI+1; DI+DI+S; DS+KOUNT NUM;%
  END;%
  IF (TYPE+SCAN)= COMMA THEN%
    BEGIN%
  IF (TYPE+SCAN)≠IDENT OR KOUNT>1 THEN GO TO ERROR;
  STREAM(S+1-KOUNT,KOUNT,ACCUM,T+[EQN[3]]);
  BEGIN SI+ACCUM; SI+SI+1; DI+DI+S; DS+KOUNT NUM;%
  END; TYPE = SCAN;%
  END% CYCLE ;%
  END% CREATION DATE ;%
  END;%REEL NUMBER;%
  TPNO=@37;%
  NOLBL = 0;%
ROUND;%
  WHILE TYPE≠PERIO AND (TYPE LSS FORM OR TYPE GTR FREEF) DO
    TYPE=SCAN;
  IF TYPE = PERIO THEN GO TO EXIT;%
  GO TO D[TYPE=FORM];%
NEXT; TYPE=SCAN; GO TO ROUND;%

```

```

20397300
20397400
20398000
20399000
20400000
20401000
20402000
20403000
20404000
20405000
20406000
20408000
20409000
20410000
% (SHM)20410100
20411000
20412000
20413000
20414000
20415000
20415100
20415200
20416000
20416500
20417000
20418000
20419000
20420000
20421000
20422000
20423000
20424000
20425000
20426000
20427000
20428000
20429000
20430000
20431000
20432000
20433000
20434000
20435000
20436000
20437000
20438000
20439000
20440000
20441000
20442000
20443000
20444000
20445000
20446000
20447000
20448000
20448100
20449000
20450000
20451000

```

LFORM: %		20452000
	EQN[3],[42:1]+1; GO TO NEXT; %	20453000
LNO: %		20454000
	NOLBL + 1; GO TO NEXT; %	20455000
LDISK: %		20456000
	TPNO:=10; GO TO DSKCHECK; % (SHM)	20457000
LTAPE: %		20458000
	TPNO + 2; GO TO NEXT; %	20459000
LPUNCH: %		20460000
	TPNO:=0;	20460100
	IF (TYPE:=SCAN)=PERIO THEN GO TO EXIT;	20461000
	IF TYPE=FREEF THEN GO TO LFREE ELSE	20461050
	IF TYPE=BACK THEN	20461100
	TPNO+20 ELSE	20461200
	BEGIN TPNO+21; IF SCAN#BACK THEN GO ERROR END;	20461300
	IF SCAN=PERIO THEN GO ERROR;	20461400
	IF (TYPE#SCAN)=PERIO THEN	20461500
	TPNO+TPNO+4 ELSE	20461600
	IF TYPE=FREEF THEN GO TO LFREE ELSE	20461650
	IF TYPE=DISK THEN	20461700
	TPNO+TPNO+2 ELSE	20461800
	IF TYPE#TAPE THEN GO ERROR;	20461900
	IF TYPE#PERIO THEN GO NEXT ELSE GO EXIT;	20461950
LPAPER: %		20462000
	TYPE + SCAN; TPNO + 7; GO TO NEXT; %	20463000
LSPECIAL: %		20464000
	TPNO + 3; GO TO NEXT; %	20465000
LPRINT: %		20466000
	TPNO:=1;	20466100
	IF (TYPE:=SCAN)=PERIO THEN GO TO EXIT;	20467000
	IF TYPE=FREEF THEN GO TO LFREE ELSE	20467100
	IF TYPE=BACK THEN	%P 20468000
LBACK: %		%P 20469000
	TPNO+6 ELSE	%P 20470000
	BEGIN TPNO+4; IF SCAN#BACK THEN GO ERROR END;	%P 20471000
	IF SCAN=PERIO THEN GO ERROR;	%P 20472000
	IF (TYPE#SCAN)=PERIO THEN	%P 20473000
	TPNO+22-TPNO ELSE	%P 20473100
	IF TYPE=FREEF THEN GO TO LFREE ELSE	%P 20474000
	IF TYPE=DISK THEN	%P 20475000
	TPNO+21-TPNO ELSE	%P 20476000
	IF TYPE#TAPE THEN GO ERROR;	%P 20477000
	IF TYPE #PERIO THEN GO NEXT ELSE GO EXIT;	%P 20478000
%		20478500
LFREE:		20478504
\$ SET OMIT = NOT(PACKETS)		20478505
	EQN[3],[23:1]+1;	20478506
\$ POP OMIT		20478508
	GO TO NEXT;	20478510
LCOPY:		20478520
	IF (TYPE:=SCAN) NEQ IDENT OR KOUNT GTR 3 THEN GO TO ERROR;	20478530
	STREAM(A:=0;KOUNT,ACCUM);	20478540
	BEGIN SI:=ACCUM;SI:=SI+1;DI:=LOC A;DS:=KOUNT OCT END;	20478550
	IF (TYPE:=P(DUP)) GTR 256 OR P(XCH)LSS 1 THEN GO ERROR;	20479000
	EQN[3],[15:8]:=TYPE-1;GO TO NEXT;	20480000
ERROR: %		20481000
	PPC#TRUE;GO DOWN; %	20482000
SPO:		20483000
	TPNO+11;GO TO NEXT; %	20483100
SERIAL:		% (SHM)20484000
	TPNO:=12; GO TO DSKCHECK;	
UPDATE:		
	TPNO+13; GO TO DSKCHECK;	
PROTECT:		
	TPNO+26;	
DSKCHECK:		% (SHM)20484000

```

IF (TYPE=SCAN)=COMMA THEN GO TO DSKCHECK; % (SHM)20484050
IF TYPE=EU THEN % (SHM)20484100
BEGIN % (SHM)20484150
IF SCAN NEQ EQUAL THEN GO TO ERROR ELSE % (SHM)20484200
IF (TYPE=SCAN) NEQ IDENT OR KOUNT GTR 2 THEN GO ERROR; % (SHM)20484250
STREAM(KOUNT,ACCUM,TI=[TYPE]); % (SHM)20484300
BEGIN % (SHM)20484350
SI=ACCUM; SI=SI+1; DI=T; DS=KOUNT OCT; % (SHM)20484400
END; % (SHM)20484450
EQN[12],[18:5]=TYPE+1; % (SHM)20484500
GO TO DSKCHECK; % (SHM)20484550
END % IF EU % (SHM)20484600
ELSE IF TYPE=FAST OR TYPE=SLOW THEN % (SHM)20484650
BEGIN % (SHM)20484700
EQN[12],[16:2]=1+(TYPE=SLOW); % (SHM)20484750
GO TO DSKCHECK; % (SHM)20484800
END % (SHM)20484850
ELSE IF TYPE = SENSE THEN % (SHM)20484855
BEGIN % (SHM)20484860
EQN[12],[15:1]=1; % (SHM)20484865
GO TO DSKCHECK; % (SHM)20484870
END; % (SHM)20484875
GO TO ROUND; % (SHM)20484900
EXIT;% 20485000
IF NOLBL THEN TPNO ← IF TPNO=2 THEN 9 ELSE% 20486000
(IF TPNO =3 THEN 5 ELSE% 20487000
(IF TPNO=7 THEN 8 ELSE% 20488000
(IF TPNO=037 THEN 9 ELSE TPNO)))%; 20489000
IF FAROUT THEN IF UNITNO≥32 THEN CIDROW[UNITNO=32],[3:5] ← 0 20489100
ELSE IF UNITNO=23 THEN READERA,[FF] ← 0 20489200
ELSE IF UNITNO=24 THEN READERB,[FF] ← 0; 20489300
EQN[3],[43:5]=TPNO;% 20490000
DEX ← DEX+1;% 20491000
END;% 20492000
ELSE% 20493000
BEGIN% 20494000
DO UNTIL (IOD + SCAN) = EQUAL OR IOD = PERIO;% 20495000
IF IOD = PERIO THEN GO TO ERROR;% 20496000
IOD + SCAN;% 20497000
STREAM (K←0; A ← [ACCUM[0]],KOUNT);% 20498000
BEGIN% 20499000
SI ← A ; SI=SI+1; DI←LOC K;% 20500000
KOUNT(IF SC<"0" THEN BEGIN DS←LIT"+"; 20500100
JUMP OUT TO ERR; END; SI=SI+1); 20500200
SI=SI+KOUNT; 20500300
DS ← KOUNT OCT ;% 20501000
ERR; 20501100
END;% 20502000
IF (TPNO←P),[1:1] THEN GO TO ERROR; 20503000
IF TYPE=PROCE OR TYPE=IO THEN X[16+TYPE=PROCE]=TPNO×3600 20504000
ELSE IF TYPE=COREV THEN X[20] ← TPNO DIV 64 20504500
ELSE IF TYPE≥PRIOR AND TYPE≤SAVEV THEN X[18+TYPE=PRIOR]=TPNO; 20505000
IF TYPE = COREV THEN 20506000
BEGIN DO UNTIL (IOD=SCAN)=MAXV OR IOD=PERIO; 20507000
IF IOD=MAXV THEN P([X[20]],IOR) ELSE GO TO DOWN; 20507100
END; 20507200
DO UNTIL SCAN = PERIO;% 20507300
END;% 20508000
DOWN;% 20509000
DOWN;% 20510000

```

```

        END;%
PROCEDURE SECURITYMAINT( TYPE,SMID,SFID,CMM,SFH,CARD);
VALUE TYPE,SMID,SFID,SFH,CARD;
REAL TYPE,SMID,SFID,SFH,CARD;
ARRAY CMM[*];
BEGIN
%
REAL N4,OPTN,T1;
REAL T=TYPE;
LABEL SEC3,FUNCO,FUNC1,FUNC2,FUNC3,SEC4,EXYT;
LABEL ERR,ERROR,FUNCJ;%
SWITCH FUNC←FUNCJ,FUNCO,FUNC1,FUNC2,FUNC3;%
LABEL EXIT;%
    N4:= ABS(CMM[5]);
    IF ((CMM[0]EQV "DECK ")=NOT 0) AND
        (((CMM[1]AND @77000000007777)EQV @12000000003714)=NOT 0)
    OR SYSTEMFILE(CMM[0],CMM[1]) THEN GO TO ERROR;
    IF TYPE = USEV AND
        ((CMM[0]EQV SMID)=NOT 0 AND (CMM[1]EQV SFID)=NOT 0) THEN
    ELSE
    IF (OPTN:=DIRECTORYSEARCH(CMM[0],CMM[1],4)) GEQ 64 THEN
    BEGIN
    IF TYPE=USEV AND M[OPTN+2]<0 THEN GO TO ERR;
    IF (T1+((N4 EQV MCP)=NOT 0) OR (CMM[5]=NOT 0)) OR
        (M[OPTN+2]>0 AND(N4 EQV ABS(M[OPTN+2]))=NOT 0)THEN
    GO TO SEC3 ELSE
    BEGIN ERR; FORGETSPACE(OPTN);
        FORGETSPACE(DIRECTORYSEARCH(CMM[0],CMM[1],14));
    END;
    END;
ERROR:
    STREAM(A:=[CMM[0]],B:=(OPTN:=SPACE(10)));
    BEGIN SI:=A; SI:=SI+1; DS:=LIT" "; DS:=7 CHR;
        SI:=SI+1; DS:=LIT"/"; DS:=7 CHR;
        DS:=24 LIT " SECURITY MAINT IGNORED*";
    END STREAM;
    SPOUTER(OPTN,CARD,1);
    GO TO EXYT;
SEC3:
%
    GO TO FUNC[TYPE=UNLOCKV];
FUNCJ: M[OPTN INX 5]←M[OPTN INX 6]←@14;%
    CMM[2] := " UNLOCK"; CMM[3] := "ED←+ ";%
    GO TO SEC4;%
FUNCO:
    M[OPTN INX 5] := SMID; M[OPTN INX 6] := SFID;
    CMM[2] := " SECURE"; CMM[3] := "D WITH ";
    M[SFH+2] := P(DUP,LOD,SSB);
    GO TO SEC4;
FUNC1:
    IF (T1←T1 AND (M[OPTN+2]=0)) THEN M[OPTN+2]←CMM[6];
    SMID:=M[OPTN+5]; SFID:=M[OPTN+6];
    M[OPTN INX 5] := M[OPTN INX 6] := 0;
    CMM[2]←" LOCKED";CMM[3]←" FROM ";CMM[4]←" WITH ";GO TO SEC4;
FUNC2:
    M[OPTN INX 2] := M[OPTN INX 5] := M[OPTN INX 6] := 0;
    CMM[2] := " FREE F"; CMM[3] := "ILE←+ "; GO TO SEC4;
FUNC3:
    M[OPTN INX 5] := @14; M[OPTN INX 6] := 0;
    CMM[2] := " PUBLIC";CMM[3] := " FILE←+";

```

```

20511000
20511100
20511110
20511120
20511130
20511140
20511145
20511150
20511155
20511160
20511165
20511170
20511171
20511181
20511182
20511184
20511188
20511295
20511300
20511305
20511306
20511311
20511312
20511315
20511320
20511330
20511340
20511350
20511360
20511363
20511365
20511370
20511380
20511390
20511400
20511410
20511420
20511430
20511440
20511445
20511450
20511455
20511457
20511459
20511460
20511470
20511480
20511490
20511500
20511510
20511515
20511520
20511525
20511530
20511540
20511550
20511560
20511570
20511580
20511590

```

SEC4:	20511600
DISKWAIT(OPTN,[CF],30,OPTN,[FF]);	20511610
P(DIRECTORYSEARCH(=CMM[0],CMM[1],14),DEL);	20511620
STREAM(AI=ABS(SMID),BI=SFID,CI=CMM,QI=(T LSS FREE) &	20511660
AND (T#UNLOCKV) AND (ABS(SMID)#12),	20511662
XI=(SFID=0 OR ABS(SFID)=12) %	20511663
AND T LSS FREE AND T#UNLOCKV,	20511664
Y+T=LOCKV AND(((N4 EQV MCP)=NOT 0)AND((CMM[6] EQV MCP)#	20511665
NOT 0)) AND T1.D+OPTN+OPTN INX 0);	20511666
BEGIN SI=C; SI=SI+1; DS=LIT" "; DS=7 CHR; DS=LIT"/";	20511670
3(SI=SI+1; DS=7 CHR);	20511680
X(DI=DI-7; DS=2 LIT"+");	20511685
Q(DS=LIT" ";SI=LOC A;SI=SI+1;DS=7 CHR; DS=LIT"/";	20511690
SI+SI+1; DS+7 CHR);	20511700
Y(X(DI+DI=18); SI=C;4(SI+SI+8);SI+SI+1;DS+7 CHR;	20511702
SI+SI+9; DS+7 CHR); DS+ LIT "+";	20511704
END STREAM;	20511710
SPOUTER(OPTN,CARD,SECMSG);	20511720
EXYT;	20511800
END SECURITYMAINT;	20511810
COMMENT THE PRT CELL "SHEET" GIVES DISK ADDRESS OF 1ST SHEET ENTRY	20512000
*** ENTRIES IN THE SHEET ARE AS FOLLOWS:	20512400
S[ 0] = 1ST NAME (7 CHRS)	20512800
. [ 2:1 ] = "CANDE" JOB (TSS ONLY)	20513200
S[ 1] = 2ND NAME (7 CHRS)	20513600
S[ 2],[ 1: 2] = 0 NORMAL	20514000
2 JOB HAS BEEN XS=ED (FORCED RUN)	20514400
3 JOB HAS BEEN ES=ED (FORCED RUN AND DS)	20514800
S[ 2],[ 4:1 ] = SUPPRESS BOJ/EOJ MESSAGES FOR SYSTEM JOBS	20515200
S[ 2],[ 8:10] = 0 GO JOB (FROM COMPILE & GO)	20515600
= 1 COMPILER (FOR COMPILE & GO)	20516000
= 2 EXECUTE JOB	20516400
= 3 COMPILER (FOR SYNTAX CHECK)(SET TO 2 LATER)	20516800
= 4 COMPILER (FOR COMPILE TO LIBRARY)	20517200
= 5 RUN JOB	20517600
S[ 2],[ 18:15] = SKELETONS DISK ADDRESS (IF S[2],[ 8:10] = 1,2,4	20518000
S[ 2],[ 33:15] = PRIORITY, SAME AS S[18]	20518400
S[ 3],[ 1:1 ] = SET BY SELECTRUN WHEN "SCHEDULED" MESSAGE	20518800
IS SENT (IF SCHEDULED)	20519200
S[ 3],[ 2: 1] = 1 RESTART JOB	20519600
S[ 3],[ 8:10] = SCHEDULE=ID FOR THIS JOB	20520000
S[ 5] = STARTING TIME FOR LOG	20520400
S[ 6] = LOCATION OF LAST PART OF LOG	20520800
S[ 7] = CORE ADDRESS OF SEGMENT ZERO (WHEN THE	20521200
SHEET IS PASSED TO SELECTRUN AS A PARAMETER)	20521600
S[13] = DISK ADDRESS OF LABEL EQUATION ENTRIES	20522000
APPLICABLE TO THIS EXECUTION ONLY (SEE BELOW)	20522400
S[14] = ACTUAL MFID OF JOB (TSS ONLY), THIS MAY BE	20522800
BE DIFFERENT FROM S[0] FOR SOME JOBS	20523200
WHICH ARE STARTED BY CANDE,	20523600
S[15] = DISK ADDRESS OF LABEL EQUATION ENTRIES	20524000
PRESENTED WHEN PROGRAM WAS COMPILED AND	20524400
APPLICABLE TO ALL EXECUTIONS	20524800
S[16] = ESTIMATED PROCESSOR TIME	20525200
S[17] = ESTIMATED I/O TIME	20525600
S[18] = PRIORITY	20526000
S[19] = COMMON VALUE	20526400
S[20] = ESTIMATED CORE REQUIREMENTS	20526800
S[20],[ 2:1 ] = "CAN=T EXPAND" BIT (TSS)	20527200
. [33:15] = ESTIMATED CORE REQUIREMENT	20527600



S[21]	= STACK SIZE	20528000
S[22]	= SAVE FACTOR FOR OBJECT FILE (COMPILATIONS)	20528400
S[23],[2:6]	= UNITNO OF CARD/PSEUDO READER IN CONTROLCARD.	20528800
S[23],[9:19]	= REMOTE STATION ADDRESS, ELSE 0	20529200
S[23],[24:24]	= TIME JOB PUT IN SHEET(FOR TS MSG)	20529600
S[24]	= USER CODE	20530000
S[25]	= DISK ADDRESS OF FILE HEADER FOR THE JOB	20530400
S[26]	= LOGLINE (TSS)	20530800
S[27]	= FID FOR COMPILES,TAPE NAME FOR LIBMAIN.	20531200
S[29]	= DISK ADDRESS OF NEXT SHEET ENTRY (=0 IF LAST)	20531600
*** ENTRIES FOR LABEL EQAT, ARE AS FOLLOWS:		20532000
F[0]	= MULTI-FILE ID (7 CHRS)	20532400
F[1]	= FILE ID (7 CHRS)	20532800
F[2],[0:18]	= REEL NO (3 CHRS)	20533200
F[2],[18:30]	= CREATION DATE (5 CHRS)	20533600
F[3],[0:6]	= CYCLE (1 CHR)	20534000
F[3],[15:18]	= NUM COPIES OF PBD OR PUD FILE	20534400
F[3],[23:11]	= 1, IF "FREEF" PBD PACKET FILE	20534800
F[3],[42:11]	= 1 FOR FORMS REQUIRED	20535200
F[3],[43:5]	= 0 FOR CP (FILE TYPES )	20535600
	1 FOR LP	20536000
	2 FOR MT	20536400
	3 FOR SPECIFIC UNIT	20536800
	4 FOR LP (MAY BACKUP)	20537200
	5 FOR SPECIFIC (UNLABLED)	20537600
	6 FOR LP (MUST BACKUP)	20538000
	7 FOR PT	20538400
	8 FOR PT (UNLABLED)	20538800
	9 FOR MT (UNLABLED)	20539200
	10 FOR DISK	20539600
F[4],[0:6]	= NO OF CHARS IN INTERNAL NAME	20540000
F[4],[6:42]	= INTERNAL NAME (MAY CONTINUE TO F[11])	20540400
F[12],[15:1]	= "SENSITIVE" BIT	20540800
F[12],[16:2]	= DISK SPEED	20541200
F[12],[18:5]	= EU NUMBER + 1	20541600
F[14]- F[25]	SAME AS ABOVE FOR NEXT FILE (F[14]=14 IF NO NEXT)	20542000
F[29]	= DISK ADRS.OF NXT.LBL,EQUAT.ENTRY(=0 IF NONE)	20542400
**** ALSO SEE PROCEDURE "SELECTRUN1" (SEQ.NO,20055600) FOR		20542800
**** FURTHER INFORMATION ON LABEL EQUATION AND THE FILE		20543200
**** PARAMETER BLOCK,		20543600
		20544000
**** CONTENTS OF THE JAR:		20544400
JAR[0],[ 1:1 ]	= COMPILE JOB	20544800
,[ 2:1 ]	= "CANDE" JOB (TSS ONLY)	20545200
,[ 6:42]	= MFID OF THE JOB	20545600
JAR[1],[ 1:1 ]	= JOB IS BEING DS=ED	20546000
,[ 2:1 ]	= JOB IS BEING ES=ED	20546400
,[ 6:42]	= FID OF THE JOB	20546800
JAR[2],[ 1:1 ]	= COBOL JOB	20547200
,[ 2:1 ]	= DECLARED SOFTWARE INTERRUPTS	20547600
,[ 3:1 ]	= JOB HAS MAINTENANCE LOG ENTRY	20548000
,[ 4:1 ]	= INTER=PROGRAM COMMUNICATION	20548400
,[ 5:1 ]	= DECLARED SOFTWARE INTERRUPTS	20548800
,[ 6:1 ]	= INVOKED OR INVOKING IPC PROG,FILE	20549200
,[ 7:1 ]	= INVOKED IPC PROGRAM FILE	20549600
,[ 8:1 ]	= INTER=PROGRAM COMMUNICATION	20550000
,[18:15]	= DISK ADDRESS FOR THE SKELETON SHEET (COMPILATIONS)	20550400
,[33:15]	= PRIORITY	20550800
JAR[3]	= PROCESS TIME LIMIT	20551200
JAR[4]	= IO TIME LIMIT	20551600

JAR[5],[ 1:23]	=	STARTING DATE (OCTAL)	20552000
,[24:24]	=	STARTING TIME (OCTAL)	20552400
JAR[6],[ 1:1 ]	=	JOB IS SD=ED	20552800
,[ 2:6 ]	=	PSEUDO-READER NUMBER	20553200
,[18:15]	=	SIZE OF LOG INFORMATION (BATCH)	20553600
,[33:15]	=	DISK ADDRESS OF FIRST RECORD FOR THE LOG	20554000
JAR[7]	=	IDLETIME ENTRY (BATCH)	20554400
JAR[7]	=	MFID OF JOB (TSS ONLY), THIS MAY BE DIFFERENT	20554800
		FROM JAR[0] FOR SOME JOBS STARTED BY CANDE,	20555200
JAR[8]	=	LENGTH OF CODE FILE ROW	20555600
JAR[9],[ 1:1 ]	=	"SYSTEM" JOB (LIBMAIN,LDCTRL,PRNPBT)	20556000
,[ 2:1 ]	=	SUPPRESS PRINTING OF BOJ/EOJ MESSAGES	20556400
,[18:15]	=	DISK ADDRESS FOR "CHAIN" IF NON=ZERO	20556800
,[33:15]	=	NUMBER FOR DISK ROWS IN CODE FILE	20557200
JAR[10] THROUGH JAR[29]	=	DISK ADDRESS OF CODE FILE ROWS	20557600
JAR[30]	=	FID OF OBJECT FILE (BATCH COMPILES ONLY)	20558000
END OF COMMENT;			20558400
REAL PROCEDURE LIBCC;			20566000
BEGIN LABEL NEXT,LOOP;			20566011
REAL	CNTSENS	= RETURNVAL+1, % BEGIN LOCALS TO LIBCC	20566245
	HOLD1	= CNTSENS+1,	20566247
	HOLD2	= HOLD1+1,	20566250
	HOLD3	= HOLD2+1,	20566255
	REPEAT	= HOLD3+1,	20566260
	TYM	= REPEAT+1;	20566265
BOOLEAN	FIRSTIME	= TYM+1;	20566270
	LABEL	CCA,QUIT,POWIE,CHAN,REMO,INCSC,GETEM,ENTE,LCOPY,SEEK,INIT;	20566600
	LABEL	DOWNR,OUTR;	20566610
	SWITCH	SW=LCOPY,LCOPY,ENTE,ENTE,REMO,CHAN;	20566700
	DEFINE	ZIPMIX=CARD,[18:6]#;	20566740
	SUBROUTINE	BOTH;	20566750
	BEGIN	CMM[0]="LIBMAIN"; CMM[1]="DISK ";	20566755
		CMM[2]=0&2[8:38:10]; CMM[13]=0;	20566760
\$	SET OMIT = PACKETS		20566765
		CMM[23]=0&CARD[9:9:9];	20566770
\$	POP OMIT		20566775
\$	SET OMIT = NOT(PACKETS)		20566780
		CMM[23]=0&CARD[9:9:9]&(IF ZIPMIX NEQ 0 THEN PSEUDOMIX[ZIPMIX]	20566785
		ELSE UNITNO)[2:42:6];	20566790
\$	POP OMIT		20566795
	END OF BOTH;		20566797
	SUBROUTINE	GETEMFORREM; %STORE NAMES OF SENSITIVE FILES IN ESPDISK	20566800
	BEGIN	CNTSENS:=CNTSENS+2;	20566805
		IF CNTSENS GTR 26	20566810
		THEN BEGIN PROG[29]=GETESPDISK;	20566815
		DISKWAIT(PROG INX 0,30,LIBNO);	20566820
		LIBNO:=PROG[29];	20566825
		CNTSENS:=2;	20566830
		END;	20566835
		PROG[CNTSENS]=CMM[2];	20566840
		PROG[CNTSENS+1]=CMM[3];	20566845
	END OF GETEMFORREM;		20566850
	REAL	SUBROUTINE SCAN;	20566875
		SCAN=SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,	20566900
		DIRECT);	20566902
	REAL	SUBROUTINE SKAN;	20566905
	BEGIN		20566910
		STREAM(X:=0;CNI=0,ACCUM);	20566915
	BEGIN		20566920
		SI:=ACCUM;SI:=SI+1;	20566925

```

8(IF SC GEQ "0" THEN BEGIN SII:=SI+1;TALLY:=TALLY+1; END ELSE
  IF SC=" " THEN JUMP OUT ELSE BEGIN TALLY:=0;JUMP OUT END);
  CN:=TALLY;SII:=SI-CN;DII:=LOC X;DS:=CN OCT;
END;
SKAN:=P;
END OF SKAN;
P(RCW,MYMSCW,STF);
RCW:=RCW & P(XCH)[CTC];
P(0,0,0,0,0,0,0); % ZERO LOCALS OF LIBCC
LIBCC:=0;
FIRSTIME:=TRUE;
UNITNO+(CARD,[3:5]+28),[43:5]+4;
CARDLOC+CARD INX 0;
GO TO SW[T-COPY];
LCOPY;
ENTE;
  IF (CN:=SCAN)=IDENT THEN
  BEGIN;
    IF (PROG[0]:=SKAN)=0 THEN PROG[0]:=511 ELSE CN:=SCAN;
    IF PROG[0]>511 THEN PROG[0]:=511;
  END ELSE PROG[0]:=511;
  CMM[19]:=0;
$ SET OMIT = NOT(B6500LOAD)
  IF CN=B6500 THEN BEGIN CMM[19],[15:1]:=1;CN:=SCAN END;
$ POP OMIT
  IF CN=TOV AND T GTR UNLOAD THEN
  BEGIN
    IF (CN:=SCAN)=EU THEN
    BEGIN
      IF (CN:=SCAN)≠IDENT THEN GO TO INCSC;
      IF P(SKAN,DUP)>19 THEN BEGIN P(DEL);GO INCSC END;
      CN+P+1;CMM[19],[9:6]+CN;
      IF CN>NEUP,NEUF THEN GO TO INCSC;
    END
    ELSE IF CN=SLOW THEN CMM[19]+P(DUP,LOD) OR M ELSE
      IF CN=FAST THEN CMM[19],[3:1]+1 ELSE GO TO INCSC;
    CN:=SCAN;
  END;
  IF CN=LATESTV AND T GTR UNLOAD THEN
  BEGIN
    CN:=SCAN;
    CMM[19],[4:1]:=1;
  END;
  IF CN=EXPIRED THEN
  BEGIN
    PROG[0],[9:1]:=1;
    CN:=SCAN;
  END;
  IF CN=ACCESSD THEN
  BEGIN
    PROG[0],[8:1]:=1;
    CN:=SCAN;
  END;
  IF UNITNO=23 OR UNITNO=24 OR UNITNO≥32 THEN
    PROG[0],[2:6]+UNITNO;
  IF SCAN≠IDENT THEN GO TO INCSC;
  PROG[1]:=CMM[27];=ACCUM[0];
  PROG[28]+0;
  BOTH;
  LIBNO+GETESPDISK;CMM[19],[CF]+LIBNO;

```

```

20566930
20566935
20566940
20566945
20566950
20566955
20567000
20567010
20567015
20567020
20567025
20567100
20567200
20567300
20567400
20567500
20567600
20567700
20567800
20567900
20568000
20568050
20568059
20568060
20568061
20568100
20568200
20568300
20568400
20568500
20568600
%022-20568610
20568611
20568620
20568630
20568640
20568650
20568700
20568710
20568712
20568714
20568716
20568718
20568728
20568729
20568730
20568740
20568750
20568760
20568770
20568780
20568790
20568795
20568900
20569000
20569100
20569200
20569250
20569300
20569800

```

CMM[19],[FF]+T;	20569810
GETEM: % SCAN FILE NAMES AND STORE THEM IN ESPDISK	20569900
FOR CN+2 STEP 2 UNTIL 26 DO	20570000
BEGIN	20570100
IF (OPTN+SCAN)=EQUAL THEN PROG[CN]+*1	20570200
ELSE IF OPTN>=IDENT THEN PROG[CN]+ACCUM[0] ELSE GO POWIE;	20570300
IF SCAN#SLASH THEN GO POWIE;	20570400
IF (OPTN+SCAN)=EQUAL THEN PROG[CN+1]+*1	20570500
ELSE IF OPTN>=IDENT THEN PROG[CN+1]+ACCUM[0] ELSE GO POWIE;	20570600
IF (OPTN+SCAN)=PERIO OR OPTN#POUND THEN GO TO QUIT%LP 1	20570700
ELSE IF OPTN#COMMA THEN GO POWIE;	20570800
END;	20570900
PROG[29]+GETESPDISK;	20571000
DISKWAIT(PROG INX 0,30,LIBNO);	20571100
LIBNO+PROG[29];	20571300
GO GETEM;	20571400
QUIT:	20571500
PROG[29]+0;	20571600
PROG[CN+2]+@14;	20571700
DISKWAIT(PROG INX 0,30,LIBNO);	20571800
LIBNO+ABS(CMM[19]);	20572000
GO INIT;	20572100
POWIE:	20572200
IF CMM[19],[CF]#LIBNO THEN % MORE THAN ONE SEGMENT USED	20572300
BEGIN	20572400
DISKWAIT(=PROG,[CF],30,CMM[19],[CF]);	20572500
FORGETESPDISK(CMM[19],[CF]);	20572700
CMM[19]+PROG[29];	20572800
GO POWIE;	20572900
END;	20573000
FORGETESPDISK(LIBNO);	20573100
GO INCSC;	20573200
REMO:	20573300
IF (CN+SCAN)=EQUAL THEN CMM[0]+*1 ELSE	20573400
IF CN>=IDENT THEN CMM[0]+ACCUM[0] ELSE GO INCSC;	20573500
IF SCAN#SLASH THEN GO INCSC;	20573600
IF (CN+SCAN)=EQUAL THEN CMM[1]+*1 ELSE	20573700
IF CN>=IDENT THEN CMM[1]+ACCUM[0] ELSE GO INCSC;	20573800
CN:=T:=0;	20573850
IF (CMM[0] OR CMM[1]) LSS 0 THEN	20573900
SEEK:	20574000
SEEKNAM(CMM[0],CMM[1],CN,CMM[2],CMM[3],OPTN) ELSE	20574100
BEGIN	20574200
CMM[2]:=CMM[0];	20574300
CMM[3]:=CMM[1];	20574400
CN:=1;	20574500
END;	20574600
IF CN NEQ 0	20574700
THEN T:=IF SYSTEMFILE(CMM[2],CMM[3])	20574750
THEN 2	20574800
ELSE DIRECTORYSEARCH(CMM[2],CMM[3],5)	20574850
ELSE IF OPTN NEQ 0 THEN GO OUTR;	20574875
IF T GEQ 64 THEN	20574900
BEGIN IF HOLD3:=NOT(M[T+4],[44:1]) THEN BEGIN FORGETSPACE(T);	20574905
T:=DIRECTORYSEARCH(CMM[2],CMM[3]&(UNITNO=25 OR UNITNO=30)	20574910
[1:47:1],4); END;	20574915
IF M[T+4],[43:2]=3 THEN BEGIN FORGETSPACE(T); T:=1; END;	20574920
END;	20574922
IF CARD,[8:1] THEN GO DOWNR;	20574925
IF T LSS 2	20574950

```

THEN IF T=1                                20575000
  THEN LBMESS(ABS(CMM[2]),CMM[3],-7,45,0,SPOUTUNIT,1) 20575050
  ELSE LBMESS(CMM[0],CMM[1],-7,15,0,SPOUTUNIT,1) 20575100
ELSE IF T=2                                20575150
  THEN LBMESS(CMM[2],CMM[3],-7,25,0,SPOUTUNIT,1) 20575200
  ELSE IF T GEQ 64                          20575250
    THEN BEGIN                              20575300
      IF M[T+2] NEQ 0 AND (USERID EQV MCP) NEQ 20575350
        NOT 0 AND (USERID EQV ABS(M[T+2])) NEQ 20575400
        NOT 0                                20575450
      THEN BEGIN                             20575500
        LBMESS(CMM[2],CMM[3],-7,41,          20575550
          0,SPOUTUNIT,1);                    20575600
        FORGETSPACE(DIRECTORYSEARCH(CMM[2], 20575650
          CMM[3],14));                       20575700
      END                                     20575750
      ELSE IF M[T+4],[43:2] NEQ 0           20575800
        THEN BEGIN                           20575850
          IF FIRSTIME                        20575900
            THEN BEGIN                       20575950
              FIRSTIME:=0;                  20576000
              CMM[19]:=(LIBNO:=           20576050
                GETESPDISK)&36[CF]);        20576100
              END;                          20576150
              M[T+4],[43:2]:=1;             20576200
              DISKWAIT(T,[CF],30,T,[FF]);  20576210
              IF HOLD3 THEN FORGETSPACE(   20576215
                DIRECTORYSEARCH(CMM[2],
                CMM[3],14));                20576216
              GETEMFORREM;                  20576220
              END                           20576250
            ELSE FORGETSPACE(DIRECTORYSEARCH(20576300
              CMM[2],CMM[3],6              20576350
              &SPOUTUNIT[9:9:9]            20576400
              &SPOUTUNIT[24:42:6]         20576405
              ));                           20576410
          FORGETSPACE(T);                   20576415
          END;                              20576450
        IF CN NEQ 0 AND (CMM[0] OR CMM[1]) LSS 0 THEN GO SEEK; 20576500
        IF (CN:=SCAN)=COMMA THEN GO REMO;  20576600
        IF CN=PERIO THEN                   20576700
          IF NOT FIRSTIME THEN             20576710
            BEGIN OPTN:=CN; PROG[29]:=0;   20576720
              PROG[CNSENS+2]:=14;         20576730
              DISKWAIT(PROG INX 0,30,LIBNO); 20576740
              LIBNO:=ABS(CMM[19]);        20576750
              BOTH;                        20576752
              GO INIT;                     20576770
            END                             20576780
          ELSE GO CCA                       20576790
        ELSE GO INQSC;                     20576800
      CHAN:                                20576850
        T:=0; % T USED AS BIT MASK FOR SYNTAX CHECK 20576900
        FOR CN:=0 STEP 1 UNTIL 3 DO % SCAN INPUT REQUEST 20576925
          BEGIN                             20576950
            OPTN := SCAN;                   20576975
            T := (OPTN=EQUAL) & T[43:44:4]; % SHIFT PREVIOUS VALUE LEFT 20577000
            IF T THEN CMM[CN] := (-1) ELSE 20577025

```

IF OPTN GEQ IDENT THEN CMM[CN] := ACCUM[0] ELSE	20577050
GO TO INCSC; % INCORRECT REQUEST	20577075
OPTN := SCAN; % SKIP "/" , " , " OR " ; "	20577100
END; % SCANNING INPUT REQUEST	20577125
IF (T NEQ 0) AND (T NEQ 5) AND (T NEQ 10) THEN GO INCSC;	20577150
% T=5 FOR #/<NAME1> TO #/<NAME2>	20577175
% T=10 FOR <NAME1>/# TO <NAME2>/#	20577200
% T=0 FOR <NAME1>/<NAME2> TO <NAME3>/<NAME4>	20577225
IF (REPEAT:=(T GTR 0)) THEN	20577250
BEGIN	20577275
HOLD1 := CMM[0]; HOLD2 := CMM[1]; TYM:=1; CN:=0;	20577300
SEEKNAM(HOLD1, HOLD2, CN, CMM[0], CMM[1], HOLD3);	20577325
IF CN = 0 THEN % NOT FOUND IN DIRECTORY	20577350
BEGIN	20577375
IF TYM = 1 THEN % FIRST PASS, NULL SEARCH	20577400
BEGIN	20577425
LBMESS(HOLD1, HOLD2, -5, 15, %NOT CHANGED, NOT ON DISK	20577450
0, SPOUTUNIT,1);	20577475
END;	20577500
GO TO NEXT;	20577525
END;	20577550
TYM := 2;	20577575
IF HOLD1 LSS 0 THEN CMM[2] := CMM[0] ELSE	20577600
IF HOLD2 LSS 0 THEN CMM[3] := CMM[1]; % USE NAME "FOUND"	20577625
END;	20577650
IF (T:=DIRECTORYSEARCH(CMM[2],CMM[3],5)) NEQ 0 THEN	20577675
BEGIN	20577700
FORGETSPACE(T);	20577725
LBMESS(CMM[0], CMM[1], -5, 29, % NOT CHANGED, DUP FILE	20577750
0, SPOUTUNIT, 1);	20577775
END ELSE	20577800
BEGIN	20577805
T:=IF SYSTEMFILE(CMM[0],CMM[1]) THEN 2 ELSE	20577810
DIRECTORYSEARCH(CMM[0],CMM[1],5);	20577815
IF T GEQ 64 THEN	20577820
BEGIN IF NOT(M[T+4],[44:1]) THEN BEGIN FORGETSPACE(T);	20577823
T:=DIRECTORYSEARCH(CMM[0],CMM[1]&(UNITNO=25))[1:47:1],	20577826
4); END;	20577827
IF M[T+4],[43:2]=3 THEN BEGIN FORGETSPACE(T); T:=1;	20577829
END;	20577832
END;	20577833
IF T LSS 2 THEN	20577835
LBMESS(CMM[0],CMM[1],-5,15+((T=1)*30), % NOT CHANGED	20577875
% 45 = IN USE, 15 = NOT ON DISK	20577900
0, SPOUTUNIT, 1)	20577925
ELSE IF T=2 THEN	20577950
LBMESS(CMM[0], CMM[1], -5, 25, % NOT CHANGED, SYSTEM FILE	20577975
0, SPOUTUNIT, 1)	20578000
ELSE IF M[T+2] NEQ 0 AND % NOT FREE FILE	20578025
(USERID EQV MCP) NEQ NOT 0 AND % NOT MCP	20578050
(USERID EQV ABS(M[T+2])) NEQ NOT 0 THEN % NOT CREATOR	20578075
BEGIN	20578100
LBMESS(CMM[0], CMM[1], -5, 41, % NOT CHANGED, INVALID USER	20578125
0, SPOUTUNIT, 1);	20578150
IF M[T+4],[43:2] NEQ 1 THEN	20578175
FORGETSPACE( DIRECTORYSEARCH(CMM[0], CMM[1], 14 ) );	20578200
FORGETSPACE(T);	20578210
END	20578225
ELSE	20578250
BEGIN % CHANGE OK	20578275

```

M[T+4]:=(*P(DUP))&2[1;46;2];
DISKLOG(CMM[0],CMM[1],IOQUE&T[CTC]);
T:=T&EUF(=CMM[2],CMM[3],T INX 0=1)[18;33;15];
FORGETSPACE(DIRECTORYSEARCH(CMM[0],CMM[1],8));
HEADERUNLOCK(CMM[2],CMM[3],T);
LBMESS(CMM[0],CMM[1],52,% CHANGED TO
      CMM[2],CMM[3],SPOUTUNIT,LIBMSG);
PBCOUNT:=PBCOUNT+((((CMM[0] EQV "PBD ")=NOT 0) OR
((CMM[0] EQV "PUD ")=NOT 0)) AND (CMM[1],[CF]=1))
+((((CMM[2] EQV "PBD ")=NOT 0) OR
((CMM[2] EQV "PUD ")=NOT 0)) AND (CMM[3],[CF]=1));
END;
END;
IF REPEAT THEN GO TO LOOP; % FIND REMAINING FILES
NEXT;
IF OPTN=COMMA THEN GO CHAN;
IF OPTN=PERIO THEN GO TO CCA ELSE GO INCSC;
NIT: LIBCC+LIBNO; GO TO CCA;
INCSC: LIBCC+1;
CCA: CADDR:=CDEX:=0;
IF (LIBNO:=PROCV) GTR 1 THEN PROCV:=2 ELSE
  IF LIBNO THEN PROCV:=6 ELSE PROCV:=0;
RETURNVAL:=PROCV;
P([RETURNRCW],STS,0,RDS,0,XCH,P&P[CTF],STF);
END; % LIBCC PROCEDURE
REAL PROCEDURE CCSET; FORWARD;
PROCEDURE CCFINISH;
BEGIN
REAL TEMP = RETURNRCW+1; % BEGIN LOCALS OF CCFINISH
P(RCW,MYMSCW,STF);
RCW:=RCW & P(XCH)[CTC];
P(0); % ZERO LOCAL OF CCFINISH
PPCPROCESS:=0;
CN:=T;
IF OPTN = PERIO OR OPTN = LIBRA THEN
BEGIN
CMM[22]:= PROG[22];
PROG[2],[CF]:= IF PROG[18] > 32767 THEN 32767
ELSE PROG[18];
IF PROG[20] > 512 THEN PROG[20]:= 512;
IF PADDR NEQ 0 THEN
BEGIN
PEQN[29]:= 0;
IF PDEX=0 THEN PEQN[0]:=14;
IF PDEX=1 THEN PEQN[14]:= 14;
DISKWAIT(PEQN,[CF],30,PADDR);
END;
PROG[29]:= 0;
CMM[2],[18;15]:= CLOSET:= GETESPDISK;
DISKWAIT(PROG,[CF],30,CLOSET);
END;
IF CADDR NEQ 0 THEN
BEGIN
CEQN[29]:= 0;
IF CDFX=0 THEN CEQN[0]:= 14;
IF CDEX=1 THEN CEQN[14]:= 14;
DISKWAIT(CEQN,[CF],30,CADDR);
END;
SLEEP([TOGGLE],SHEETMASK); LOCKTOG(SHEETMASK);
CDEX:= GETESPDISK;

```

```

20578300
20578325
20578375
20578400
20578425
20578525
20578550
20578575
20578600
20578625
20578650
20578675
20578685
20578700
20578725
20579900
20580000
20580100
20580200
20580300
20580305
20580310
20580330
20580340
20580350
20580400
20580800
20580852
20581080
20581125
20581130
20581140
20581150
20581200
20581250
20581300
20581350
20581400
20581450
20581500
20581550
20581600
20581650
20581700
20581750
20581800
20581850
20581900
20581950
20582000
20582050
20582100
20582150
20582200
20582250
20582300
20582350
20582400
20582450
20582500

```

```

CMM[2],[CF]:= IF CMM[18] > 32767 THEN 32767 ELSE CMM[18];
PDEX:=IF CMM[18]>SHEETMAX THEN SHEETMAX ELSE CMM[18];
IF LIBNO NEQ 0 THEN CMM[19]:= LIBNO;
IF CMM[20] > 512 THEN CMM[20] := 512;
STREAM(A:=0;S:=P(,SCHEDULE[DS]));
BEGIN S1:=S;
    47(SKIP SB; SKIP DB; TALLY:=TALLY+1;
    IF SB THEN BEGIN END ELSE JUMP OUT);
    DS:= SET; A:= TALLY;
END STREAM;
TEMP:= P; CMM[3]:= 0&TEMP[8;38;10];
CMM[23],[24;24]+(CLOCK+P(RTR))DIV 60;
IF SHEET[PDEX],[CF] NEQ 0 THEN
    BEGIN
    DISKWAIT(=PROG,[CF],30,PADDR:=SHEET[PDEX],[FF]);
    PROG[29]:= CDEX;
    DISKWAIT(PROG,[CF],30,PADDR);
    END
ELSE SHEET[PDEX]:= CDEX;
SHEET[PDEX],[18;15]:= CDEX;
CMM[29]:= 0;
DISKWAIT(CMM,[CF],30,CDEX);
UNLOCKTOG(SHEETMASK);
T:= CN;
P([RETURNRCW],STS,0,RDS,0,XCH,P&P[CTF],STF);
END CCFINISH;
REAL PROCEDURE CCCOMPILE;
BEGIN COMMENT SETUP OF COMPILER LABEL EQUATION CODE; PN1/PN2;
REAL SUBROUTINE SCAN;
    SCAN=SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,DIRECT);
LABEL SKN,EXIT;
DEFINE ZIPMIX=CARD,[18;6]#;
DEFINE DISKTYPE = 10#;%
    P(RCW,MYMSCW,STF);
    RCW:=RCW & P(XCH)[CTC];
    T:=SCAN;%
    CEQN[0]:=ACCUM[0];%
    T:=SCAN;%
    T:=SCAN;%
    CEQN[1]:=ACCUM[0];%
    CEQN[2]:=0;%
    CEQN[3]:=DISKTYPE;%
    CEQN[4]:=@423462425606060;%
    CEQN[12]:=0; % EU/SPEED CELL
    CDEX 1=1;%
    IF ((UNITNO+1)AND 24)=24 OR UNITNO GEQ 32 THEN%
    BEGIN CEQN[14]:=CEQN[16]:=CEQN[17]:=0;%
        CEQN[15]:= "CARD 00" OR ((IF UNITNO GEQ 32 THEN%
            "C/" ELSE @5772) + UNITNO);%
    CEQN[18]:=@423215124000000; CDEX:=2;%
    IF UNITNO GEQ 32 THEN CIDROW[UNITNO-32],[3;5]:=1 ELSE%
    IF UNITNO=23 THEN READERA,[FF] = 1 ELSE
    IF UNITNO=24 THEN READERB,[FF] = 1;
    END;
    WHILE (CN:=SCAN) LSS ALGOL OR CN GTR COBOL DO
    IF CN=PERIO THEN BEGIN CCCOMPILE:=1; GO EXIT END;
        IF CN=WITH THEN
        IF (CN+SCAN)=PERIO THEN BEGIN CCCOMPILE:=1; GO EXIT END;
        IF CN<ALGOL OR CN>COBOL THEN
            IF(T:=DIRECTORYSEARCH(ACCUM[0],"DISK ",5))=0 THEN

```

```

20582550
20582600
20582650
20582700
20582750
20582800
20582850
20582900
20582950
20583000
20583050
20583100
20583150
20583200
20583250
20583300
20583350
20583400
20583450
20583500
20583550
20583600
20583650
20583700
20583710
20583750
20583800
20583860
20584150
20584200
20584250
20584275
20584300
20584325
20584330
20584350
20584400
20584450
20584500
20584550
20584600
20584650
20584700
% (SHM)20584710
20584750
20584800
20584850
20584900
20584950
20585000
20585050
20585100
20585125
20585150
20585200
20585250
20585300
20585350
20585355
20585360

```

F



```

                BEGIN IF NOT M[T+4],[8:1] THEN
                    BEGIN LBMESS(ACCUM[0],"DISK  ",-22,0,0,
                                SPOUTUNIT,1);
                    FORGETSPACE(T); CCCOMPILE+1; GO EXIT;
                    END; FORGETSPACE(T);
                END;
COMMENT SET UP NOMINAL VALUES FOR PROGRAM PARAMETERS;%
    CMM[0]:=-(CMPLR:=ACCUM[0]); CMM[1]:=CEQN[0];
    CMM[2]:=0;
    CMM[13]:= CADDR:= GETESPDISK;
    $ SET OMIT = PACKETS
    CMM[23]:=0&UNITNO[2:42:6];
    $ POP OMIT
    $ SET OMIT = NOT(PACKETS)
    CMM[23]:=0&(IF ZIPMIX#0 THEN PSEUDOMIX[ZIPMIX]
                ELSE UNITNO)[2:42:6];
    $ POP OMIT
    CMM[27]:=CEQN[1]; %FID FOR SCHED MESS,
% GET OPTION (GO,SYNTAX CHECK, OR LIBRARY)
SKN: DO OPTN:=SCAN UNTIL OPTN=PERIO OR OPTN=SYNTA OR OPTN=LIBRA
    OR OPTN=QUEST; % IN CASE OF HYPHEN IN COMMENT PORTION
    IF OPTN=QUEST THEN
        IF SOURCE=(CARDLOC&1[30:45:3]) THEN
            BEGIN
                OPTN:=PERIO; SOURCE:=CARDLOC;
            END ELSE GO TO SKN;
    CMM[2],[8:10] := IF OPTN=PERIO THEN 1 ELSE
                    IF OPTN=SYNTA THEN 3 ELSE 4;%(OPTN=LIBRA)
    IF OPTN NEQ SYNTA THEN
% SET UP PROG ARRAY FOR COMPILE AND GO OR COMPILE TO LIBRARY JOBS
    BEGIN
        PROG[0]:= CEQN[0];
        PROG[1]:= CEQN[1];
        PROG[2]:= 0;
        MOVE(27,[PROG[2]],[PROG[3]]);
        PROG[16]:=PROG[17]:= @37777777777777;
        PROG[18]:= SHEETMAX DIV 2;
%VOID
        PROG[20]:= -1;
        PROG[21]:= 512;
        PROG[22]:= 10;
        PROG[23]:= CMM[23];
        PROG[24]:= USERID;
        PROG[26]:=IF LOGLINE GTR 0 THEN -31 ELSE LOGLINE;
    END;
EXIT: RETURNVAL:=PROCVL; % ADJUST RESULT OF TYPED PROC
    P([RETURNRCW],STS,0,RDS,0,XCH,P&P[CTF],STF);
END CCCOMPILE;
REAL PROCEDURE INITIALIZEIT;
BEGIN LABEL TRYAGAIN,LS,SPLAT,SPOT,EXIT;
REAL CMM1 = RETURNVAL+1; % BEGIN LOCAL TO INITIALIZEIT
REAL SUBROUTINE SCAN;
    SCAN+SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,DIRECT);
    P(RCW,MYMSCW,STF);
    RCW:=RCW & P(XCH)[CTC];
    P(0); % ZERO LOCAL TO INITIALIZEIT
    PROG[13]:=PADDR:=PDEX:=0; % IN CASE PROGRAM NOT IN DIRECTORY
TRYAGAIN:
    IF (T:=DIRECTORYSEARCH(ABS(CMM[0]),CMM1:=IF CMM[0] LSS 0 THEN
        "DISK  " ELSE CMM[1],3))=0 THEN

```

20585365  
20585370  
20585375  
20585380  
20585385  
20585390  
20585400  
20585450  
20585500  
20585550  
20585599  
20585600  
20585601  
20585609  
20585610  
20585620  
20585621  
20585630  
20585650  
20585700  
20585705  
20585710  
20585715  
20585720  
20585725  
20585730  
20585750  
20585800  
20585850  
20585900  
20585950  
20586000  
20586050  
20586100  
20586110  
20586150  
20586200  
20586250  
20586300  
20586350  
20586400  
20586450  
20586500  
20586510  
20586550  
20586600  
20586625  
20586650  
20586700  
20586715  
20586950  
20587050  
20587100  
20587110  
20587120  
20587130  
20587150  
20587170  
20587200  
20587250

R

```

BEGIN
  IF (CMM[0] EQV "LIBMAIN")=NOT 0 THEN
  IF (CMM[1] EQV "DISK  ")=NOT 0 THEN
  BEGIN
  ENTERSYSFILE(1);
  GO TRYAGAIN;
  END;
  BEGIN
  LBS:    LBMESS(ABS(CMM[0]),CMM1,-15,0,0,SPOUTUNIT,1);
  SPLAT: REPORTBACK(NOTIN,0,0);
  IF UNITNO GEQ 32 THEN BEGIN INITIALIZEIT:=5;GO EXIT END;
  END;
  DO T:=SCAN UNTIL T GEQ UNLOCKV AND T LEQ RESETV;
  IF UNITNO=31 AND NT1 GEQ 0 THEN BEGIN INITIALIZEIT:=7;
  GO EXIT END;
  NT1:=0; INITIALIZEIT:=1; GO EXIT ;
  END ELSE IF M[T INX 4],[9:2]=2 THEN
  BEGIN FORGETSPACE(T);
  GO TO SPOT;
  END;
  IF SECURITYCHECK(ABS(CMM[0]),
  CMM1,
  USERID,T)=0
  THEN BEGIN OPTN:=0; CMM[2]:=T;
  REPORTBACK(SECURED,0,0);
  FORGETSPACE(DIRECTORYSEARCH(ABS(CMM[0]),
  CMM[1]:=CMM1,
  13)); INITIALIZEIT:=4; GO EXIT END;
  DISKIO(N1,-(PEQN INX 0=1),30,M[T+10]);
  P(M[T INX 4],[9:2]=3);
  FORGETSPACE(T);
  CMM[24]:= USERID;
  CMM[25]:= T,[FF];
  CMM[26]:=IF LOGLINE GTR 0 THEN =31 ELSE LOGLINE;
  CMM[14]:= ABS(CMM[0]);
  SLEEP([N1],IOMASK);
  FOR T:=1 STEP 1 UNTIL 4 DO
  IF (NOT ABS(PEQN[T]&0[CTC])) NEQ NOT 0 THEN T:= 7;
  IF NOT T THEN
  BEGIN
  IF P AND PEQN[2],[3:1] THEN
  LBMESS(ABS(CMM[0]),CMM1,-46,0,0,SPOUTUNIT,1) ELSE
  SPT:  LBMESS(ABS(CMM[0]),CMM1,-19,0,0,SPOUTUNIT,1);
  FORGETSPACE(DIRECTORYSEARCH(ABS(CMM[0]),
  CMM1,13));
  REPORTBACK(NOTX,0,0);
  GO TO SPLAT;
  END;
  IF PEQN[6] LSS 0 THEN FOR T:=15 STEP 1 UNTIL 22 DO
  CMM[T]:=PEQN[T] ELSE
  BEGIN
  CMM[15]:= 0;
  CMM[16]:= CMM[17]:= @3777777777777;
  CMM[18]:=SHEETMAX DIV 2;
  CMM[19]:= 0;
  CMM[20]:= PEQN[7],[FF];
  CMM[21]:= 512;
  END;
  INITIALIZEIT:=3;

```

```

20587300
20587310
20587320
20587330
20587340
20587350
20587360
20587500
20587550
20587610
20587650
20587700
20587750
20587800
20587850
20587900
20587950
20588000
20588010
20588020
20588030
20588050
20588100
20588150
20588200
20588210
20588250
20588300
20588350
20588400
20588410
20588450
20588500
20588550
20588560
20588570
20588600
20588650
20588700
20588750
20588800
20588810
20588850
20588900
20589000
20589050
20589110
20589150
20589200
20589250
20589300
20589350
20589400
20589450
20589460
20589470
20589480
20589490
20589500
20589550

```

```

EXIT:      RETURNVAL:=PROCVAL; % ADJUST RESULT OF TYPED PROC          20589600
          P([RETURNRCW],STS,0,RDS,0,XCH,P&P[CTF],STF);                20589610
END INITIALIZEIT;                                                    20589650
REAL PROCEDURE CCUNIT;                                               20589700
BEGIN LABEL U1,ERROR,EXIT;                                          20589720
REAL SUBROUTINE SCAN;                                               20589950
  SCAN*SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,DIRECT);     20590000
  P(RCW,MYMSCW,STF);                                                 20590010
  RCW:=RCW & P(XCH)[CTC];                                           20590020
  T:=SCAN; CN:=ACCUM[0];                                             20590050
  T*SCAN; IF T#EQUAL THEN GO ERROR;                                  20590100
  FOR T:= 0 STEP 1 UNTIL 31 DO                                       20590150
    IF CN,[6:18]=TINU[T],[30:18] THEN GO TO U1;                    20590200
    GO ERROR;                                                         20590250
U1: IF LABELTABLE[T] NEQ @314 THEN BEGIN CCUNIT:=6; GO EXIT END;   20590300
  CN:=SCAN;                                                          20590350
  MULTITABLE[T]:=RDCTABLE[T];=0;                                    20590400
  LABELTABLE[T]:=ACCUM[0];                                          20590450
  IF (CN:=SCAN) = SLASH THEN                                         20590500
    BEGIN MULTITABLE[T]:=LABELTABLE[T];                             20590550
      CN*SCAN; LABELTABLE[T]*ACCUM[0]; CN*SCAN;                    20590600
    END;                                                             20590610
    IF CN=COMMA THEN                                                 20590650
      BEGIN IF(CN*SCAN)#IDENT OR KOUNT>3 THEN GO ERROR;           20590655
        STREAM(R*0;KOUNT,ACCUM);                                    20590660
        BEGIN SI*ACCUM;SI*SI+1;DI*LOC R;DS*KOUNT OCT END;        20590665
        RDCTABLE[T]*P(XCH,RDCTABLE[T])&P(XCH)[14:38:10];         20590668
        IF(CN*SCAN)=COMMA THEN                                       20590670
          BEGIN IF(CN*SCAN)#IDENT OR KOUNT>5 THEN GO ERROR;       20590675
            STREAM(R*0;KOUNT,ACCUM);                                  20590680
            BEGIN SI*ACCUM;SI*SI+1;DI*LOC R;DS*KOUNT OCT END;    20590685
            RDCTABLE[T]*P(XCH,RDCTABLE[T])&P(XCH)[24:31:17];     20590688
            IF(CN*SCAN)=COMMA THEN                                       20590690
              BEGIN IF(CN*SCAN)#IDENT OR KOUNT>2 THEN GO ERROR;   20590695
                STREAM(R*0;KOUNT,ACCUM);                              20590700
                BEGIN SI*ACCUM;SI*SI+1;DI*LOC R;DS*KOUNT OCT END; 20590705
                RDCTABLE[T]*P(XCH,RDCTABLE[T])&P(XCH)[41:41:7];  20590710
              END %CYCLE                                             20590715
              END %CREATION DATE                                     20590720
            END; %REEL NUMBER                                         20590725
          IF CN#PERIO THEN DO CN*SCAN UNTIL CN=PERIO;CCUNIT*0;GO EXIT; 20590730
        ERROR; CCUNIT*6;                                           20590740
      EXIT:      RETURNVAL:=PROCVAL; % ADJUST RESULT OF TYPED PROC 20590750
              P([RETURNRCW],STS,0,RDS,0,XCH,P&P[CTF],STF);        20590751
    END CCUNIT;                                                      20590800
  REAL PROCEDURE CCSECMAINT;                                         20590850
  BEGIN LABEL EXIT,CCC;                                             20590910
  REAL SUBROUTINE SCAN;                                             20591350
    SCAN*SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,DIRECT);  20591400
    LABEL OPTNO,OPTN1,OPTN2,SEC1,SEC2,SEC5,ST1,                    20591450
      ST2,LS;                                                         20591500
    SWITCH SW:=OPTNO,OPTN1,OPTN2;                                    20591550
    P(RCW,MYMSCW,STF);                                               20591600
    RCW:=RCW & P(XCH)[CTC];                                           20591610
    GO TO SW[OPTNN];                                                 20591620
  OPTNO:  USERID:=ABS(USERID);                                       20591650
          IF SCAN LSS IDENT THEN BEGIN CCSECMAINT:=6;GO EXIT END;  20591700
          SMID:=CMM[0];=ACCUM[0]; CN:=SCAN;                          20591750
          IF SCAN LSS IDENT THEN BEGIN CCSECMAINT:=6; GO EXIT END;  20591800
          20591850

```

```

SFIDI:= CMM[1]:= ACCUM[0]; CDEXI:= 0; 20591900
IF (SFH:=DIRECTORYSEARCH(SMID,SFID,4))=0 THEN GO TO LS; 20591950
IF NOT((SMID EQV "PBD ")=NOT 0) AND (M[SFH+5]=0 20592050
AND M[SFH+2] NEQ 0) THEN 20592100
% INHIBIT USE ON PUBLIC, SECURE FILES 20592150
BEGIN CN:=SCAN; GO TO OPTN2 END; 20592200
OPTN1:=0; CMM[2]:= SFH; 20592250
FORGETSPACE(DIRECTORYSEARCH(CMM[0],CMM[1],14)); 20592300
20592350
OPTN1: STREAM(USERID,QI:=USERID>0,B:=[CMM],DI:=CN:=SPACE(10)); 20592400
BEGIN Q(SI:=LOC USERID; SI:=SI+1;DS:=LIT " "; DS:= 7CHR); 20592450
DS:= 17LIT " INVALID USER OF "; SI:=B; 20592500
SI:=SI+1; DS:= 7CHR; DS:=LIT "/"; SI:=SI+1; DS:= 7CHR; 20592550
DS:=LIT" "; 20592600
END STREAM; 20592650
SPOUTER(CN, SPOUTUNIT,1); 20592700
FORGETSPACE(CMM[2]); 20592725
IF OPTN NEQ 0 THEN GO TO SEC5; 20592750
IF UNITNO GEQ 32 THEN BEGIN CCSECMAINT:=5;GO EXIT END; 20592800
GO TO CCC; 20592850
OPTN2: CMM[5]:=USERID; 20592900
ST:= CDEXI:= 0; 20592950
SEC1: FOR OPTN:=0 STEP 1 UNTIL 1 DO 20593000
BEGIN CN:= SCAN, IF T=OPEN AND CN=UNLOCKV AND OPTN=0 THEN 20593050
BEGIN T:= UNLOCKV; GO TO SEC1; END 20593100
ELSE IF CN LSS IDENT AND CN NEQ EQUAL THEN GO TO ST1; 20593150
CMM[OPTN]:= IF CN=EQUAL THEN -1 ELSE ACCUM[0]; 20593200
CN:=SCAN; 20593250
END; 20593300
IF CN=WITH THEN BEGIN CN=SCAN;CMM[6]+IF CN>IDENT THEN ACCUM[0] 20593310
ELSE USERID; CN+SCAN END ELSE CMM[6]+USERID; 20593320
IF CMM[0] GEQ 0 AND CMM[1] GEQ 0 THEN GO TO SEC2; 20593350
N1:= CMM[0]; N2:= CMM[1]; N3:= 0; ST:= 1; 20593400
ST2: SEEKNAM (N1,N2,N3,CMM[0],CMM[1],T1); 20593450
IF N3 NEQ 0 THEN GO TO SEC2; 20593500
ST:= 0; GO TO SEC5; 20593550
SEC2: IF (ABS(USERID)EQV MCP) NEQ NOT 0 THEN 20593600
IF (CMM[0] EQV "PBD ") = NOT 0 THEN GO TO SEC5; 20593650
SECURITYMAINT(T,SMID,SFID,CMM,SFH,SPOUTUNIT); 20593750
SEC5: IF ST THEN GO TO ST2; 20593800
IF CN=COMMA THEN GO SEC1; 20593850
IF T=USEV THEN 20593900
BEGIN DISKWAIT(SFH,[CF],30,SFH,[FF]); 20593950
P(DIRECTORYSEARCH(=SMID,SFID,14),DEL); 20593960
END; 20593970
GO TO CCC; 20594000
20594050
LS: LBMESS(CMM[0],CMM[1],-15,0,0,SPOUTUNIT,1); 20594350
REPORTBACK(NOTIN,0,0); 20594360
IF UNITNO GEQ 32 THEN BEGIN CCSECMAINT:=5; GO EXIT END; 20594400
CCC: DO T=SCAN UNTIL T>IDENT AND T<RESETV; 20594450
IF UNITNO=31 AND NT1 GEQ 0 THEN BEGIN CCSECMAINT:=7; GO EXIT END; 20594500
NT1:= 0; CCSECMAINT:=1; GO EXIT; 20594550
ST1: IF T=USEV THEN 20594600
FORGETSPACE(DIRECTORYSEARCH(SMID,SFID,SFH&1[214711])); 20594650
CCSECMAINT:=6; 20594700
EXIT: RETURNVAL:=PROCVAL; % ADJUST RESULT OF TYPED PROC 20594750
P([RETURNRCW],STS,0,RDS,0,XCH,P&P[CTF],STF); 20594751
END CCSECMAINT; 20594800
REAL PROCEDURE CCLABEL; 20594850

```

```

BEGIN LABEL EXIT;
P(RCW,MYMSCW,STF);
RCW:=RCW & P(XCH)[CTC];
CN:=0;
UNITCODE[UNITNO=23]:= USERID;
MULTITABLE[UNITNO]:= 0;
RDCTABLE[UNITNO]:= 1&1[14:38:10];
IF UNITNO=23 THEN BEGIN CN:=READERA,[FF];READERA:=CARDLOC END
ELSE IF UNITNO=24 THEN BEGIN CN:=READERB,[FF];READERB:=CARDLOC END
ELSE IF UNITNO GEQ 32 THEN BEGIN CN:= CIDROW[UNITNO=32],[3:5];
CIDROW[UNITNO=32]:=(*P(DUP))&0[3:43:5]&
CARDLOC[CTF];
END;
IF CN THEN BEGIN LABELTABLE[UNITNO]:= "CARD 00" OR
((IF UNITNO GEQ 32 THEN "C/" ELSE @5772) + UNITNO);
CCLABEL:=8; GO EXIT;
END;
IF T = LABEV THEN BEGIN
MULTITABLE[UNITNO]:=M[CARDLOC+1],[6:42];
STREAM(A:=0,B:=0,C:=0;D:=CARDLOC+3);
BEGIN DI:=LOC A; SI:=D;DS:=30CT;
DS:=50CT; DS:=20CT; END;
P(P(XCH)&P[24:31:17]&P(XCH)[14:38:10],
[RDCTABLE[UNITNO]],+);%
LABELTABLE[UNITNO]:=M[CARDLOC+2],[6:42];
END
ELSE IF SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,DIRECT)
GEQ IDENT THEN LABELTABLE[UNITNO]:=ACCUM[0]
ELSE BEGIN IF UNITNO GEQ 32 THEN
CIDROW[UNITNO=32],[18:15]:=0;
CCLABEL:=6; GO EXIT;
END;
CCLABEL:=8;
EXIT: RETURNVAL:=PROCVAL; % ADJUST RESULT OF TYPED PROC
P([RETURNRCW],STS,0,RDS,0,XCH,P&P[CTF],STF);
END CCLABEL;
PROCEDURE CONTROLCARD(CARD); VALUE CARD; REAL CARD;
BEGIN
LABEL CC,CCTYPE,COMPILE,INITIALIZATION,BEFORETRYNEXT,TRYNEXT,
CONTROLLER,CONTROLA,COMPILEJOB,COMJOB,EXEC,EXRUN,RUN,
USERS,USES,SECBOMB,UNLOX,LOX,FREES,OPENS,ENTE,
LCOPY,CHANGE,REMOVE,UNITI,INCSC,ENDF,ENDECK,SAVEND,
LABE,FINIS,ZIPEXIT,EXIT,SET,RSET,DOWN;
LABEL CCC,PACK,PACK2,WAIT,ZIPLIST;
SWITCH TYPE+ UNLOX,USES,LOX,FREES,OPENS,PACK,USERS,
RUN,COMPILE,EXEC,LCOPY,LCOPY,ENTE,ENTE,REMOVE,
CHANGE,UNITI,ENDF,WAIT,LABE,LABE,SET,RSET;
SWITCH SW+ CC,CCTYPE,INITIALIZATION,BEFORETRYNEXT,SECBOMB,ENDECK,
INCSC,ZIPEXIT,EXIT,PACK2;
DEFINE ZIPMIX=CARD,[18:6]#, PSOURCE=CARD,[24:6]#;
REAL SUBROUTINE SCAN;
SCAN:=SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,DIRECT);
% SET OMIT = NOT(PACKETS)
SUBROUTINE LISTHECARD;
IF LASTSCAN,[2:1] THEN
IF SPOUTUNIT GEQ 32 THEN
IF T#PACKET THEN
BEGIN
LASTSCAN,[2:1]:=0; ABORT:=0;
IF UNITNO=31 THEN

```

20594870  
20595080  
20595090  
20595150  
20595200  
20595250  
20595300  
20595350  
20595400  
20595450  
20595500  
20595550  
20595600  
20595650  
20595700  
20595750  
20595800  
20595850  
20595900  
20595950  
20596000  
20596050  
20596100  
20596150  
20596200  
20596250  
20596300  
20596350  
20596400  
20596450  
20596500  
20596550  
20596600  
20596650  
20596651  
20596700  
20597550  
20597600  
20597650  
20597700  
20597750  
20597800  
20597850  
20597880  
20597900  
20597950  
20598000  
20599000  
20599100  
20600000  
20600020  
20600040  
20600099  
20600100  
20600110  
20600120  
20600130  
20600140  
20600150  
20600160

STREAM(E:="END....", CARDLOC);	20600170
BEGIN SII=CARDLOC; DII=LOC E; DII=DI+1;	20600180
L1: IF SC=" " THEN BEGIN SII=SI+1; GO L1; END;	20600190
IF SC="*" THEN GO FINI;	20600200
IF SC=ALPHA THEN	20600210
IF SC="E" THEN	20600220
BEGIN	20600230
IF 3 SC=DC THEN IF SC=ALPHA THEN ELSE	20600240
BEGIN	20600250
CARDLOC:=SI; DII=CARDLOC; DS:=LIT"+";	20600260
GO FINI;	20600270
END;	20600280
SII=SI-3; DII=DI-3; GO L2;	20600290
END ELSE % ALPHANUMERIC	20600300
BEGIN	20600310
L2: SII=SI+1; IF SC=ALPHA THEN GO L2;	20600320
END ELSE % SPECIAL CHR	20600330
SII=SI+1;	20600340
GO L1;	20600350
FINI;	20600360
END;	20600370
ZIPLIST:	20600380
STREAM(ZL:=0; CARDLOC, ABORT, PPC:=PPCPROCESS,	20600390
ZZP:=UNITNO=31; D:=NT1:=SPACE(10));	20600400
BEGIN SII=CARDLOC; ABORT(SII=SI+36);	20600410
DS:=LIT">"; PPC(DS:=4LIT">"); ZZP(DS:=2LIT">");	20600420
2(36(IF SC="*" THEN JUMP OUT 2 TO ZER; DS:=CHR));	20600430
TALLY:=1; ZL:=TALLY;	20600440
ZER; DS:=LIT"+";	20600450
END;	20600460
SPOUTER(NT1, SPOUTUNIT, 64);	20600470
IF P AND (UNITNO=31) THEN	20600480
IF (ABORT:=ABORT+2) < 30 THEN GO ZIPLIST;	20600490
ABORT:=0;	20600500
END LISTHECARD;	20600510
\$ POP OMIT	20600511
P(0,0,0,0,0,0,0,0,0,0);%	20600600
P(0,0,0,0,0,0,0,0,0,0);%	20600650
P(0,0,0,0,0,0,0,0,0,0);%	20600700
P(0,0,0,0,0,0,0,0,0,0);	20600750
% DO NOT ZERO THE LAST THREE LOCALS (RETURN=MSCW, RCW, & VAL)	20600755
RCW:=RCW & P(.,.CONTROLCARD,LOD)[CTC];	20600760
UNITNO:= (CARD,[3:5]+28),[43:5]+4;	20600850
IF CARD,[33:15] = 0 THEN	20600900
BEGIN CARD,[33:15] := SPACE(13)+2;	20600950
IF WAITIO(CARD INX @40000000,@15,UNITNO),[45:3] NEQ 0%	20601000
THEN	20601050
BEGIN LABELTABLE[UNITNO] := @114;%	20601100
RRRMECH := NOT TWO (UNITNO) AND RRRMECH;%	20601150
FORGETSPACE(CARD INX NOT 1);%	20601200
KILL([CARD] INX NOT 1);%	20601250
END;	20601300
END;	20601350
IF SWAPEND=0 THEN SLEEP([SWAPEND],63);	20601360
COMMENT SET UP ACCUM ARRAY FOR SCAN;%	20601450
ACCUM:=M[SPACE(10)]&10[8:38:10];%	20601500
ACCUM[0] := 0;%	20601550
IF (CCTBLWORD:=P(CCTBLWORD,DUP)&(P.[FF]+1)[CTF]),[FF]>1 THEN	20601600
BEGIN	20601620
IF CCTBLADDR=0 THEN SLEEP([CCTBLWORD],@77777);	20601640

DIRECT:=[M[CCTBLWORD]]&CCTABLSZ[8:38:10];	20601660
END ELSE	20601680
BEGIN	20601700
DIRECT:=[M[T]=SPACE(CCTABLSZ)]&CCTABLSZ[8:38:10];	20601720
DISKWAIT(-T,CCTABLSZ,MESSAGETABLE[3],[22:26]);	20601740
CCTBLADDR:=T;	20601760
END;	20601780
CMM:= IOQUE&SPACE(130)[CTC];	20601850
PEQN:=(31 INX (CEQN:=(31 INX(PROG:=(31 INX CMM)))));%	20601900
% PLACE ", " IN COL 73 ;%	20601950
CARDLOC := CARD INX 0;%	20602000
IF UNITNO=25 OR UNITNO=26 OR UNITNO=30 OR UNITNO=31 THEN	20602050
SOURCE:=CARDLOC ELSE	20602100
M[(SOURCE:=CARDLOC)+9] := @3277320000000000; % , " , 2B XTRA SAFE	20602150
IF UNITNO GEQ 32 AND UNITCODE[UNITNO=23],[1:1] THEN	20602200
UNITCODE[UNITNO=23]:=M[CARDLOC + 10];	20602250
IF UNITNO=25 OR UNITNO=30 OR UNITNO=31 THEN USERID:=MCP ELSE	20602300
BEGIN IF UNITNO=26 THEN UNITNO:=31;%	20602350
USERID:=UNITCODE[UNITNO=23];%	20602400
END;%	20602450
SPOUTUNIT:=(	20602460
\$ SET OMIT = NOT(PACKETS)	20602469
IF ZIPMIX#0 AND PSEUDOMIX[ZIPMIX] GEQ 32 THEN	20602470
PSEUDOMIX[ZIPMIX] ELSE	20602480
IF UNITNO GEQ 32 THEN UNITNO ELSE	20602490
0);	20602491
\$ SET OMIT = NOT(PACKETS)	20602500
IF UNITNO GEQ 32 THEN	20602509
IF PSEUDO[UNITNO=32]=0 THEN	20602510
PRINTTHECOVER(CARDLOC,UNITNO,PSOURCE);	20602520
LASTSCAN:=0&1[2:47:1];	20602530
\$ POP OMIT	20602540
COMMENT SCAN FOR CARD WITH QUESTION MARK IN COL, 1;%	20602541
CC:        IF SCAN NEQ QUEST THEN GO TO INCSC;%	20602550
T:=SCAN;	20602600
CCTYPE:        IF (T LSS UNLOCKV) OR (T GTR RESETV) THEN%	20602650
GO TO INCSC;%	20602700
\$ SET OMIT = NOT(PACKETS)	20602750
LISTHECARD;	20602800
\$ POP OMIT	20602850
% BRANCH ON 1ST WORD ON CONTROL CARD%	20603359
CMPLR:= -1;	20603360
LIBNO:=0;	20603361
TOG:= FALSE;	20603450
IF (T LEQ ENTER) AND (T GEQ RUNV) THEN	20603460
BEGIN	20603500
M[CARDLOC-2] := 0;	20603550
DISKWAIT(CARDLOC,10,CMM[6]:=GETESPDISK);	20603555
END;	20603560
GO TO TYPE[ T = UNLOCKV ];%	20603565
% COMPILER CALL OUT CARD%	20603570
COMPILE: IF CCCOMPILE THEN GO INCSC;	20603575
INITIALIZATION: OPTNN:=INITIALIZEIT; GO DOWN;	20603600
BEFORETRYNEXT: IF OPTN=PERIO THEN GO TO CONTROLER;	20603650
TRYNEXT: IF KOUNT=@14 THEN	20603700
IF SOURCE=(CARDLOC&1[30:45:3]) THEN	20603750
	20603900
	20604050
	20604100
	20604105

BEGIN	20604110
PPCPROCESS:=1; T:=SCAN; GO CONTROLA;	20604115
END;	20604120
IF SCAN NEQ PERIO THEN GO TRYNEXT;	20604125
CONTROLLER: PPCPROCESS:= 1;	20604150
IF SCAN NEQ QUEST THEN GO TO INCSC;	20604200
T:= SCAN;	20604250
CONTROLA: IF (T LSS SETV OR T>COBOL) AND ACCUM[0] NEQ CMPLR THEN	20604300
IF T GEQ UNLOCKV AND T LEQ LABEV THEN GO TO FINIS	20604350
ELSE GO TO INCSC;	20604360
\$ SET OMIT = NOT(PACKETS)	20604479
LISTHECARD;	20604480
\$ POP OMIT	20604481
IF T GEQ ALGOL OR ACCUM[0]=CMPLR THEN	20604500
IF OPTN=EXECU OR OPTN=RUNV THEN	20604550
GO TO TRYNEXT	20604600
ELSE GO TO COMPILEJOB;	20604650
IF OPTN=SYNTA THEN GO TO TRYNEXT;	20604700
IF OPTN=EXECU OR OPTN=RUNV THEN GO TO COMJOB;	20604750
% CALL PPC FOR COMPILE AND GO JOB%	20604800
IF PPC(PADDR,PEQN,PROG,PDEX,T,UNITNO,CARDLOC,SOURCE,ACCUM,	20604850
LASTSCAN,DIRECT) THEN GO TO INCSC;	20604900
GO TO CONTROLLER;	20604950
COMPILEJOB: T:=SCAN;	20605000
COMJOB: IF PPC(CADDR,CEQN,CMM,CDEX,T,UNITNO,CARDLOC,SOURCE,ACCUM,	20605050
LASTSCAN,DIRECT) THEN GO TO INCSC;	20605100
GO TO CONTROLLER;	20605150
	20605200
COMMENT EXECUTE CARD;%	20605250
EXEC: P(EXECU);	20605300
EXRUN: OPTN:=P;	20605320
CMM[13]:=CADDR:=CDEX:=0;	20605340
T:=SCAN; CMM[0]:=ACCUM[0];	20605360
T:=SCAN; T:=SCAN;	20605380
IF ((CMM[1]:=ACCUM[0]) EQV "DISK ") = NOT 0 THEN	20605400
IF ((T:=CMM[0]) EQV "LDCNTRL") = NOT 0 THEN OPTN:=RUNV ELSE	20605420
IF (T EQV "PRNPBT ") = NOT 0 OR (T EQV "LIBMAIN") = NOT 0 THEN	20605440
IF UNITNO#31 THEN GO TO INCSC;	20605460
CMM[2]:=0&(IF OPTN=RUNV THEN 5 ELSE 2)[8:38:10];	20605480
CMM[23]:=0&	20605500
\$ SET OMIT = NOT(PACKETS)	20605509
IF ZIPMIX#0 THEN PSEUDOMIX[ZIPMIX] ELSE	20605510
\$ POP OMIT	20605511
UNITNO)[2:42:6];	20605520
GO TO INITIALIZATION;	20605550
RUN: P(RUNV);	20605600
GO TO EXRUN;	20605650
USERS: T:=SCAN; T:=SCAN;	20605700
IF (USERID,[1:1] AND USERID#MCP)	20605750
THEN BEGIN	20605800
USERID:=ACCUM[0];	20605810
\$ SET OMIT = NOT(PACKETS)	20605819
IF UNITNO GEQ 32 THEN UNITCODE[UNITNO-23]:=USERID;	20605820
\$ POP OMIT	20605821
END;	20605830
CCC: %COME HERE TO FLUSH TO NEXT INITIAL WORD	20605870
\$ SET OMIT = NOT(PACKETS)	20605879
DO T:=SCAN UNTIL T=QUEST;T:=SCAN;	20605880
\$ POP OMIT	20605881
\$ SET OMIT = PACKETS	20605899



DO T←SCAN UNTIL T>IDENT AND T≤RESETV;	20605900
\$ POP OMIT	20605901
GO TO CCTYPE;	20606000
USES: OPTNN:=0; OPTNN:=CCSECMAINT; GO DOWN;	20606050
SECBOMB: OPTNN:=1; OPTNN:=CCSECMAINT; GO DOWN;	20606100
UNLOX;	20606150
LOX;	20606200
FREES;	20606250
OPENS;	20606300
OPTNN:=2; OPTNN:=CCSECMAINT; GO DOWN;	20606350
ENTE:;	20606400
LCOPY;	20606450
CHANGE;	20606500
REMOVE;	20606550
OPTNN:=LIBCC;	20606600
DOWN: GO SW(OPTNN);	20606610
SET: TOG:= TRUE;	20606650
RSET: IF CCSET THEN GO CC ELSE GO INCSC;	20606700
UNIT: OPTNN:=CCUNIT; GO DOWN;	20606800
INCSC:	20606850
IF PPCPROCESS THEN	20606860
P(DIRECTORYSEARCH(←CMM[0],IF CMM[0] LSS 0 THEN "DISK " ELSE	20606865
CMM[1], 13),DEL);	20606870
\$ SET OMIT = NOT(PACKETS)	20606874
LISTHECARD;	20606875
\$ POP OMIT	20606876
LASTSCAN := 0;	20607000
STREAM(CARDLOC, U:=TINU[UNITNO], ACCUM, MIX:=ZIPMIX,	20607020
ZZP:=UNITNO=31, CRD:=SPOUTUNIT=0,	20607040
DI:=T:=SPACE(15));	20607060
BEGIN	20607080
DS:=20LIT"#CONTROL CARD ERROR ";	20607100
SI:=LOC U; SI:=SI+5; DS:=3 CHR;	20607120
ZZP(DI:=DI-22; DS:=24LIT"ZIP ERROR, IGNORED, MIX=";	20607140
SI:=LOC MIX; DS:=2 DEC; DS:=LIT"!";	20607160
D:=DI; DI:=DI-3; DS:=FILL; DI:=D);	20607180
DS:=4LIT" AT ";	20607200
SI:=ACCUM; SI:=SI+1;	20607220
7(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR);	20607240
CRD(DS:=LIT"!"; SI:=CARDLOC; 2(DS:=36 CHR));	20607260
DS:=LIT" ";	20607280
END;	20607300
IF UNITNO≠25 THEN	20607500
BEGIN IF UNITNO=30 THEN TWXOUT(T,100,1 OR M,ABS(SPOWORD)) ELSE	20607550
SPOUTER(T,SPOUTUNIT,1);	20607600
IF UNITNO=31 THEN REPORTBACK(ZIPER,0,0);	20607650
IF UNITNO=30 OR UNITNO=31 THEN GO ZIPEXIT;	20607700
IF UNITNO GEQ 32 THEN GO ENDECK;	20607750
END ELSE	20607800
BEGIN P(WAITIO(T,0,25),DEL);	20608000
FORGETSPACE(T);	20608050
\$ SET OMIT = PACKETS	20608059
GO TO ENDF;	20608060
\$ POP OMIT	20608061
\$ SET OMIT = NOT(PACKETS)	20608069
FETCH(UNITNO,CARDLOC,SOURCE);	20608070
IF SCAN NEQ QUEST THEN GO TO INCSC;	20608072
T:=SCAN;	20608074
IF PPCPROCESS THEN GO TO CONTROLA;	20608076
IF(T GEQ PACKET)AND(T LEQ LABEV)AND(T NEQ RUNV) THEN	20608078

GO TO CCTYPE; GO TO INCSC;	20608080
\$ POP OMIT	20608081
END;	20608100
\$ SET OMIT = NOT(PACKETS)	20608109
ENDECK;	20608110
IF UNITNO GEQ 32 THEN	20608120
BEGIN ABORT:=TRUE;	20608130
PACKETERR[UNITNO-32]:=TRUE;	20608140
GO TO PACK2;	20608142
END;	20608144
\$ POP OMIT	20608146
DO DO	20608150
FETCH(=UNITNO,CARDLOC,SOURCE)	20608200
UNTIL SCAN=QUEST	20608250
UNTIL SCAN=ENDFI;	20608300
ENDFI;	20608450
\$ SET OMIT = NOT(PACKETS)	20608459
IF UNITNO LSS 32 THEN	20608460
\$ POP OMIT	20608461
IF UNITNO NEQ 30 THEN UNITCODE[UNITNO-23]:=-0;	20608500
IF UNITNO=23 THEN READERA:=0 ELSE	20608510
IF UNITNO=24 THEN READERB:=0 ELSE	20608520
IF UNITNO GEQ 25 THEN	20608550
IF UNITNO GEQ 32 THEN	20608600
PACK2:; %PACKET CARDS END HERE FROM PSEUDO-READERS	20608610
IF CIDTABLE[UNITNO-32,3] LSS CIDTABLE[UNITNO-32,7] THEN	20608650
BEGIN FETCH(=UNITNO,CARDLOC,SOURCE);	20608700
\$ SET OMIT = NOT(PACKETS)	20608709
IF ABORT THEN	20608710
BEGIN	20608720
IF (T:=SCAN)=QUEST THEN	20608730
IF (T:=SCAN)=ENDFI OR T=WAITV THEN	20608740
ABORT:=FALSE;	20608750
IF T#WAITV THEN GO PACK2;	20608760
END ELSE T:=0;	20608770
LASTSCAN:=0&1[2:47:1];	20608780
PACKETERR[UNITNO-32]:=FALSE;	20608790
IF T=WAITV THEN GO WAIT;	20608800
\$ POP OMIT	20608801
GO CC;	20608810
END ELSE	20608820
BEGIN	20608830
\$ SET OMIT = NOT(PACKETS)	20608839
LABELTABLE[UNITNO]:=@114;	20608840
IF PACKETACT[UNITNO-32]=0 THEN	20608850
\$ POP OMIT	20608851
\$ SET OMIT = PACKETS	20608859
ENDECK;	20608860
\$ POP OMIT	20608861
ENDOFDECK((UNITNO-32)&CARD[1:1:1]);	20608870
GO ZIPEXIT;	20608880
END ELSE	20608890
GO ZIPEXIT;	20608900
IF(TWO(UNITNO) AND SAVEWORD) NEQ 0 THEN GO TO SAVEND;	20608950
IF WAITIO(CARDLOC&400[18:33:15],@15,UNITNO).[45:3] NEQ 0 THEN	20609000
BEGIN	20609050
SAVEND: LABELTABLE[UNITNO]:= @114;	20609100
RRRMECH:= NOT (NT1:= TWO(UNITNO)) AND RRRMECH OR	20609150
NT1 AND SAVEWORD;	20609200
GO TO ZIPEXIT;	20609250

END;	20609300
M[(SOURCE:=CARDLOC)+9]=0&","[1143:5];	20609350
USERID:=UNITCODE[UNITNO=23];	20609400
GO TO CC;	20609410
PACK: IF UNITNO<32 THEN GO INCSC;	20609420
GO PACK2;	20609450
LABEL: OPTNN:=CCLABEL; GO DOWN;	20609500
WAIT;	20609555
\$ SET OMIT = NOT(PACKETS)	20609559
IF UNITNO<32 THEN GO TO CCC;	20609560
IF PACKETACT[UNITNO=32]=0 THEN GO TO CCC;	20609570
LABELTABLE[UNITNO]:=@214; GO TO ZIPEXIT;	20609580
\$ POP OMIT	20609581
FINIS: CCFINISH;	20609600
\$ SET OMIT = NOT(PACKETS)	20609659
IF (NT1-IF ZIPMIX#0 THEN PSEUDOMIX[ZIPMIX] ELSE UNITNO)	20609660
GEQ 32 THEN PACKETACT[NT1=32]:=PACKETACT[NT1=32]+1;	20609670
\$ POP OMIT	20609671
SELECTION;	20609700
IF UNITNO NEQ 31 THEN	20609750
BEGIN	20609760
\$ SET OMIT = PACKETS	20609799
IF LIBNO NEQ 0 AND (UNITNO=23 OR UNITNO=24 OR UNITNO GEQ 32)	20609800
AND T NEQ ENDFI THEN	20609850
BEGIN	20609900
LABELTABLE[UNITNO]:=@214;	20609950
SLEEP([LABELTABLE[UNITNO]],@100);	20610000
LABELTABLE[UNITNO]:=-@14;	20610050
END;	20610060
\$ POP OMIT	20610061
GO CCTYPE;	20610100
END;	20610150
ZIPEXIT: FORGETSPACE(CARDLOC=2);	20610200
EXIT:;	20610250
\$ SET OMIT = NOT(PACKETS)	20610259
IF ZIPMIX NEQ 0 THEN	20610260
BEGIN NYLONZIPPER[ZIPMIX],[2:1]:=1;	20610265
IF PSEUDOMIX[ZIPMIX] NEQ 0 THEN	20610270
IF MEMROW[ZIPMIX],[CF] GEQ FENCE THEN	20610275
BRINGBACK[ZIPMIX];	20610280
END;	20610285
\$ POP OMIT	20610286
FORGETSPACE(ACCUM INX 0);%	20610300
FORGETSPACE(CMM INX 0);%	20610350
IF (CCTBLWORD:=P(CCTBLWORD,DUP)&(P,[FF]=1)[CTF]),[FF]=0 THEN	20610400
BEGIN	20610410
FORGETSPACE(CCTBLADDR);	20610420
CCTBLADDR:=0;	20610430
END;	20610440
KILL([CARD] INX NOT 1);%	20610600
END CONTROLCARD;	20610650
REAL PROCEDURE CCSET;	20700000
BEGIN LABEL MORE,SEEK,SKIP,CCERR,L1,L2;	20701000
REAL FXTOG = RETURNVAL+1, % BEGIN LOCALS OF CCSET	20702000
LOK = FXTOG+1,	20702100
N = LOK+1,	20703000
SENSETOG = N+1;	20704000
REAL SUBROUTINE SCAN;	20705000
SCAN=SCN(UNITNO,CARDLOC,SOURCE,ACCUM,KOUNT,LASTSCAN,DIRECT);	20706000
P(CRW,MYMSCW,STF);	20707000

RCW:=RCW & P(XCH)[CTC];	20708000
P(0,0,0,0); % ZERO LOCALS OF CCSET	20709000
UNITNO:=CARD.[2:6];	20710000
CARDLOC:=CARD INX 0;	20711000
IF NOT (FXTOG:=(CN:=SCAN)=FIXED) THEN	20712000
IF NOT (SENSETOG:=(CN:=SENSE)) THEN	20713000
IF CN ≠ ACCESSD THEN GO TO CCERR;	20714000
MORE:	20715000
IF (CN:=SCAN)=EQUAL THEN CMM[0]:=1 ELSE	20716000
IF CN GEQ IDENT THEN CMM[0]:=ACCUM[0] ELSE GO CCERR;	20717000
IF SCAN NEQ SLASH THEN GO TO CCERR;	20718000
IF (CN:=SCAN)=EQUAL THEN CMM[1]:=1 ELSE	20719000
IF CN GEQ IDENT THEN CMM[1]:=ACCUM[0] ELSE	20720000
GO TO CCERR;	20721000
CN:=T:=0;	20722000
SEEK:	20723000
IF (CMM[0] OR CMM[1]) LSS 0 THEN	20724000
SEEKNAM(CMM[0],CMM[1],CN,CMM[2],CMM[3],N) ELSE	20725000
BEGIN CN:=1; CMM[2]:=CMM[0]; CMM[3]:=CMM[1] END;	20726000
IF CN NEQ 0 THEN	20727000
BEGIN	20728000
IF NOT FXTOG THEN IF SYSTEMFILE(CMM[2],CMM[3]) THEN	20729000
BEGIN T:=2; GO SKIP; END;	20730000
T:=DIRECTORYSEARCH(CMM[2],CMM[3],19);	20731000
END ELSE IF N=0 THEN BEGIN CMM[2]:=CMM[0]; CMM[3]:=CMM[1]; GO L1;	20732000
END	20733000
ELSE GO L2;	20734000
SKIP:	20735000
IF T GEQ 64 THEN	20736000
BEGIN	20737000
IF M[T+4],[43:2]=3 THEN	20738000
BEGIN DISKWAIT(T,[CF],-30,T,[FF]); T:=1; GO L1; END;	20739000
IF (USERID EQV MCP)= NOT 0 OR	20740000
(USERID EQV ABS(M[T+2]))= NOT 0 OR	20741000
(NOT SENSETOG AND (M[T+2]=0)) THEN	20742000
BEGIN	20743000
LOK:=0;	20744000
IF FXTOG	20745000
THEN M[T+4],[42:1]:=TOG	20746000
ELSE IF SENSETOG	20747000
THEN IF LOK:=(M[T+4],[43:2]=1) AND NOT TOG)	20748000
THEN M[T+4],[43:2]:=0	20749000
ELSE IF M[T+4],[43:2]=1	20750000
THEN ELSE M[T+4],[43:2]:=TOG×2	20751000
ELSE BEGIN	20752000
M[T+4],[11:1]:=TOG;	20753000
STREAM(DATE,J:=5);	20754000
BEGIN S1:=LOC DATE; DS:=80CT; END;	20755000
M[T+3],[12:18]:=JUNK;	20756000
END;	20757000
DISKWAIT(T,[CF],-30,T,[FF]);	20758000
\$ SET OMIT = SHAREDISK	20759000
UNLOCKDIRECTORY;	20760000
\$ POP OMIT	20761000
\$ SET OMIT = PACKETS	20762000
IF RSTOG THEN	20763000
\$ POP OMIT	20764000
IF LOK THEN P(DIRECTORYSEARCH(=CMM[2],CMM[3],6),DEL)	20765000
ELSE LBMESS(CMM[2],CMM[3],IF TOG THEN 12 ELSE 11,	20766000
13+(SENSETOG×47)-(FXTOG×3),0,SPOUTUNIT,RSTOG)	20767000



```

LINK[SWAPEND]+MIX; 21012600
LINK[SWAPEND+MIX]+0; 21012610
END 21012700
$ SET OMIT = NOT(STATISTICS) 21012709
END 21012710
$ POP OMIT 21012711
END BRINGBACK; 21012800
SAVE PROCEDURE MCPIN(S); VALUE S; REAL S; 21013000
BEGIN REAL T,X,Y,Z; 21015000
IF NOT M[S],[7:1] THEN 21016000
SLEEP([M[S]],0&1[7:47:1]); 21017000
IF (Y+NFLAG(M[S]),[CF])=(*P(,ESPBIT)),[CF] THEN 21018000
BEGIN Z+Y,[8:10]; M[S],[7:1]+0; 21019000
T+GETSPACE(Z,65,0)+2; 21020000
$ SET OMIT = NOT(AUXMEM) 21020010
IF Y,[6:1] THEN % STORED ON AUXMEM 21020020
DISKWAIT(-T,Z,=(0&Y[32:21:12])) 21020030
ELSE 21020040
$ POP OMIT 21020041
DISKWAIT(-T,Z,Y,[FF]+MCPBASE); 21021000
M[T-2],[2:1]+0; 21022000
M[T-1]+S&Z[CTF]; 21023000
M[S] := (*P(DUP)) & T[CTC] & 1[7:47:1]; 21024000
$ SET OMIT = MONITOR OR NOT(AUXMEM) 21024100
AUXTRACE(0,S); 21024200
$ POP OMIT 21024201
END; 21025000
END; 21026000
PROCEDURE EXPANDER(MIX,R); 21026500
VALUE MIX,R; 21026600
REAL MIX,R; 21026700
FORWARD; 21026800
BOOLEAN PROCEDURE UNHOOKANDWAIT(MIX,MCP); VALUE MIX,MCP; REAL MIX,MCP; 21027000
BEGIN REAL I,J,K,L,N,MSTART,MEND; 21028000
ARRAY A=MEND[*]; % INTABLE ROW 21028500
ARRAY LINKR=MEND[*], LINKOR=MEND[*,*]; % BED DESCRIPTOR 21028600
LABEL FOUND; 21029000
IF IOCOUNT[MIX] GEQ 0 THEN 21029100
BEGIN 21029200
$ SET OMIT = SHAREDISK 21029249
CLICK+CLOCK+P(RTR)+900; % GIVE HIM 15 SECS, 21029250
$ POP OMIT 21029251
SLEEP([IOCOUNT[MIX]],-0); 21029300
IF IOCOUNT[MIX] GEQ 0 THEN 21029400
BEGIN STASUS[MIX]:=RUNNING; 21029500
IF NOTERMSET(MIX) THEN PRTRW[MIX],[FF] := 19; 21029700
PRTRW[MIX],[PSF] := 1; 21029750
P(1,RTN); 21029800
END END; 21029900
WAITSTORE(0); 21031000
STOREDY[0]+0; 21032000
LINKR+BED1; 21034000
WHILE TRUE DO 21034500
IF LINKR[2]=MIX THEN GO TO FOUND 21035000
ELSE LINKR+LINKR[1]; 21035500
FOUND:J+TOPSK[MIX]+LINKR[FREG],[FF]; 21036000
LINKOR[0,1] + LINKR[1]; 21037000
LINKOR[1,0],[CF] + LINKR[0]; 21038000
L:=LINKR[4]; 21039000
MSTART:=(I:=MEM[0,MLINK1]),[CF]; 21039500

```

```

MEND:=M[I],[FF],[FF];
DO IF (N+M[J]),[CF]>MSTART THEN
  IF N,[CF] LSS MEND THEN
    BEGIN K+MSTART;
      DO BEGIN I+K;
        K+M[I],[CF];
        END UNTIL N,[CF]<K;
        M[N],[FF],[CF]+M[I+1];
        M[J]+ NFLAG((NOT(I+1))INX N);
      END UNTIL (J+N,[FF])<64;
    * HERE BEE DRAGONS . . .

N + MIK + PRT[MIX,8],[CF],[CF]; J + M[0],[CF];
IF N<FENCE THEN BEGIN
DO J + M[I + J],[CF] UNTIL J>N;
IF P(M[I],[3:12],DUP)=@700 OR P(XCH)=@1500 THEN
BEGIN M[K] + NFLAG(*P(DUP))&(N-I-2)[CTC];
  N:=REENTRANT[INTABLEMAP(M[I+1],[8:10]);
  STREAM(N+N AND 3: T=[INTABLE[MIX,N,[36:10]]]);
  BEGIN SI+LOC T; SI+SI+N; SI+SI+N;
    DI+LOC N; DI+DI+6; DS+2 CHR; END STREAM;
  N + POLISH;
END ELSE N + 0;
END ELSE N + 0;
M[K-1] + (*P(DUP))&N[18:38:10];
N + INTRNSC[0]-1; I + J + 0; A + INTABLEROW[MIX];
IF A#0 THEN
BEGIN
I:=2; DO BEGIN; * STEP THRU REENTRANT INTRINSICS
  J:=REENTRANT[INTABLEMAP(I);
  STREAM(I:=J AND 3:T=[A[J],[36:10]]);
  BEGIN SI+LOC I; DI+DI+I; DI+DI+I;
    IF 2 SC#DC THEN TALLY+1; I+TALLY;
  END STREAM;
  IF POLISH THEN
    IF((J:=INTRNSC[I]),[CF] GTR 1023) AND J LSS 0 THEN
      J + OLAY(J,[CF]-2,MIX);
    IF I=2 THEN I:=18 ELSE I:=I+1;
  END UNTIL I GTR 20;
END;
DAT[MIX]+=*P(DUP);
* THERE BEE A SLAIER OF DRAGONS . . .
STOREDY[0]+1;
IF L.[33:7]#0 THEN
BEGIN CLEARANK(L,MCP);
  STABLE[L,[40:8]]:=(*P(DUP))&3[2:46:2];
END;
END UNHOOKANDWAIT;
PROCEDURE HOOKUPMCP(MIX); VALUE MIX; REAL MIX;
BEGIN REAL I,J,N,S,T,U;
  FORK(P,REENTER),(J+TOPSK[MIX])&[T][CTF]&MIX[3:43:5],-2,0,0);
  IF (N+M[J+5]),[33:7]#0 THEN
  IF (STABLE[N,[40:8]]:=(*P(DUP))&2[2:46:2]),STATIONTYPE=TWX
  THEN
  IF TANKS[N,[40:8]],[SOUSE]=0
  THEN IF SLN[MIX]#0 THEN
    BEGIN S+TWO(CLOCK,[40:2]x6)x"+00";
      TWXOUT([S],[CF],8,1&3[1:46:2],N);
    END;
  SLEEP([T],1);

```

```

21039750
21040000
21040100
21041000
21042000
21043000
21044000
21045000
21046000
21047000
21048000
21049000
21050000
21050100
21051000
21052000
21053000
21054000
21054100
21054200
21054300
21054400
21055000
21055100
21056000
21057000
21057100
21057200
21058000
21058500
21059000
21060000
21061000
21062000
21063000
21064000
21065000
21065500
21066000
21066500
21067000
21068000
21093000
21094000
21095000
21096000
21097000
21099000
21100000
21101000
21102000
21103000
21103050
21103100
21103150
21103200
21103300
21103500
21103600
21104000

```

DO IF P(N+M[J],TOP,XCH,DEL) THEN	21105000
BEGIN S+M[N,[FF]],[CF];	21106000
M[N,[FF]],[CF]+0;	21107000
MCPIN(S);	21108000
M[J]+FLAG(M[S] INX N);	21109000
END UNTIL (J+N,[FF])<64;	21110000
IF (J + M[(S + PRT[MIX,8],[CF])=1],[18:10])#0 THEN	21110100
BEGIN N + J INX PRT[MIX,4];	21110200
MAKEPRESENT(P[MIX,PRTROW[MIX] INX N,[8:10]]);	21110300
M[S] + FLAG(*P(DUP))&(P(DUP) INX (N,[FF]))[CTC];	21110400
M[S-1],[18:10] +0;	21110500
END;	21110600
DAT[MIX]+*P(DUP);	21110700
FOR J+0 STEP 1 UNTIL 29 DO	21111000
IF J<16 OR J>19 THEN	21111100
IF RDCTABLE[J],[8:6]=MIX THEN	21111200
IF UNIT[J],[13:5]=31 THEN	21111300
IF ((T+TWO(J)) AND P(RRR))#0 THEN	21111400
BEGIN UNIT[J],[13:5]+16;	21111410
READY+READY AND NOT T;	21111420
RRRMECH+RRRMECH AND NOT T;	21111430
END;	21111440
IF (NT1+PROCLIMIT[MIX]) LEQ 0 OR	21111500
(NT2+ELAPSEDLIMIT[MIX]) LEQ 0 OR	21111510
OLAYCTR[MIX] LEQ 0 THEN	21111520
BEGIN IF (NT1+PRYOR[MIX])>10 THEN NT1+10;	21111530
NT1+(SLN[MIX]*4+COUNT[MIX]-NT1)*8+208;	21111540
NT2+2*NT1;	21111550
OLAYCTR[MIX]+UPOLAY(NT1)+	21111560
DONTXPANDBITS[MIX]*@10000000000;	21111570
END;	21111580
PROCLIMIT[MIX]+PROCTIME[MIX]+NT1;	21111590
ELAPSEDLIMIT[MIX]+IOTIME[MIX]+NT2;	21111600
STATUS[MIX]+RUNNING;	21111610
\$ SET OMIT = NOT(STATISTICS)	21111619
TIMING[MIX]+CLOCK+P(RTR);	21111620
\$ POP OMIT	21111621
\$ SET OMIT = NOT(SHAREDISK)	21111629
FOR I+0 STEP 1 UNTIL LQAVAIL=1 DO	21111630
IF LQUE[I],[12:1] THEN	21111640
IF LOCATQUE[S+LQUE[I],[1:7]],[3:5]=MIX THEN % IO TO DO	21111650
BEGIN	21111660
\$ SET OMIT = DFX OR OMIT	21111669
U+(LOCATQUE[S]+(*P(DUP))&(NOT 0)[CTF]],[12:6];	21111670
IOCOUNT[MIX]+*P(DUP)+1;	21111680
IF UNIT[U],[FF]>1023 THEN	21111690
BEGIN	21111700
UNIT[U]+(*P(DUP))&S[CTF]&S[CTC];	21111710
STARTIO(U);	21111720
END ELSE	21111730
BEGIN	21111740
LOCATQUE[UNIT[U],[CF]],[FF]+S;	21111750
UNIT[U],[CF]+S;	21111760
END;	21111770
\$ POP OMIT	21111771
\$ SET OMIT = NOT(DFX) OR OMIT	21111779
T+IOQUE[S]&6[3:43:5];	21111780
RETURNIOSPACE(S);	21111800
P[MIX]+MIX;	21111810
IOREQUEST(FINALQUE[S],T,LOCATQUE[S]&18[12:42:6]);	21111820



```

P1MIX+0;
$ POP OMIT
IF I LSS (LQAVAIL+LQAVAIL-1) THEN
BEGIN
    STREAM(A+LQAVAIL-1,B+[LQUE[1]]);
    BEGIN SI+B; SI+SI+8; DS+A WDS; END;
    I+I-1;
END;
END;
$ POP OMIT
END HOOKUPMCP;
PROCEDURE SWAPINGIO(MIX,R); VALUE MIX,R; REAL MIX,R;
BEGIN INTEGER S;
    REAL I,J,K,F,T0,T1,T2,T3,B,D,MSTART;
    ARRAY A#B[*];
    NAME N;
    STREAM PROCEDURE MOVE(N,H,T); VALUE N,H,T;
    BEGIN LOCAL I,J;
        SI+LOC N; SI+SI+5; DI+LOC I; DI+DI+7; DS+CHR;
        DI+LOC J; DI+DI+7; DS+CHR;
        SI+H; DI+T;
        I(32(DS+32 WDS; DS+32 WDS; DS+32 WDS; DS+32 WDS));
        J(DS+32 WDS; DS+32 WDS);
        DS+N WDS;
    END SUPERMOVE;
    SUBROUTINE DODISK;
    BEGIN; STREAM(B,J); BEGIN SI+LOC B; DS+8DEC END;
        B+B+S+2;
        D+J&S[CTF]&3[5:46:2]&R[24:47:1];
        N:=K INX [T0];
        IF (N[0] AND IOMASK)=0 THEN SLEEP([N[0]],IOMASK);
        N[0]+0;
        IOREQUEST(NABS(D)&@377[25:40:8],D,[N[0]]&18[12:42:6]);
$ SET OMIT = NOT(STATISTICS)
COUNTUP(15,S);
$ POP OMIT
J+30*S+J+1;
K:=(K+1),[46:2];
END;
SUBROUTINE SWAP;
BEGIN T0:=T1:=T2:=T3:=IOMASK;
    B+DISKSTORE[MIX];
    K+0;S+63;
    WHILE J+1890<I DO DODISK;
    IF (S+I-J-1)>1023 THEN
        BEGIN S+S DIV 30; DODISK; S+I-J-1 END;
    IF S>0 THEN
    BEGIN
$ SET OMIT = NOT STATISTICS
COUNTUP(15,(S+29) DIV 30);
$ POP OMIT
DISKWAIT((J+1)&R[1:47:1],S,B);
END;
N:=T0;
FOR K:=0 STEP 1 UNTIL 3 DO
    IF (N[K] AND IOMASK)=0 THEN SLEEP([N[K]],IOMASK);
END;
J+MSTART+[MEM[MIX,MLINK1]],[CF];
IF SQ[MIX]>0 THEN SLEEP([SQ[MIX]],-0);
SQ[MIX]+=*P(DUP);

```

```

21111830
21111831
21111840
21111850
21111860
21111870
21111880
21111890
21111900
21111901
21111950
21112000
21113000
%R0221114000
21115000
21116000
21116100
21116200
21116300
21116400
21116500
21116600
21116700
21116800
21116900
21117000
21118000
21119000
21120000
%R0221121000
21122000
21122500
21123000
21123099
21123100
21123101
21124000
%R0221125000
21126000
21127000
%R0221128000
21129000
21130000
21131000
21132000
21133000
21134000
21134500
21134900
21135000
21135100
21135500
21136000
21136500
21137000
21137500
21138000
21139000
21139100
21139200

```

IF R THEN	21140000
BEGIN F←J+(I+(LC[MIX]+(EXPAND[MIX]≠1))×CHUNKSIZE+FENCE)-	21141000
ACTLEN[MIX];	21141100
SWAP;	21142000
A←F←MSTART INX M[F+ADDRESSES]; K←0;	21143000
FOR J←F STEP 1891 UNTIL I=1 DO	21144000
BEGIN M[J]←A[K]; K←K+1 END;	21145000
IF I≠J THEN	21146000
IF (S←I←J+1890)>1023 AND (S+S MOD 30)≠0 THEN	21147000
M[I←S]←A[K];	21148000
J←S←F;	21149000
I←MSTART;	21150000
WHILE I≠J DO	21151000
BEGIN WHILE (K←M[J])>0 DO J←K,[CF]←S←I;	21152000
J←J+3;	21153000
MOVE(J←S,S,I);	21154000
S←J;	21155000
I←K,[CF];	21156000
END;	21157000
END INPUT ELSE	21158000
BEGIN	21158100
BEGIN DO IF (K←M[J])<0 OR (K←K,[CF])<J THEN	21159000
BEGIN J←J+3;	21160000
IF I≠F THEN MOVE(J←F←F,I);	21161000
I←I+J←F;	21162000
F←K;	21163000
END UNTIL (J←K)=MSTART;	21164000
A←MEM[MIX,ADDRESSES]; K←0;	21165000
FOR J←MSTART STEP 1891 UNTIL I=1 DO	21166000
BEGIN A[K]←M[J]; K←K+1 END;	21167000
IF I≠J THEN	21168000
IF (S←I←J+1890)>1023 AND (S+S MOD 30)≠0 THEN	21169000
A[K]←M[I←S];	21170000
J←MSTART;	21171000
SWAP;	21172000
ACTLEN[MIX]←I←MSTART;	21173000
END;	21173100
END OUTPUT;	21174000
IF EXPAND[MIX]≠0 THEN EXPANDER(MIX,R);	21174400
SQ[MIX]←*P(DUP);	21174500
END SWAPPING;	21175000
PROCEDURE SWAPPER;	21200000
BEGIN LABEL START,TIMENDL,RDY,L1,L2,WAITSWAPL,EOJL,BOJL,SATISFYL,	21201000
COMMON,BUMP,FORCEL,RR;	21202000
SWITCH STATE←TIMENDL,WAITSWAPL,BOJL,SATISFYL,EOJL,FORCEL;	21203000
REAL I,J,K,TA,TT,MIX,RCW=+0;	21204000
REAL S,L;	21205000
REAL BJOB;	21205100
START:	21206000
IF (MIX←LINK[31])=0 THEN	21207000
BEGIN STASUS[0]←READYSTATE; KILL([RCW] INX NOT 2) END;	21208000
LINK[31]←LINK[MIX];	21209000
IF MIX=SWAPEND THEN SWAPEND←31;	21210000
S←SC[MIX]; L←LC[MIX];	21210100
BJOB:=BATCHJOB[MIX];	21210200
GO TO STATE[STASUS[MIX]];	21211000
TIMENDL:	21212000
\$ SET OMIT = NOT(STATISTICS)	21212099
SWAPS[MIX]:=*P(DUP)+1;	21212100
\$ POP OMIT	21212101

K←RDYRPTEND;	21213000
RDYRPTEND←MIX;	21214000
SLN[MIX]←SLN[MIX]+(SLN[MIX]≠7);	21215000
RR: IF UNHOOKANDWAIT(MIX,1) THEN GO TO START;	21215100
ELAPSEDLIMIT[MIX]:=P(DUP)-IOTIME[MIX];	%R5921215200
PROCLIMIT[MIX]←P(DUP)-PROCTIME[MIX];	21215300
IF DONTEXPANDBITS[MIX] = 0 THEN	21215350
IF OLAYCTR[MIX] LSS 0 THEN EXPAND[MIX]:=3;	%R3821215400
TA←TT←0;	21216000
I←RDYRPT;	21216100
RDY: LINK[MIX]←LINK[K];	21216200
LINK[K]←MIX;	21216300
NLS[MIX]←SLN[MIX]+2;	21216400
IF TA=0 THEN	21217000
L1: BEGIN STASUS[MIX]←TRANSIT;	21218000
SWAPINGIO(MIX,0);	21218100
END;	21218300
GO TO COMMON;	21219000
WAITSWAPL:	21220000
IF UNHOOKANDWAIT(MIX,0) THEN GO TO START;	21221000
ELAPSEDLIMIT[MIX]←0;	21221100
TA←1;TT←0;	21222000
I←WAITSTATE;	21223000
GO TO L1;	21224000
FORCEL:	21224100
\$ SET OMIT = NOT(STATISTICS)	21224109
SWAPOUTS[MIX]:=P(DUP)+1;	%R6321224110
\$ POP OMIT	21224111
IF (K←FORCEND)≠RDYRPTEND THEN RDYRPTEND←MIX;	21224200
PRTROW[FORCEND←MIX],[PSF]←0;	21224300
SLN[MIX]←I+(SLN[MIX]),[45:2];	21224400
IF I=0 THEN ELAPSEDLIMIT[MIX]:=IOTIME[MIX];	%R5921224500
GO TO RR;	21224900
EOJL:TA←TT←1;	21225000
I←STABLE;	21225100
GO TO COMMON;	21225300
BOJL: TT←1;	21226000
I←READYBOJ;	21227000
L2: TA←1;	21228000
J←MIX;	21229000
IF (K←LINK[READYEND])≠0 THEN	21229100
IF NLS[K]=0 THEN J←K ELSE NLS[K]←NLS[K]-1;	21229110
IF (K←READYEND)≠FORCEND THEN FORCEND←J;	21229120
IF RDYRPTEND=READYEND THEN RDYRPTEND←J;	21229130
SLN[MIX]←0;	21229200
READYEND←J;	21230000
J←MIX;	21230100
GO TO RDY;	21231000
SATISFYL:	21232000
TT←0;	21233000
I←READYSTATE;	21234000
GO TO L2;	21235000
COMMON:	21236000
COUNT[MIX]←63;	21237000
STASUS[MIX]←1;	21237100
\$ SET OMIT = NOT(STATISTICS)	21237199
IF (I←READYSTATE) OR (I←RDYRPT) OR (I←READYBOJ) THEN	21237200
QUETIMING[MIX]:=CLOCK+P(RTR);	21237300
\$ POP OMIT	21237301
I←S-1;	21238000

BUMP;	21239000
IF (I+I+1)>L THEN	21240000
IF (TA+EXPAND[MIX])=0 THEN GO TO START	21240100
ELSE IF S<0 THEN GO TO START	21240200
ELSE BEGIN L+I+IF TA THEN L+1 ELSE S-1;	21240300
S+-(TA+TT+1);	21240400
J+MIX;	21240500
END;	21240600
ACTIVE[I]+ACTIVE[I]+TA;	21241000
TOTAL[I]+TOTAL[I]+TT;	21242000
IF BJOB THEN BATCHED[I]=TT GEQ 0;	21242100
IF TA#1 THEN	21243000
BEGIN POSSESS[I]+J+0;	21244000
DO IF (J+LINK[K+J])=0 THEN GO TO BUMP	21245000
UNTIL SC[J]#1 AND LC[J]#I;	21246000
END ELSE IF (NT1+POSSESS[I])#0 THEN	21246100
IF STASUS[NT1]=RUNNING THEN	21246200
BEGIN IF SLNENT1]=0 OR NLS[NT1]=0 THEN GO TO BUMP;	21246300
IF PRTRW[NT1],[PSF]=0 THEN	21246400
PRTRW[NT1],[PSF]+3;	21246500
GO TO BUMP;	21246600
END ELSE	21247000
IF STASUS[NT1]#RDYRPT THEN GO TO BUMP	21247100
ELSE	21247110
BEGIN NT2+0;	21247120
WHILE (NT2+LINK[NT2])#J DO	21247200
IF NT1=NT2 THEN GO TO BUMP;	21247300
COUNT[NT1]+COUNT[NT1]+63;	21247400
END;	21247500
POSSESS[I]+J;	21248000
COUNT[J]+NT1+COUNT[J]+1 AND 63;	21249000
IF SC[J]+NT1#LC[J] THEN GO BUMP;	21250000
IF STASUS[J]#READYBOJ THEN	21252000
BEGIN STASUS[J]+TRANSIT;	21253000
\$ SET OMIT = NOT(STATISTICS)	21253009
READYQUETIME[J]:=(P(DUP))+CLOCK+P(RTR)-QUETIMING[J];	21253010
\$ POP OMIT	21253011
SWAPINGIO(J,1);	21253100
HOOKUPMCP(J);	21254000
END ELSE STASUS[J]+SELECTING;	21259000
\$ SET OMIT = NOT(STATISTICS)	21259099
IF STASUS[J] = SELECTING THEN	21259100
BEGIN	21259200
INITIALRQTIME[J]:=READYQUETIME[J]+CLOCK+P(RTR)-	21259300
QUETIMING[J];	21259400
READYQUETIME[J]:=QUETIMING[J]:=0;	21259500
END;	21259600
IF SWAPDELAY[J] NEQ 0 THEN	21259650
BEGIN COUNTUPBY(33,CLOCK+P(RTR)-SWAPDELAY[J]);	21259700
SWAPDELAY[J]:=0; COUNTUPBY(34,1);	21259800
END;	21259900
\$ POP OMIT	21259901
LINK[K]+LINK[J];	21260000
IF J=FORCEND THEN FORCEND+K;	21260800
IF J=RDYRPTEND THEN RDYRPTEND+K;	21260900
IF J#READYEND THEN GO TO BUMP;	21261000
READYEND+K;	21262000
GO TO BUMP;	21263000
END SWAPPER;	21264000
PROCEDURE INITIALSWAP(N); VALUE N; REAL N;	21265000

```

BEGIN REAL I,J,K,S,T,U;                                21266000
  LABEL L;                                              21266100
  IF N LSS 0 THEN                                       21266200
  BEGIN K:=N,[27:6]; N:=N,[21:6] END                   21266300
  ELSE BEGIN                                           21266400
  IF (N+(64*N-(N#0)) DIV CHUNKZIZE)>CHUNKMAX THEN N*CHUNKMAX; 21267000
  N ← (N#0)+N;                                         21267100
  U:=N+N LSS CHUNKMAX;                                 %R1121267200
L: S:=4096;                                             %R1121268000
  FOR I ← CHUNKMAX STEP -1 UNTIL N DO                 21269000
  BEGIN T←0;                                           21270000
    FOR J ← 0 STEP 1 UNTIL N DO                       21271000
      IF (NT1:=ACTIVE[I-J]*64+TOTAL[I-J]) GTR T THEN T:=NT1; 21272000
      IF T=U LSS S THEN BEGIN S:=T; K:=I END;         %R1121273000
    END;                                               21274000
  IF S GEQ 4095 THEN BEGIN N:=N-1; GO TO L END;      21274100
  END;                                                 21274200
  SC[P1MIX]←K-N;                                       21275000
  LC[P1MIX]←K;                                         21276000
  CANTEXPAND[P1MIX]:=N=CHUNKMAX;                      21276050
$ SET OMIT = NOT(STATISTICS)                          21276099
  SWAPDELAY[P1MIX]:=CLOCK+P(RTR);                    21276100
$ POP OMIT                                             21276101
  SWAP(BOJSTATE,0);                                   21277000
  SLEEP([SQ[P1MIX]],0&SELECTING[18:42:6]);           21278000
  MEMROW[P1MIX]:=MEMROW[0]&((T:=(K-N)*CHUNKZIZE+FENCE))[CTC]; 21279000
  U:=T+4;                                              21279500
  M[S←(I+(N+1)*CHUNKZIZE)+T-3]+K+T&U[CTF]&1[2:47:1]& 21280000
    P1MIX[9:42:6];                                     21281000
  M[T]+K&U[CTC]&S[CTF]&1[17:47:1];                   21281500
  M[U]+K&S[CTC]&T[CTF];                               21281600
  M[MEM[P1MIX,AVAIL]+S+S+1]+S&@77777[CTF];          21282000
  M[S+1]+S;                                           21283000
  MEM[P1MIX,LEFTLIT]:=T;                              21283500
  FORGETSPACE(U+2);                                   21284000
  FRONTEND(P1MIX);                                    21285000
  IF LOGLINE,[33:7]#0 THEN                            21291100
  BEGIN                                               21291200
  TANKS[LOGLINE,[40:8],[CF]+I+GETSPACE(30,0,1)+2; 21291300
  M[I]←0;                                             21291400
  END;                                               21291500
  END INITIALSWAP;                                    21292000
PROCEDURE FRONTEND(MIX);                               21293000
  VALUE MIX;                                          21294000
  REAL MIX;                                           21295000
  BEGIN REAL I,J;                                     21296000
  J←(I+NOT FENCE INX 1) DIV 1890+2;                 21297000
  IF MEMROW[MIX],[CF] NEQ 0 THEN                      %R7521299900
  MEM[MIX,ADDRESSES]←[M[GETSPACE(J,0&MIX[CTF],1)+1]]&64[8:38:10]; 21302000
  I←GETSPACE(30,0&MIX[CTF],1)+2;                   21304000
  MOVE(30,JARROW[MIX],I);                             21305000
  FORGETSPACE(JARROW[MIX]);                          21306000
  JARROW[MIX],[CF]+I;                                 21307000
  I←GETSPACE(UVSIZE,0&MIX[CTF],1)+2;                21308000
  MOVE(UVSIZE,UVROW[MIX],I);                        21309000
  FORGETSPACE(UVROW[MIX]);                          21310000
  UVROW[MIX],[CF]+I;                                 21311000
  END FRONTEND;                                       21312000
PROCEDURE EXPANDER(MIX,R);                             21313000
  VALUE MIX,R;                                       21314000

```

REAL MIX,R;	21315000
BEGIN REAL I,J;	21316000
LABEL L,L1,SOL;	21317000
IF R THEN	21318000
BEGIN	21319000
IF EXPAND[MIX] THEN	21320000
BEGIN MEM[MIX,0],[FF]+I+(J+MEM[MIX,AVAIL]-1)+CHUNKSIZE;	21321000
M[I]+M[J]&J[CTF];	21322000
M[J]+(*P(DUP))&I[CTC];	21323000
M[(M[I+I+1]+M[J+1])+1]+I;	21324000
M[M[I+1]+M[J+2]],[CF]+MEM[MIX,AVAIL]+I;	21325000
END ELSE	21327000
BEGIN I+(J+[MEM[MIX,0]],[CF])=CHUNKSIZE;	21328000
M[(M[I]+M[J]&(I+4)[CTC]],[FF]],[CF]+I;	21329000
M[J]+(*P(DUP))&(I+4)[CTF];	21330000
M[I+4]+M[I]&J[CTC]&I[CTF];	21331000
M[I+1]+M[J+1]; M[I+2]+M[J+2];	21332000
MEMROW[MIX]+MEMROW[0]&I[CTC];	21333000
FORGETSPACE(I+6);	21334000
FORGETSPACE(J+6);	21335000
FRONTEND(MIX);	21337000
END;	21338000
FORGETSPACE(J+2);	21338100
EXPAND[MIX]=0;	21339000
END ELSE	21340000
BEGIN J:=LC[MIX];	21340500
IF (I:=SC[MIX])=0 THEN	21341000
IF TOTAL[J+1] LSS 63 THEN	21341500
L1: BEGIN LC[MIX]+LC[MIX]+1; I+1 END	21342000
ELSE GO TO SOL	21342500
ELSE BEGIN	21343000
IF J#CHUNKMAX THEN	21343500
IF TOTAL[I-1] GTR TOTAL[J+1] THEN GO TO L1;	21344000
IF TOTAL[I-1] = 63 THEN	21344500
SOL: BEGIN CANTEXPAND[MIX]:=1; I:=0; GO TO L1 END;	21345000
SC[MIX]+I-1; I+2;	21346000
END;	21347000
IF COUNT[MIX] GEQ CHUNKMAX=1 THEN CANTEXPAND[MIX]:=1;	21347500
EXPAND[MIX]:=1;	21348000
IF MAXCORE[MIX] THEN	21349000
BEGIN STREAM(J:=JARROW[MIX], MIX, DI:=I:=SPACE(10));	21350000
BEGIN DS:=LIT" "; SI:=J; SII:=SI+1; DS:=7 CHR;	21351000
SII:=SI+1; DS:=LIT"/"; DS:=7 CHR; DS:=LIT"=";	21352000
SII:=LOC MIX; DS:=2 DEC; J:=DI;	21353000
DII:=DI-2; DS:=FILL; DI:=J;	21354000
DS:=19 LIT" EXPANDED (NO MEM)+";	21355000
END;	21356000
SPOUTER(I,PSEUDOMIX[MIX],1);	21357000
END;	21359000
END;	21360000
END EXPANDER;	21361000
REAL SECONDCTR, LASTSCHEDSELECT;	22000000
\$ SET OMIT = NOT(SHAREDISK)	22000499
PROCEDURE FINDFREEADDRESS(N);VALUE N;REAL N;FORWARD;	22000500
REAL FINDFREECTR; % USED TO DETERMINE HOW FREQUENTLY	22000600
% FINDFREEADDRESS IS BEING CALLED	22000700
\$ POP OMIT	22000701
PROCEDURE NSECOND;%	22001000
BEGIN REAL RCW#+0, I,J,JJ,S; % J MUST BE AT F+2 (SEE 22049400)	22002000
ARRAY A[*];	22003000

\$ SET OMIT = NOT SHAREDISK	22003990
REAL KLUDGE,HOLDER,NEXTSLOT,BYPASS;	22004000
\$ POP OMIT	22004010
BOOLEAN W;	22005000
LABEL TRYWY,DOIT;	22005100
\$ SET OMIT = TWXONLY	22005199
LABEL LOOP;	22005200
\$ POP OMIT	22005201
IF (J:=TOGGLE,NOMEM)#0 THEN	22010000
TOGGLE,NOMEM:=IF J THEN 6 ELSE J-2;	22011000
J:=NEUP,NEUF ;	22011100
\$ SET OMIT = NOT(SHAREDISK )	22011190
DISKWAIT(=(EUIO INX 0),(J+EUIOFFSET),EUIOHOLDER);	22011200
\$ POP OMIT	22011201
FOR I:=J-1 STEP -1 UNTIL 0 DO	22011300
BEGIN	22011400
EUIO[I+EUIOFFSET]:=*P(DUP)*EUTAPER+PEUIO[I];	22011500
PEUIO[I]:=0;	22011600
END;	22011700
\$ SET OMIT = NOT(SHAREDISK )	22011790
DISKWAIT(EUIO INX 0,(J+EUIOFFSET),EUIOHOLDER) ;	22011800
\$ POP OMIT	22011801
WHILE XCLOCK+P(RTR) GEQ WITCHINGHOUR DO MIDNIGHT;	22012000
CHANGEDATE(0);	22013000
FOR I + 20 STEP 1 UNTIL 21 DO%	22032000
BEGIN IF NOT UNIT[I],[16:1] THEN%	22033000
UNIT[I],[17:1] + 0;%	22034000
STARTIO(I);%	22035000
END;%	22036000
\$ SET OMIT = NOT(DFX)	22036099
DISKOUNT+(P(RRR),[29:1] AND (UNIT[18],[FF]>1023))+	%DFX22036100
(P(RRR),[28:1] AND (UNIT[19],[FF]>1023));	%DFX22036200
\$ POP OMIT	22036201
IF SYSDISKADR NEQ 0 THEN	22036300
BEGIN	22036400
A:=IOQUE&SPACE(10)[CTC];	22037000
SYSDISKIO(1,0,A);	22037500
FOR I:=1 STEP 1 UNTIL STATIONMAX DO	22038000
BEGIN SYSDISKIO(1,I,A);	22039000
IF SCH(A) THEN	22039100
BEGIN IF STABLE[I],DIALEDUP THEN	22039150
BEGIN IF (W:=SEQARRAY[I],[CF]) GTR 511 THEN	22039200
IF (W:=M[W+37]&M[W+77][CTF]) NEQ A[3] THEN	22039250
BEGIN A[3]:=W; SYSDISKIO(0,I,A) END;	22039300
END ELSE	22039350
IF NOT SCHEDBUSY[I] THEN S:=S+1;	22039400
END ELSE	22039450
IF REMOTE THEN	22039470
BEGIN	22039500
W:=FALSE;	22039550
IF A[1]#0 OR A[0],DIALEDUP THEN	22040000
BEGIN IF (JJ:=STABLE[I]),ACTIVITY OR I=ABS(SPOWORD)	22040500
THEN W:=TRUE ELSE	22041000
IF A[1]#0 THEN	22042500
BEGIN	22042600
IF A[2]+A[3]<CLOCK THEN	22043000
BEGIN IF (J:=JJ,MIXNR) # CANDEMIX[I] THEN	22044000
TERMINATE(J&15[CTF])	22045000
ELSE BEGIN M[J+GETAREA(0) +	22045100
O&I[CTF]&10[18:41:7];	22045200

```

M[J+1] ← 54; 22045300
  QUEVENT(J,CANDEMIX[I]); 22045400
  END; 22045500
M[J:=GETAREA(0)]:=0&I[25:40:8]&54[18:41:7]; 22047000
  IF JJ,STATIONTYPE=TWX THEN 22047050
    SEQARRAY[I]:=0; %R27 22047100
    QUEVENT(J,CANDEMIX[I]); 22048000
    W←TRUE; 22048100
  END ELSE 22048200
  IF PAPERTAPE[I] THEN % END THE TAPE IF STA 22048300
  IF A[2]+A[3]/4 LSS CLOCK THEN % IS IDLE, 22048350
  BEGIN M[J:=GETAREA(0)]=(*P(DUP))&I[10:40:8] 22048400
    &I[5:47:1]; 22048450
    STREAM( 22048500
      A:=LINEDISC(IF I LEQ LMAX THEN I ELSE 22048550
        STABLE[I],LEENKER) ≠ TWX, 22048560
      J:=J+1); 22048600
    BEGIN 22048650
      A(DS:=LIT" "; DS:=LIT"≠"); 22048695
      DS:=LIT MARK; 22048700
      A(DS:=2 LIT"≠#"); 22048705
      DS:=LIT"≠"; 22048750
      DS:=LIT"≠"; 22048795
      DS:=LIT"≠"; 22048800
      DS:=LIT"≠"; 22048850
    END; 22048850
    GIVEAWAY(J); 22048900
  END; 22048950
  END; 22049000
  END ELSE 22049100
  IF A[2]+3600<CLOCK THEN 22049200
  BEGIN 22049250
    IF BLASTREAD(I,7) THEN 22049300
    BEGIN STREAM(J:=JJ:=(RCW INX 2)); 22049350
      DS:=8 LIT"≠BYE≠*S"; 22049400
      TWXOUT(JJ,8,-1,I); 22049450
    END ELSE A[0],DIALEDUP:=0; 22049500
    W:=TRUE; 22049550
  END; 22049600
  IF W THEN 22049700
  BEGIN A[2]:=CLOCK; 22049800
    SYSDISKIO(0,1,A); 22049900
  END; 22049950
  P([STABLE[I]],PRL); 22050000
  END; 22050100
  END; 22050150
  END; 22050200
  SYSDISKIO(1,STAMAX+1,A); 22050250
  FORGETSPACE(A); 22050300
  END; 22050302
  $ SET OMIT = TWXONLY 22050304
  IF REMOTE THEN 22050305
  IF (I:=CLOCK+P(RTR)) GTR LINETABLE[0] THEN 22050310
  BEGIN %UPDATE TIMER TO CHECK AGAIN IN A MINUTE 22050315
    LINETABLE[0]:=I+1800; 22050320
    FOR I:=1 STEP 1 UNTIL LMAX DO 22050325
    IF LINEDISC[I]=MULTI THEN 22050330
    BEGIN JJ:=LINETABLE[I],DIRECTLINE; 22050335

```



```

LOOP:          J:=1; W:=FALSE;                22050340
DO IF P(SEQARRAY[J],DUP),[3:1]=JJ THEN          22050345
BEGIN IF JJ THEN                               22050350
    P(P&O[3:47:1],[SEQARRAY[J]],SND);         22050355
    W:=TRUE;                                   22050360
END UNTIL (J:=P,LINELINK)=1;                  22050365
IF (JJ:=JJ=W) THEN IF W THEN ENTEREADYQ(I) ELSE 22050370
GO TO LOOP; % IF DIAL-UP AND NO ONE ON LINE  22050375
END; END;                                     22050380
$ POP OMIT                                     22050381
IF (XCLOCK=LASTSCHEDESELECT)GEQ 54000 THEN    22050400
BEGIN LASTSCHEDESELECT:=(XCLOCK DIV 54000)*54000; 22050500
    IF S GTR 0 THEN                           22050600
        IF SCHEDNUM NEQ FRSTSCHEDE THEN      22050700
            FORK(P(,SCHEDIDLE),0,0,160,0);    22050800
    END;                                       22050900
$ SET OMIT = NOT(STATISTICS)                  22050909
COUNTARRAY[1]:=CLOCK;                       22050910
IF SYSTATBASE GTR 0 THEN BEGIN               22050915
DISKIO([JUNK],COUNTARRAY,[CF]-1,60,SYSTATBASE); 22050920
IF XCLOCK GEQ COUNTARRAY[29] THEN           22050930
BEGIN                                         22050940
    COUNTARRAY[29]:=XCLOCK+INTERVAL;         22050960
    FORK(P(,FILLSYSTAT),0,0,128,1);          22050970
END END;                                     22050980
$ POP OMIT                                     22050981
$ SET OMIT = NOT(SHAREDISK)                  22050999
DISKWAIT(=[HOLDER],[CF]),=3,DIRECTORYSEG);  22051100
$ SET OMIT = NOT STATISTICS OR OMIT          22051140
BYPASSBOTTOM:=BYPASS,[CF];                 22051150
$ POP OMIT                                     22051160
IF (J:=HOLDER,[FF])>0 THEN % HOLD LIST IS NOT EMPTY 22051200
BEGIN                                         22051250
    A:=[M[SPACE(J)]&J[8:38:10]];             22051300
    DISKWAIT(=(A INX 0),J,HOLDER,[CF]);      22051400
    FOR I:=JJ:=0 STEP 1 UNTIL J-1 DO         22051500
        IF A[I] LSS 0 THEN                   22051600
            IF A[I],[2:2]=SYSNO THEN         22051700
                BEGIN IF (JJ:=A[I]),[FF] GEQ FENCE THEN 22051800
                    BRINGBACK(JJ,[10:8]) ELSE M[JJ,[FF]]:=1; 22051900
                    A[I]:=P(DUP,LOD,SSP);    22052000
                END;                          22052100
            IF JJ>0 THEN DISKWAIT(A INX 0,J,HOLDER,[CF]); 22052200
            FORGETSPACE(A);                  22052300
        END;                                  22052350
    UNLOCK(DIRECTORYSEG);                   22052400
    IF NOT FINDINGADDRESS THEN              22052700
        IF LQAVAIL>0 THEN                   22052800
            BEGIN                             22052900
                FINDINGADDRESS:=1;          22053000
                FINDFREEADDRESS(0);         22053100
            END;                               22053200
        IF FINDFREECTR LSS 32 THEN% IT IS NOT GETTING CALLED OFTEN 22053300
        IF (J:=(*P(,FINDFREEADDRESS))INX 0)>M[0],[CF] THEN 22053400
            M[J-2],[2:1]:=0;                 22053500
        FINDFREECTR:=0;                      22053600
        IF RUNUMBER GTR 0 THEN STARTADECK(0); 22053650
$ POP OMIT                                     22053651
IF CLOCK,[35:7]=0 THEN                       22053660
FOR S:=MIXMAX-1 STEP -1 UNTIL 1 DO          22053670

```

```

IF JARROW[S]≠0 THEN
BEGIN TABCNT[S]:=TABCNT[S]+1;
  IF (I:=REPLY[S]) LSS 0 THEN
  BEGIN
    IF PUTORTAKE(S,[PRYOR[S]],1,0) LSS 1023 THEN
    BEGIN
      TRYWY:
      JJ:=VFM;
      STREAM(I:JJ:=[JJ]);
      BEGIN SI:=LOC I; SI:=SI+1; DI:=DI+7;
        7(IF SC=DC THEN
          BEGIN I:=TALLY; JUMP OUT END;
            DI:=DI-1);
        END;
        IF JJ=VFM THEN % FIRST TIME THRU
          IF P=0 THEN % FM
          BEGIN JJ:=VWY;
            GO DOIT
          END ELSE % NO FM
          BEGIN JJ:=VOK;
            GO TRYWY
          END ELSE % NOT FIRST TIME THRU
          IF P NEQ 0 THEN % IMPROPER MSG
            IF JJ=VOK THEN % TRY WY IF OK IS NOT
            BEGIN JJ:=VWY;
              GO TRYWY;
            END
            ELSE ELSE % MSG IS OK
            DOIT:
            BEGIN REPLY[S]:=JJ;
              BRINGBACK(S);
            END END END;
            TABCNT[S]:=TABCNT[S]-1;
          END;
          NSECONDREADY:=TRUE;
          SECONDCTR:=0;
          KILL([RCW] INX NOT 2);
        END;%
        PROCEDURE STATUS;%
          BEGIN REAL U:=+1,%
            T=U+1,%
            T1=T+1;%
            INTEGER%
            I=T1+1;%
            ARRAY AREA=I+1[*];%
            REAL HDR = AREA+1,
              SEGO= HDR + 1,
              F = SEGO+1;
            ARRAY SHEAT = F+1[*];
            LABEL TRYAGAIN,LDCNTRL,DISK;
            LABEL L,EL,NOTREADY,DIE,ACCEPT,SCRATCH,INPUT,TESTBACKUP,
              COMMON;
            LABEL CARD,PRINTER,TAPE,DRUM,DISC,SPO,PUNCH,UNLD,
              PAPERPUNCH,PAPER,DATAKOM;
            SWITCH S := CARD,PRINTER,TAPE,DRUM,DISC,SPO,PUNCH,UNLD,
              PAPERPUNCH,PAPER,DATAKOM;%
            REAL RCW:=+0;%
          SUBROUTINE SPACE;%
            BEGIN AREA I= [M[SPACE(12)]]&10[8:38:10]; END;
          SUBROUTINE AUTOLOADER;
            BEGIN
22053672
22053674
22053680
22053690
22053715
22053720
22053725
22053730
22053735
22053740
22053745
22053750
22053753
22053754
22053755
22053756
22053757
22053758
22053759
22053760
22053761
22053762
22053765
22053770
22053775
22053780
22053785
22053788
22053790
22053795
22053800
22053820
22053830
22053850
22053900
22053950
22054000
22055000
22056000
22057000
22058000
22059000
22060000
22061000
22061100
22061110
22061120
22061130
22061200
22062000
22063000
22064000
22064500
22065000
22066000
22067000
22068000
22069000
22069010
22069020

```

```

TRYAGAIN:
  IF (HDR:=DIRECTORYSEARCH(P(LDCNTRL),P(DISK),3)) # 0 THEN
  BEGIN
    SHEAT := [M[F:=GETSPACE(31,64,0)+2]] & 30[8:38:10];
    STREAM(S:=F-1, D:=F); % ZERO OUT THE SHEAT ENTRY
    BEGIN
      SII=S; DS:=30 WDS;
    END;
    SEGO := GETSPACE(30,64,0)+2;
    DISKWAIT(=SEGO, 30, M[HDR INX 10]);
    F,[FF] := HDR; % CORE ADDRESS OF HEADER IN [FF] OF PARAM,
    SHEAT[7] := SEGO; % CORE ADRS,OF SEGMENT ZERO IN SHEAT[7]
    SHEAT[0] := SHEAT[14] := P(LDCNTRL);
    SHEAT[1] := P(DISK);
    SHEAT[2] := 0 & 5[8:38:10] & 1[4:47:1];
    % [4:1] IN SHEET[2] MEANS SUPPRESS BOJ/EQJ MESSAGES
    SHEAT[16] := SHEAT[17] := @377777777777; % TIME LIMITS
    SHEAT[19] := U; % COMMON VALUE
    SHEAT[20] := 4; % CORE ESTIMATE
    SHEAT[21] := 150; % STACK SIZE

    STREAM(A:=0 : S := P(,SCHEDULEIDS));
    BEGIN
      SII=S;
      47(SKIP SB; SKIP DB; TALLY:=TALLY+1;
      IF SB THEN ELSE JUMP OUT);
      DS:=SET; A:=TALLY;
    END STREAM STATEMENT;

    I := P;
    SHEAT[3],[8:10] := I; % SCHEDULE NUMBER
    SHEAT[23] := (CLOCK + P(RTR)) DIV 60;
    SHEAT[24] := MCP;
    SHEAT[25] := HDR,[FF]; % DISK ADDRESS OF FILE HEADER
    SHEAT[26] := "31"; % LOGLINE
    STREAM(U, I:=I:=GETSPACE(10,0,0)+2);
    BEGIN
      DS:=27LIT"CC RUN LDCNTRL/DISK;COMMON=";
      SI:=LOC U; DS:=8DEC;
      DS:=6LIT";END,+";
    END STREAM STATEMENT;
    SHEAT[6] := GETESPDISK;
    DISKWAIT(1, 10, SHEAT[6]);
    FORGETSPACE(I);
    MULTITABLE[U] := "CONTROL";
    LABELTABLE[U] := ("DECK ");
    RDCTABLE[U] := 1 & 1[14:38:10];
    IF U THEN READERA:=0 ELSE READERB:=0;
    FORK(P(,SELECTRUN), F, 0, 160, 0);
    END ELSE % IF IN DIRECTORY
    BEGIN
      ENTERSYSFILE(2);
      GO TRYAGAIN;
    LDCNTRL::: "LDCNTRL";
    DISK::: "DISK ";
    END;
  END SUBROUTINE AUTOLOADER;

P(0,0,0,0,0,0,0,0,0,0);%

```

```

22069025
22069030
22069040
22069050
22069060
22069070
22069080
22069090
22069100
22069110
22069120
22069130
22069140
22069150
22069160
22069170
22069180
22069190
22069200
22069210
22069220
22069230
22069240
22069250
22069260
22069270
22069280
22069290
22069300
22069310
22069320
22069330
22069335
22069340
22069350
22069360
22069370
22069380
22069390
22069400
22069410
22069420
22069430
22069440
22069450
22069460
22069470
22069480
22069490
22069510
22069520
22069530
22069540
22069550
22069560
22069570
22069600
22070000
22071000
22072000

```

```

SPACEA;% 22073000
WHILE (T ← P(RRR) OR RRRMECH) ≠ READY DO% 22074000
  BEGIN I ← 0&TINU[U ← (P(T EQV NOT READY,DUP,DUP,x,x)% 22075000
    x@10000000000000),% 22076000
    [3:6]][5:11:7]/@1000000000000;% 22077000
  IF T < READY THEN% 22078000
    BEGIN COMMENT SOMETHING WENT NOT READY;% 22079000
      READY ← READY AND NOT I;% 22080000
      IF LABELTABLE[U] ≥ 0 THEN% 22081000
        BEGIN% 22082000
          LABELTABLE[U] ← @114;% 22083000
          IF (U AND @774) NEQ 16 THEN 22084000
            MULTITABLE[U]:=0; 22084500
        END;% 22085000
      RRRMECH ← RRRMECH AND NOT I;% 22086000
    END OF NOT READY% 22087000
  ELSE BEGIN COMMENT SOMETHING WENT READY;% 22088000
    READY ← READY OR I;% 22089000
    UNIT[U],[13:1] ← 0;% 22090000
    IF LABELTABLE[U]≠@114 OR UNIT[U],[13:5]=15 THEN 22091000
      BEGIN RRRMECH ← RRRMECH OR I;% 22092000
        IF LABELTABLE[U] = @214 THEN% 22093000
          BEGIN I ← I AND NOT SAVEWORD;% 22094000
            GO TO L;% 22095000
          END;% 22096000
        IF STATUS[T+RDCTABLE[U],[8:6]]= 22096100
          RUNNING OR T=0 THEN 22096200
          BEGIN T1←UNIT[U],[FF]; 22096300
            WHILE T1<1023 22096310
              DO BEGIN T1←LOCATQUE[T1],[FF]; 22096400
                IOCOUNT[T1]*P(DUP)+1; 22096500
              END;% 22096600
              UNIT[U],[13:5]←0; 22096610
              STARTIO(U); 22096700
            END ELSE 22096800
            BEGIN IF REPLY[T]=(-VOK) THEN 22096900
              BEGIN REPLY[T]←0; 22097000
                BRINGBACK(T); 22097100
              END;% 22097200
              UNIT[U],[13:5]←31; 22097300
            END;% 22097400
            GO TO COMMON;% 22098000
          END;% 22099000
        IF (U AND @774) NEQ 16 THEN 22100000
          MULTITABLE[U]:=RDCTABLE[U]:=0; 22100500
        IF (I AND SAVEWORD) ≠ 0 THEN% 22101000
          BEGIN RRRMECH ← I AND SAVEWORD OR RRRMECH; 22102000
            GO TO COMMON;% 22103000
          END;% 22104000
          GO S[UNIT[U],[1:4]];% 22105000
        TAPE: P(WAITIO(@4200000000,5,U),DEL);% 22106000
        IF (T ← WAITIO(AREA INX @120540000000,@7500045,U)),% 22107000
          [45:3] ≠ 0 THEN% 22108000
          NOTREADY: BEGIN READY ← READY AND NOT I;% 22109000
            GO TO L;% 22110000
          END;% 22111000
          IF MOD3IOS AND NOT T,[42:1] THEN BEGIN %AI22111500
            DO UNTIL (T1←WAITIO(AREA INX @340000012,@55,U))≠0;%AI22112000
          END ELSE T1←WAITIO(@4200000000,5,U); %AI22112500
          IF T1,[45:3]≠0 THEN GO TO NOTREADY; %AI22113000

```

DO UNTIL NOT (T1+WAIT10(@500000000,@165,U)) OR	22114000
(TRANSACTION[U]+0);%	22115000
IF T1,[42:1] THEN	22115020
BEGIN; STREAM(T+TINU[U],A+AREA);	22115030
BEGIN SI+LOC T; SI+SI+5; DS+LIT"#";	22115040
DS+3 CHR; DS+10 LIT "-BAD LOAD";	22115050
END;	22115060
SPOUT(AREA INX 0); SPACEA; GO TO L;	22115070
END;	22115075
IF T1,[45:1] THEN GO TO NOTREADY;%	22116000
PRNTABLE[U]+0&(NOT T1)[1:43:1];%	22117000
IF T,[43:1] THEN%	22118000
BEGIN;STREAM(T+TINULUJ,AREA);%	22119000
BEGIN SI + LOC T; SI + SI+5;%	22120000
DS + LIT "#"; DS + 3 CHR;%	22121000
DS + 14 LIT " PARITY, RW/L=";	22122000
END;%	22123000
DIE; SPOUIT(AREA INX 0,HRDWREK); SPACEA; %PAR,RW/L,	22124000
LABELTABLE[U] + @314;%	22125000
GO TO EL;%	22126000
END;%	22127000
IF T,[42:1] THEN%	22128000
BEGIN;STREAM(T+TINU[U],AREA);%	22129000
BEGIN SI + LOC T; SI + SI+5;%	22130000
DS + LIT "#"; DS + 3 CHR;%	22131000
DS + 15 LIT " TAPE MK, RW/L=";	22132000
END;%	22133000
GO TO DIE;%	22134000
END;%	22135000
STREAM(Y+0:AREA,X+[T]);%	22136000
BEGIN DS + 8 LIT " LABEL ";%	22137000
SI + AREA; DI + DI-8;%	22138000
IF 8 SC = DC THEN TALLY + 1;%	22139000
AREA + TALLY;%	22140000
SI + SI+45; DI + LOC Y; DS + 5 OCT;%	22141000
SI + LOC AREA; DI + X; DS + WDS;%	22142000
END;%	22143000
NT1 + P;%	22144000
IF T THEN PRNTABLE[U],[30:18];=NT1 ELSE	22145000
BEGIN STREAM(Y:=0:AREA,X:=[T]);	22145050
BEGIN DS:=4 LIT "VOL1";	22145100
SI:=AREA; DI:=DI-4;	22145150
IF 4 SC=DC THEN TALLY:=1;	22145200
AREA:=TALLY; SI:=SI+1;	22145250
DI:=LOC Y; DS:=5 OCT;	22145300
SI:=LOC AREA; DI:=X; DS:=WDS;	22145350
END;	22145400
NT1:=P;	22145450
IF T THEN BEGIN PRNTABLE[U],[30:18];=NT1;	22145500
USASITAPE([AREA],[CF],T,1,U,1);	22145550
END;	22145600
END;	22145650
IF NOT T1,[43:1] THEN%	22146000
BEGIN IF T THEN%	22147000
BEGIN	22148000
IF P(AREA[1],DUP)="PBTMCP " OR	22156000
P(XCH)="PUTMCP " THEN GO INPUT;	22156100
IF AREA[4],[12:30] > DATE THEN%	22157000
BEGIN IF RETMSG THEN	22158000
STREAM(T+TINU[U],A+[AREA[6]]);	22159000

```

ACCEPT:
        BEGIN SI←LOC T;SI←SI+5;DS←3 CHR;22160000
            DS←5 LIT " RET "; 22161000
        END ELSE GO TO INPUT; 22162000
T1 := SPACE(4); 22163000
STREAM(A+[AREA[1]],T1);% 22164000
        BEGIN SI ← A; SI ← SI+40;% 22165000
            DS ← LIT "#";% 22166000
            DS ← 8 CHR; SI ← A;% 22167000
            2(DS ← LIT " ";% 22168000
            SI ← SI+1; DS ← 7 CHR);22169000
            DS ← LIT "+";% 22170000
        END;% 22171000
        SPOUT(T1);% 22172000
        GO TO INPUT;% 22173000
        END ELSE% 22174000
SCRATCH: LABELTABLE[U] ← 0;% 22175000
        END ELSE GO TO UNLD; 22176000
        END;% 22177000
ELSE IF T THEN BEGIN% 22178000
INPUT: LABELTABLE[U] ← AREA[2];% 22179000
        MULTITABLE[U] ← AREA[1];% 22180000
        STREAM(A+[AREA[3]],B+[T]);% 22181000
            BEGIN SI ← A; DS ← 3 OCT;% 22182000
                DS ← 5 OCT; DS ← 2 OCT;% 22183000
            END;% 22184000
            RDCTABLE[U] ← I&T1[24;31;17]&T[14;38;10];% 22185000
            IF (MULTITABLE[U]="PBTMCP " OR 22188000
                MULTITABLE[U]="PUTMCP ") AND 22188100
                LABELTABLE[U] = "BACK-UP" THEN% 22189000
                BEGIN LABELTABLE[U] ← @322212342546447;% 22190000
                    STREAM(A+TINU[U],PN+MULTITABLE[U]="PUTMCP ", 22191000
                        AREA); 22191100
                        BEGIN SI ← LOC A; SI ← SI+5;% 22192000
                            PN(DS←3 LIT"#CP"; JUMP OUT TO L);22192100
                            DS←3 LIT"#LP"; LI 22192200
                            DS←12 LIT" BACK-UP ON "; 22193000
                            DS ← 3 CHR; DS ← LIT "+";% 22194000
                        END;% 22195000
                            SPOUT(AREA INX 0); SPACEA; 22196000
                        END;% 22197000
                    END ELSE% 22198000
PAPER;% 22199000
UNLD: LABELTABLE[U] ← @314;% 22200000
        GO TO COMMON;% 22201000
PRINTER;% 22202000
        T ← WAITIO(@6000000000,4,U),[45;1];% 22203000
        UNIT[U],[16;2] ← 0;% 22204000
        IF T THEN GO TO NOTREADY;% 22205000
TESTBACKUP: 22205500
        IF AUTOPRINT THEN 22206000
            IF PRINTORPUNCHWAIT(=U,0) THEN GO TO COMMON; 22207000
        GO TO SCRATCH; 22208000
CARD;% 22209000
        RRRMECH:=RRRMECH OR I; 22209200
        IF CDONLY THEN 22209400
            BEGIN 22209500
                AUTOLOADER; 22209600
                GO TO COMMON; 22209700
            END; 22209800
        LABELTABLE[U]:=-@14; 22212200

```

CCARD(O&U[3:43:5]);	22212400
GO TO COMMON;%	22213000
PUNCH:	22213500
STARTIO(U);	22213600
IF UNIT[U].[15:3]=0 THEN GO TESTBACKUP ELSE GO TO SCRATCH;	22213700
DRUM;%	22214000
DISC;	22215000
SPO;%	22216000
PAPERPUNCH;%	22218000
DATAKOM;%	22219000
STARTIO(U);%	22220000
GO TO SCRATCH;%	22221000
COMMON: END OF READY;%	22222000
END;%	22223000
STATUSBIT + TRUE;%	22224000
FORGETSPACE(AREA,[33:15]);%	22225000
KILL([RCW] INX NOT 2);	22226000
END STATUS;%	22227000
BOOLEAN PROCEDURE OLAY(LOC,MIXX);	22228000
VALUE LOC,MIXX; REAL LOC,MIXX;	22229000
BEGIN REAL LINK, MOM, FRONT, BACK, CHAR, BS, STACK, S, SB,%	22230000
T, X, DESC, DISK, IOD, MIX, JOBKILLED, MIXUP, SEGNO;%	22231000
ARRAY NAME SEGDICT;%	22232000
REAL RESULT=+1;%	22233000
ARRAY SPRT[*];	22234000
REAL CORE, CUED; REAL INITCW=MIXUP;	22235000
REAL TYPE13, RSLT, NOAUX;	22235500
\$ SET OMIT = NOT(NEWLOGGING)	22235599
REAL MCPROCTEMP;	22235600
\$ POP OMIT	22235601
LABEL EXIT; % ALL AVENUES MUST LEAD TO HERE	22235700
LABEL AROUND, CODE, BACKAGAIN, MCP, INTRINSIC;%	22236000
LABEL ZAP;%	%R422236100
LABEL RETRY, AGAIN;	22236110
LABEL SKIPIT;	22236120
BOOLEAN SUBROUTINE AWAKEN;%	22237000
BEGIN COMMENT AWAKEN CHECKS TO SEE IF WE HAVE HALTED	22238000
THE JOB ON PROCESSOR 2, IF SO, IT RESTARTS THE	22239000
TIMING FOR HIM, AND CALLS "HALT" TO CHECK INTERRUPTS;%	22240000
IF JOBKILLED THEN%	22241000
BEGIN	22242000
\$ SET OMIT = NEWLOGGING	22242099
STARTLOG(P2MIX,0);	22242100
\$ POP OMIT	22242101
JOBKILLED + FALSE; OLAY + RESULT OR 2;%	22243000
HALT; NOPROCESSTOG + NOPROCESSTOG=1;%	22244000
END;%	22245000
AWAKEN + RESULT END;%	22246000
SUBROUTINE STOP;%	22247000
IF NOT JOBKILLED THEN	22247100
BEGIN COMMENT STOP HALTS THE JOB ON PROCESSOR 2, AND	22248000
CLOCKS HIM OFF, IT SETS JOBKILLED SO THAT AWAKEN	22249000
CAN DO ITS DIRTY WORK BEFORE RETURNING;%	22250000
JOBKILLED + TRUE; P(HP2);%	22251000
STOPLOG(P2MIX,0);	22252000
END STOPPER;%	22253000
SUBROUTINE CODEOVERLAY;%	22254000
BEGIN COMMENT CODEOVERLAY HANDLES ALL CASES OF MARKING	22255000
A NORMAL-STATE SEGMENT AS NOT-PRESENT, IT DOES THIS	22256000
A SINGLE PRT AND STACK AT A TIME, AND IS ONLY CALLED	22257000

```

REPEATEDLY FOR RE-ENTRANT CODE OR INTRINSICS;% 22258000
IF CHAR THEN S ← M[SB ← M[S],[FF]],[FF] ELSE S ← S-1;% 22259000
SPRT ← PRT[MIX,10];% 22260000
IF SPRT[X],[2:1] THEN BEGIN% 22261000
% NEED TO DO PRT AND STACK SEARCH ONLY IF PRESENT IN THIS PRT 22262000
DO UNTIL (X ← (SPRT[X] ← (*P(DUP))&0[2:2:1])% 22263000
&(SPRT[X],[CF]=FRONT)[CTC]),[6:12])≥2048;% 22264000
AROUND;% %R422265000
WHILE (STACK := HUNT(BS),[CF]) LSS S DO 22266000
BEGIN CORE ← (DESC ← NFLAG(M[STACK]),[CF]);% 22267000
IF CORE≥FRONT AND CORE≤BACK THEN% 22268000
IF DESC LSS 0 THEN%PROG, DESC OR RCW, 22269000
IF DESC,[3:1] THEN%DESC 22270000
IF DESC,[2:1] THEN%PRESENT 22270050
IF DESC,[6:2]=1 THEN %TYPE 13 INTRINSIC DESC 22270100
M[STACK]:=FLAG(DESC & 0[2:2:1] 22270200
& (MOM,[8:10])[CTC]) ELSE 22270300
% DESCRIPTOR == INSERT OFFSET AND RESET P-BIT 22271000
M[STACK] ← FLAG(DESC&0[2:2:1])% 22272000
&(CORE=FRONT)[CTC])% 22273000
ELSE 22273100
ELSE BEGIN% 22274000
% CONTROL WORD (RCW) == UNFLAG IN STACK, PUT OFFSET INTO 22275000
% CORRESPONDING MSCW, AND MOM INTO RCW,[CF] 22276000
M[X ← DESC,[FF]] ←% 22277000
(*P(DUP))&(CORE=FRONT)[CTC];% 22278000
M[STACK] ← DESC&SEGNO[CTC];% 22279000
END;% 22280000
BS ← STACK+1;% 22281000
END;% 22282000
IF CHAR AND (STACK<SB) THEN% 22283000
BEGIN BS ← SB; S ← HUNT(BS+1),[CF]; GO AROUND END; 22284000
IF P(SPRT[19],TOP) THEN P(DEL) ELSE %DS22284100
BEGIN CORE:=POLISH,[CF]; %DS22284200
IF CORE LSS FRONT OR CORE GTR BACK THEN %DS22284300
ELSE SPRT[19]:=(*P(DUP))&0[2:2:1] %DS22284400
&(CORE=FRONT)[CTC]; %DS22284500
END; %DS22284600
% SET OMIT = NOT(STATISTICS) 22284699
CODEOLAYS[MIX]:=*P(DUP)+1; %R622284700
% POP OMIT 22284701
END OF PRESENT IN PRT CASE;% 22285000
END OF CODEOVERLAY;% 22286000
SUBROUTINE INT13; %STACK SEARCH FOR TYPE 13 INTRINSIC CALLS 22286010
BEGIN CHAR:=P(PRT[MIX,8],DUP),[32:1]; 22286020
S:=P INX OF BS:=PRT[MIX,10],[FF]; 22286030
IF CHAR THEN S:=M[SB:=M[S],[FF]],[FF] ELSE S:=S-1; 22286040
AGAIN; WHILE (STACK := HUNT(BS),[CF]) LSS S DO 22286050
BEGIN CORE:=(DESC:=NFLAG(M[STACK]),[CF]);% 22286060
IF CORE GEQ FRONT AND CORE LSS BACK THEN% 22286070
IF DESC,[1:2] NEQ 0 THEN 22286075
IF DESC,[1:3]=7 THEN% 22286080
M[STACK]:=FLAG(DESC&0[2:2:1]&(MOM,[8:10])[CTC])22286090
ELSE 22286100
BEGIN M[DESC,[FF]]:=(*P(DUP))&(CORE=FRONT)[CTC];22286110
M[STACK]:=DESC&(MOM,[8:10])[CTC]& 22286120
1[33:47:1]; 22286130
END; 22286140
BS:=STACK+1; 22286150
END; 22286160

```



```

        IF CHAR AND (STACK LSS SB) THEN%           22286170
        BEGIN BS:=SB; SI:=HUNT(BS+1),[CF]; GO AGAIN; END; 22286180
END OF TYPE 13 INTRINSIC STACK SEARCH;           22286190
        COMMENT OLAY HANDLES OVERLAYS, THERE ARE 3 CLASSES 22287000
        OF THINGS WHICH MAY BE OVERLAID:          22288000
        1) OBJECT PROGRAM DATA SEGMENTS          22289000
        2) OBJECT PROGRAM CODE SEGMENTS          22290000
        AND 3) MCP (NON-SAVE) PROCEDURES,         22291000
        EACH OF THESE CLASSES GETS SPECIAL HANDLING, 22292000
        WHICH WILL BE DESCRIBED AS WE COME TO IT;  22293000
% THIS CODE IS COMMON TO ALL CLASSES AND ALL CASES 22294000
$ SET OMIT = NOT(NEWLOGGING)                     22294099
        IF P1MIX>0 THEN                            22294100
        IF NOT LOGSTOPPED[P1MIX] THEN             22294200
        IF NOT MCPROCTIME[P1MIX],[1:1] THEN       22294300
        BEGIN                                       22294400
                MCPROCTEMP+PROCTIME[P1MIX]+CLOCK+P(RTR); 22294500
                MCPROCTIME[P1MIX]+NABS(*P(DUP));        22294600
        END;                                       22294700
$ POP OMIT                                       22294701
        LINK + M[LOC]; MOM + M[LOC+1];%           22295000
        FRONT + LOC+2; BACK + LINK,[CF]-1;%      22296000
        IF (MIX + LINK,[9:6])=0 THEN GO TO MCP;%  22297000
% <MIX>=0 AND NON-SAVE MEANS MCP PROCEDURE OR INTRINSIC 22298000
        IF MIX=P2MIX THEN STOP;%                 22299000
        CHAR + (INITCW + PRT[MIX,8]),[32:1];%     22300000
        S + INITCW,[CF]; BS + PRT[MIX,10],[FF];%  22301000
% CHAR IS CWMF, S IS TOP-OF-STACK, BS IS BASE OF STACK 22302000
        IF P(LINK,[3:6],DUP)=1 OR P(XCH)=13 THEN GO TO CODE; 22303000
% TYPE=1 MEANS PROGRAM, 13 MEANS TYPE 13 INTRNSC, OTHERWISE DATA, 22304000
        IF TERMSET(MIX) THEN GO TO ZAP;%         %R4222304100
        IF M[MOM],[CF] NEQ FRONT THEN %BAD MOM DESC, %10722304200
        BEGIN% TERMINATE THIS GUY, %10722304300
                PRTROW[MIX],[FF]=32; %MEMORY ERROR, %10722304400
                PRTROW[MIX],[PSF]=1; %10722304500
                GO TO ZAP; %10722304600
        END BAD MOM; %10722304700
        IF CHAR THEN%                             22305000
% SPECIAL CHECKS FOR ADDRESS SAVED IN CHARACTER MODE 22306000
        BEGIN CHAR + ((CT + M[S-1],[CF])>FRONT AND T<=BACK) OR% 22307000
% M=REGISTER FROM ICW (SOURCE ADDRESS)           22308000
                (CT + M[S-2],[FF])>FRONT AND T<=BACK));% 22309000
% S=REGISTER FROM ILCW (DESTINATION ADDRESS)     22310000
        IF NOT CHAR THEN%                         22311000
        BEGIN X + M[S + M[S],[FF]],[FF]+1;%       22312000
% M[S],[FF] IS ADDRESS OF RCW, M[RCW],[FF] IS ADDRESS OF MSCW 22313000
        DO CHAR + (CT + M[S + S-1],[CF])>FRONT 22314000
                AND T<=BACK) UNTIL (S<=X) OR CHAR;% 22315000
% SEARCH THROUGH STREAM LOCALS AND PARAMETERS FOR ADDRESSES 22316000
        S + X;%                                    22317000
        END;%                                       22318000
        END;%                                       22319000
        IF CHAR THEN                               22320000
        BEGIN P(AWAKEN); GO EXIT;                 22320100
        END;                                       22320200
% CANNOT OVERLAY IF MAY BE ADDRESSES IN CHAR MODE STACK 22321000
        IOD+M[MOM],[8:10];                         22322000
        IF (DISK:=MOM,[FF]) NEQ 0 THEN % OLAY ADDRESS PRESENT 22323000
        BEGIN                                       22323200
$ SET OMIT = NOT(AUXMEM)                         22323400

```

IF DISK,[33:3] = 7 THEN % AUXMEM ADDRESS	22323600
IF UNIT[16+DISK,[36:1]],[14:1] THEN % PREVIOUS ERROR	22323800
BEGIN	22324000
DISK:=0; NOAUX:=(=0); % DONT USE AUXMEM	22324200
END;	22324400
\$ POP OMIT	22324600
MOM,[FF]:=0;	22324800
END;	22325000
IF DISK=0 THEN DISK:=DISKSPACE(IOD,MIX,NOAUX);	22325200
IF DISK LSS 0 THEN % NO OLAY DISK	22325400
BEGIN P(AWAKEN); GO EXIT;	22325410
END;	22325430
\$ SET OMIT = NOT(STATISTICS)	22325600
DATAOLAYS[MIX]:=*P(DUP) + 1;	22325800
COUNTUP(21+(FRONT GTR FENCE),(IOD+29) DIV 30);	22326000
\$ POP OMIT	22326200
IF (S:=S-1) GTR MOM THEN IF MOM GTR BS THEN BS:=MOM+1;	22326400
% IF MOTHER IS IN STACK, ONLY SEARCH ABOVE IT	22326600
WHILE (STACK:=HUNT(BS),[CF]) LSS S DO	22326800
BEGIN	22327000
IF (DESC:=NFLAG(M[STACK]),[1:2]=1 THEN	22327200
% ONLY WORRY ABOUT DATA DESCRIPTORS WHICH ARE PRESENT	22327400
IF DESC,[FF]=MOM THEN	22327600
% THIS ONE DEMANDS ACTION -- IT POINTS INTO OUR ARRAY	22327800
IF DESC,[8:10]=0 THEN	22328000
% NAME DESCRIPTOR -- PUT IN OFFSET AND RESET P=BIT	22328200
M[STACK]:=FLAG((DESC,[CF]"FRONT"&MOM[CTF])	22328400
ELSE	22328600
% NORMAL ROW DESCRIPTOR -- ZERO CORE FIELD AND RESET P=BIT	22328800
M[STACK]:=FLAG(0&DESC[8:8:25]);	22329000
BS:=STACK+1;	22329200
END;	22329400
IF M[MOM],[3:3] NEQ 7 THEN % NOT READ ONLY, ALREADY WRITTEN	22329600
BEGIN	22330600
\$ SET OMIT = NOT(AUXMEM)	22330800
M[FRONT-1] := MIX & IOD[CTF]; % AUXMEM LINK	22331000
\$ POP OMIT	22331200
RETRY:	22331400
DISKIO(RSLT,FRONT-1,IOD&1[3:47:1],DATADDRESS(MIX,DISK));	22331600
% [3:1] IN SIZE MARKS I/O AS ORIGINATING FROM OLAY	22331800
M[MOM]:=(*P(DUP))&0[2:47:1]&5[CTC];	22332000
P(AWAKEN,DEL);	22332200
% CF=5 IN MOTHER IS INTERLOCK FOR MAKEPRESENT	22332400
SLEEP([RSLT],IOMASK);	22332600
IF RSLT,[26:7] NEQ 0 THEN % I/O ERROR	22332800
BEGIN	22333000
\$ SET OMIT = NOT(AUXMEM)	22333200
IF DISK,[33:3]=7 THEN % AUXMEM ADDRESS	22333400
AUXDATA[MIX]:=(*P(DUP)) - IOD,[38:6] -1;	22333600
\$ POP OMIT	22333800
IF (DISK:=DISKSPACE(IOD,MIX,=0)) LSS 0 THEN	22334000
BEGIN % NO OLAY DISK	22334200
M[MOM]:=(*P(DUP))&6[CTC]; % TERMINATE MARKER	22334400
GO TO ZAP;	22334600
END;	22334800
GO TO RETRY; % TRY AGAIN WITH ANOTHER ADDRESS	22335000
END; % IF I/O ERROR	22335200
M[MOM]:=(*P(DUP))&DISK[CTC]; % PUT DISK ADDRESS IN MOTHER	22335400
END % IF READ ONLY, NOT YET WRITTEN	22335600
ELSE	22335800
BEGIN	

```

M[MOM]:=(*P(DUP))&0[2:47:1]&DISK[CTC]; 22336000
P(AWAKEN,DEL); 22336200
END; 22336400
IF M[MOM],[3:3]=6 THEN M[MOM],[5:1] := 1; 22349100
ZAP: FORGETSPACE(FRONT);% %R422350000
P(TRUE); GO EXIT; 22351000
CODE:;% %R422352000
% OBJECT PROGRAM CODE TO BE OVERLAID 22353000
IF (T + M[S],[CF])≥FRONT AND T≤BACK THEN% 22354000
% CANNOT OVERLAY NORMAL STATE SEGMENT HE WILL RETURN TO 22355000
BEGIN P(AWAKEN); GO EXIT; 22356000
END; 22356020
IF LINK,[3:6]=13 THEN %TYPE 13 INTRNSC 22356100
BEGIN 22356200
INT13; 22356300
TYPE13:=INTABLEROW[MIX],[8:10]=1; S:=MOM,[8:10]; 22356400
FOR X:=INT13START STEP 1 UNTIL TYPE13 DO 22356500
IF INTABLE[MIX,X],[FF]=S THEN % ZERO OUT TYPE 13 ENTRY 22356600
BEGIN INTABLE[MIX,X]:=0; TYPE13:=0; END; %JUMP OUT 22356700
FORGETSPACE(FRONT); 22356800
P(AWAKEN,DEL,TRUE); GO EXIT; 22356810
END; 22356900
IF (MIXUP + (SEGDICT + PRT[MIX,4]),[FF])≠0 THEN% 22357000
% RE-ENTRANT CODE TO BE OVERLAID == CHECK OTHER USERS. TOO 22358000
BEGIN MIXUP + MIXUP,[39:6];% 22359000
DO BEGIN% 22360000
IF MIXUP=P2MIX THEN STOP;% 22361000
% STOP OTHER USER OF THIS CODE IF RUNNING ON PROCESSOR 2 22362000
IF (T + M[PRT[MIXUP,8]],[CF])≥FRONT AND T≤BACK% 22363000
THEN BEGIN P(AWAKEN); GO EXIT; 22364000
END; 22364100
% SAME CRITERIA APPLY TO ALL USERS OF THIS CODE 22365000
END UNTIL (MIXUP + PRT[MIXUP,4],[24:6])=077;% 22366000
% CHECK ALL USERS ON MIX=INDEX LINKED LIST 22367000
END;% 22368000
% IF WE REACH THIS POINT, WE CAN AND WILL OVERLAY THE AREA 22369000
$ SET OMIT = NOT(AUXMEM) 22369999
IF ((SEGDICT[SEGN0:=MOM,[CF]]),[2:4])=2 THEN 22370000
% TO GO TO AUXILIARY MEMORY, HAS NOT YET BEEN WRITTEN THERE 22370100
IF (DISK := AUXILIARYSPACE(MOM,[FF]))>0 THEN 22370200
BEGIN % TRY TO WRITE TO AUXMEM 22370300
T := SEGDICT[SEGN0]&0[3:45:3]&MOM[FTF]; % SAVE FOR AUX,ERR, 22370400
M[FRONT=1]:=NABS(MIX&MOM[FTF]&T[3:33:15]); % AUXMEM LINK 22370500
SEGDICT[SEGN0]:=(*P(DUP))&1[5:47:1]&DISK[CTC]; 22370600
DISK := CODEADDRESS(MIX,SEGDICT[SEGN0]); 22370700
IF (CUED:=IOQUESLOTS≠0) THEN 22370800
DISKIO(RSLT,FRONT=1,(MOM,[FF]&1[3:47:1]),DISK); %OLAY 1/0 22370900
END; 22371000
SEGDICT[SEGN0],[3:1]:=1; 22371100
$ POP OMIT 22371101
BACKAGAIN:; 22371200
$ SET OMIT = AUXMEM 22371210
X:=SEGDICT[SEGN0+MOM],[8:10];CODEOVERLAY; 22371220
$ POP OMIT 22371221
$ SET OMIT = NOT(AUXMEM) 22371299
X:=SEGDICT[SEGN0],[8:10]; CODEOVERLAY; 22371300
$ POP OMIT 22371301
IF MIXUP THEN% 22372000
% RE-ENTRANT CODE BEING OVERLAID == MUST FIX ALL STACKS AND PRTS 22373000
IF (MIX + PRT[MIX,4],[24:6])≠077 THEN% 22374000

```

```

% SET UP CHAR, S, AND BS FOR NEXT USERS STACK                22375000
  BEGIN CHAR ← (S ← PRT[MIX,8]),[32:1];%                    22376000
    S ← S INX 0; BS ← PRT[MIX,10],[FF];%                    22377000
% GO DO STACK SEARCH AND PRT FIX=UP FOR ANOTHER USER        22378000
  GO TO BACKAGAIN;%                                         22379000
  END;%                                                       22380000
$ SET OMIT = NOT(AUXMEM)                                     22380049
  P(AWAKEN,DEL);                                           22380050
  IF DISK,[1:1] THEN % AUXMEM WRITE                          22380100
    BEGIN                                                    22380200
      IF NOT CUED THEN                                       22380300
        DISKIO(RSLT,FRONT=1,(MOM,[FF]&1[3:47:1]),DISK); % OLAY I/O 22380400
        IF (RSLT AND IOMASK)=0 THEN SLEEP([RSLT],IOMASK);   22380500
        IF RSLT,[26:7] NEQ 0 THEN SEGDICT[SEGN0];=T ELSE % OLD VALUE 22380600
          BEGIN                                               22380700
            SEGDICT[SEGN0];=(*P(DUP))&MOM[FTF]&0[3:47:1];   22380800
            AUXCODE[LINK,[9:6]] := (*P(DUP)) + MOM,[23:6] + 1; 22380900
          END;                                                 22381000
        END % IF AUXMEM WRITE                                  22381400
      ELSE SEGDICT[SEGN0];=(*P(DUP))&0[3:47:1]&MOM[FTF];   22381500
      FORGETSPACE(FRONT); P(TRUE); GO EXIT;                  22383000
$ POP OMIT                                                  22383001
$ SET OMIT = AUXMEM                                         22383049
  SEGDICT[MOM];=(*P(DUP))&MOM[FTF];%                        22383050
  FORGETSPACE(FRONT); P(AWAKEN,DEL,TRUE); GO EXIT;         22383100
$ POP OMIT                                                  22383101
% NOW WAS THAT NOT TRIVIALITY PERSONIFIED,..              22384000
MCP:;%                                                      %R4222385000
  IF P(LINK,[3:6],DUP)=7 OR P(XCH)=13 THEN GO TO INTRINSIC; 22386000
  SPRT ← P(,BED); BS ← (P(,RDF)),[FF];                     22387000
% SET BS TO POINT AT RCW FOR CALL ON OLAY                  22388000
  DO BEGIN%                                                 22389000
    OLAY ← NOT(S ← ((CORE ← (T ← M[BS]),[CF])≤BACK          22390000
      AND CORE≥FRONT));%                                     22391000
% S IS TRUE IF THE RCW POINTS TO THE ROUTINE TO BE OVERLAID 22392000
  BS ← T,[FF];%                                             22393000
% POINT T TO CORRESPONDING MSCW                             22394000
  WHILE (T ← M[BS]),[16:1] DO BS ← T,[FF];%                22395000
% RUN DOWN STACK OF MSCWS UNTIL NOT MSFF                   22396000
  IF (BS ← T,[FF])≤64 THEN                                  22397000
% END OF STACK -- THIS IS RATIONALE FOR OBSCURE USE OF "P(,STF)" 22398000
  BEGIN S ← S OR (SPRT ← FLAG(M[SPRT]),[CF])=P(,BED);      22399000
    BS ← SPRT[FREG],[FF];                                    22399100
  END END UNTIL S;                                          22400000
  IF RESULT THEN%                                           22401000
  BEGIN M[MOM] ← (*P(DUP))&(*P(,ESPBIT))[CTC];%            22402000
    FORGETSPACE(FRONT);%                                     22403000
  END;%                                                      22404000
  P(RESULT AND 1); GO EXIT;                                  22405000
INTRINSIC:;%                                                %R4222406000
  IF MIXX=0 THEN BEGIN                                       22406100
    FOR MIX←1 STEP 1 UNTIL MIXMAX DO%                        22407000
      IF DAT[MIX]<0 THEN                                     22407500
        IF INTABLEROW[MIX]≠0 THEN%                          22408000
          BEGIN IF MIX=P2MIX THEN STOP;%                    22409000
            IF (T ← M[PRT[MIX,8]),[CF])≥FRONT AND T≤BACK% 22410000
              THEN BEGIN P(AWAKEN); GO EXIT;                22411000
            END;                                             22411020
          END;%                                               22412000
    END;%                                                     22413000
    FOR MIX←1 STEP 1 UNTIL MIXMAX DO%

```

IF DAT[MIX]<0 THEN	22413500
IF INTABLEROW[MIX]≠0 THEN%	22414000
BEGIN SEGNO ← MOM,[8:10]-1;%	22415000
IF MEMROW[MIX] INX 0 GEQ FENCE THEN	22415100
IF NOT INTRNSC[SEGNO+1],[4:1] THEN GO TO SKIPIT ELSE	22415150
SEGNO:=REENTRANTINTABLEMAP(SEGNO+1);	22415200
STREAM(A ← SEGNO AND 3; T ← [INTABLE[MIX,SEGNO DIV 4]]);	22416000
BEGIN SI ← T; SI ← SI+A; SI ← SI+A; DI ← LOC A;%	22417000
DI ← DI+6; DS ← 2 CHR; END STREAMING;%	22418000
IF (SEGNO ← POLISH)≠0 THEN%	22419000
IF SEGNO = @2000 THEN INT13 ELSE	22419500
BEGIN CHAR ← P(PRT[MIX,8], DUP),[32:1];%	22420000
TYPE13:=SEGNO,[37:1];	22420200
SEGNO:=SEGNO AND @1777; %IGNORE TYPE 13 BIT	22420500
S ← POLISH INX 0; BS ← PRT[MIX,10],[FF];%	22421000
SEGDICT ← PRT[MIX,4];%	22422000
X:=SEGDICT[SEGNO],[8:10];	22423000
IF TYPE13 AND NOT PRT[MIX,X],[2:1] THEN	22423100
% TYPE 13 REFERENCE ALSO EXISTS AND TYPE 7 REFERENCE IS NOT PRESENT	22423200
INT13 ELSE	22423300
BEGIN	22423400
CODEOVERLAY;	22423500
SEGDICT[SEGNO] ← (*P(DUP))&MOM[FTF];%	22424000
END;	22424500
END;%	22425000
SKIPIT: END;	22426000
INTRNSC[MOM,[8:10]] ← (*P(DUP))&MOM[FTC];%	22427000
FORGETSPACE(FRONT);%	22428000
P(AWAKEN,DEL,TRUE); GO EXIT;	22429000
END NORMAL CASE;	22430000
SEGNO:=MOM,[8:10]; MIX:=MIXX;	22431000
SEGNO:=REENTRANTINTABLEMAP(SEGNO);	22431100
STREAM(A←SEGNO AND 3; T←[INTABLE[MIX,SEGNO,[36:10]]]);	22432000
BEGIN SI←T; SI←SI+A; SI←SI+A; DI←LOC A;	22433000
DI←DI+6; T←SI; DS←2 CHR; DI←T; DS←2 LIT "00";	22434000
END STREAM;	22435000
SEGNO ← POLISH; SEGDICT ← PRT[MIX,4];	22436000
CHAR ←P(PRT[MIX,8], DUP),[32:1];	22437000
S ←POLISH INX 0; BS ←PRT[MIX,10],[FF];	22438000
X ← SEGDICT[SEGNO],[8:10]; CODEOVERLAY;	22439000
SEGDICT[SEGNO] ← (*P(DUP))&MOM[FTF];	22440000
P(0);	22441000
EXIT:	22441100
\$ SET OMIT = NOT(NEWLOGGING)	22441199
IF MCPROCTEMP≠0 THEN	22441200
BEGIN	22441300
MCPROCTEMP←PROCTIME[PIMIX]+CLOCK+P(RTR)-MCPROCTEMP;	22441400
IF MCPROCTEMP<0 THEN MCPROCTEMP←0;	22441500
MCPROCTIME[PIMIX]←ABS(*P(DUP))+MCPROCTEMP;	22441600
END;	22441700
\$ POP OMIT	22441701
P(RTN);	22441800
END OF OVERLAY;% REVISION OF 5 JANUARY 1968 . . .	22442000
REAL SPACESTACK;	23399000
SAVE PROCEDURE FORGETSPACE(LOC);%	24000000
VALUE LOC;%	24001000
REAL LOC;%	24002000
BEGIN%	24003000
REAL B←BACK,F←FRONT,LINK,X,T,SIZE;%	24004000
REAL MIX;	24004100

```

DEFINE LEFTOFF=MEM[MIX,LEFTLIT]#;
MIX*(LINK+M[LOC+(*P(,LOC)),[CF]=2]),[9:6];
$ SET OMIT = NOT(CHECKLINK OR DEBUGGING)
IF CHECK THEN CHECKLINKS(MIX,LOC);
$ POP OMIT
IF P(MEM[MIX,MLINK1],DUP),[CF] GTR LOC OR P(XCH),[FF] LSS LOC
OR (B:=M[BACK:=LINK,[FF]]),[CF] NEQ LOC
OR (F:=M[FRONT:=LINK,[CF]]),[FF] NEQ LOC
OR LINK LSS 0 THEN PUNT(4); % INVALID LINK
IF F LSS 0 THEN
    BEGIN%
        M[LOC]*LINK &F[CTC];%
        M[F]*M[F] & LOC[CTF];%
        M[T+M[FRONT+2]]*M[T] &(X+M[FRONT+1])[CTC];%
        M[X+1]*T%
    END;%
IF B LSS 0 THEN
    BEGIN%
        M[BACK]*B&(T+M[LOC],[CF])[CTC];%
        M[T]*M[T]& BACK[CTF];%
        M[BACK+1]*M[BACK+1]&(SIZE+T-BACK-2)[CTF];%
    END;%
ELSE%
    BEGIN%
        M[LOC+1]*(T+M[MEM[LOC+2]+MEM[MIX,AVAIL]])&
        (SIZE+M[LOC],[CF]-LOC-2)[CTF];
        M[MEM[MIX,AVAIL]]*T&(M[T+1]+LOC+1)[CTC];
        M[LOC]*NABS(*P(DUP));
    END;%
IF LOC<LEFTOFF THEN IF M[LOC],[CF]>LEFTOFF THEN LEFTOFF=M[LOC],[FF];
$ SET OMIT = NOT(CHECKLINK OR DEBUGGING)
IF CHECK THEN CHECKLINKS(MIX,LEFTOFF);
$ POP OMIT
END FORGETSPACE;%
SAVE REAL PROCEDURE ACTSPACE(SIZE,SAVEF,MIX);
VALUE SIZE,SAVEF,MIX;
REAL SIZE,SAVEF,MIX;
BEGIN REAL LINK,LOC,X,Y,T,SIZEF;
REAL LOS=SIZEF+1,MSTART=LOS+1;
DEFINE LEFTOFF=MEM[MIX,LEFTLIT]#;
LABEL GOTIT;
LABEL NEWSTART, SVSEARCH, ROCKABYE, START, OVERLAY;%
LABEL OVSEARCH,XX;
P(0,MEMROW[MIX],[CF],PRIORITY); % SET UP LOS, MSTART, AND TEMP
PRIORITY*=-5;
IF SAVEF THEN% ATTEMPT TO ALLOCATE AT START OF MEMORY
IF SAVEF<4 OR TOGLE,NOMEM#0 THEN
    BEGIN LINK:=M[MSTART];
SVSEARCH;%
IF (LOC+LINK,[CF])=MSTART THEN
    GO TO ROCKABYE;
IF (LINK + M[LOC])>0 THEN%
    BEGIN IF NOT LINK,[2:1] THEN%
        BEGIN % OVERLAY ONLY IF POTENTIAL SPACE ADEQUATE
            SIZEF * =2; X * T * LOC;
            IF (Y+LINK,[FF])#MSTART THEN
                IF M[Y] < 0 THEN SIZEF*M[(T+Y)+1],[FF];
            WHILE SIZE>SIZEF AND (Y+M[X]),[1:2]#1 DO

```

24004200  
24005000  
24005099  
24005100  
24005101  
24005200  
24005300  
24005400  
24005500  
24007000  
24008000  
24009000  
24010000  
24011000  
24012000  
24013000  
24015000  
24016000  
24017000  
24018000  
24019000  
24020000  
24021000  
24022000  
24023000  
24024000  
24025000  
24026000  
24027000  
24028000  
24029000  
24030000  
24030099  
24030100  
24030101  
24031000  
24032000  
24033000  
24034000  
24034050  
24034100  
24034200  
24034900  
24035000  
24035050  
24035100  
24035200  
24040000  
24040500  
24041000  
24042000  
24043000  
24044000  
24055000  
24056000  
24056100  
24056200  
24056250  
24056300  
24056400

SIZEF ← SIZEF - X + (X + Y,[CF]);	24056500
IF SIZE > SIZEF THEN	24056600
BEGIN LINK ← Y; GO TO SVSEARCH END;	24056700
IF OLAY(LOC,0) THEN % RE=SET "LINK"	24057000
IF (Y+M[LINK+T])>0 OR Y,[CF]=LOC	24057100
OR M[Y],[FF]≠LINK THEN	24057150
% MEM LINK AT "T" NO LONGER VALID	24057200
LINK ← M[MSTART];	24057300
END;%	24057500
GO TO SVSEARCH;%	24058000
END;%	24059000
IF (SIZEF + M[T+LOC+1],[FF])<SIZE THEN GO SVSEARCH;%	24060000
M[Y:=M[(X:=M[T])+1]:=M[T+1]]:=(*P(DUP))&X[CTC];	24063000
X:=LINK;	24064000
IF SIZEF>SIZE+DELTA THEN%	24065000
BEGIN M[LOC]:=X&(Y:=LOC+SIZE+2)[CTC];	24066000
M[X] ← (*P(DUP))&Y[CTF];%	24067000
M[Y]:= X,[CF]&LOC[CTF]&M[X[9:42:6]];	24068000
FORGETSPACE(Y+2);%	24069000
X,[CF]:=Y;	24069100
END;%	24070000
GO TO GOTIT;	24070500
END;	24071000
START;%	24072000
IF (LINK + P(M[MEM[MIX,AVAIL]],0,SIZE,CFX,LLL,	24073000
0, INX, ,T, STD)),[FF]=@77777 THEN%	24074000
BEGIN%	24075000
OVSEARCH:	24075050
IF (LINK+M[LEFFOFF]),[1:2] = 0 THEN	24075100
BEGIN % OVERLAY ONLY IF POTENTIAL SPACE ADEQUATE	24075150
SIZEF ← -2; X ← LEFFOFF;	24075200
IF (Y+LINK,[CF]) ≠ MSTART THEN	24075300
IF M[Y] < 0 THEN SIZEF ← M[Y+1],[FF];	24075400
WHILE SIZE > SIZEF AND (Y+M[X]),[1:2]≠1 DO	24075500
BEGIN SIZEF←SIZEF+Y,[CF]=X; X+Y,[FF] END;	24075600
IF SIZE > SIZEF THEN	24075700
BEGIN LEFFOFF ← Y,[FF];	24075800
IF P(MSTART,DUP)=LEFFOFF OR P(XCH)=X THEN	24075900
GO TO XX ELSE GO TO OVSEARCH END;	24075950
OVERLAY: %	24075960
OVERLAY ATTEMPTED CYCLICALLY, USING LEFFOFF	24076000
	24076500
	24077000
	24078000
IF OLAY(LEFFOFF,0) THEN GO TO START;	24078500
END;%	24079000
IF (LEFFOFF+LINK,[FF])=MSTART THEN	24080000
IF LOS THEN GO TO ROCKABYE ELSE LOS ← 1;	24080000
GO TO OVSEARCH;	24081000
END;%	24082000
	24083000
IF (SIZEF + LINK,[FF])>SIZE+DELTA THEN%	24084000
BEGIN M[T] ← LINK&(X ← SIZEF-SIZE-2)[CTF];%	24085000
LOC ← T+X+1;%	24086000
X ← (Y + M[T-1])&(T-1)[CTF];%	24087000
M[Y] ← (*P(DUP))&LOC[CTF];%	24088000
M[T-1] ← Y&LOC[CTC];%	24089000
END ELSE BEGIN%	24090000
M[LINK+1] ← Y + M[T+1];%	24091000
M[Y] ← (*P(DUP))&LINK[CTC];%	24092000
X ← M[LOC + T-1];%	24093000
END;%	24094000

GOTIT;	24095000
M[ACTSPACE+LOC]+ABS(X&1[2:47:1]);	24096000
M[LOC+1]+0;	24096500
ROCKABYE;	24096600
PRIORITY+P;	24096700
END ACTSPACE;	24097000
SAVE INTEGER PROCEDURE DISKSPACE(WORDS,MIX,AUX);	24101000
VALUE WORDS,MIX,AUX;	24102000
INTEGER WORDS,MIX; REAL AUX;	24103000
BEGIN ARRAY LOC=+2[*];	24104000
INTEGER	24105000
INDEX=NT1;	24106000
SEG =NT2;	24107000
CNTRS=NT3;	24108000
SIZE =NT4;	24109000
LIMIT=NT5;	24110000
REAL	24110000
T =NT6;	24111000
LABEL	24112000
L1;	24112000
FINAL;	24112500
BADEXIT;	24113000
EXIT;	24114000
DEFINE HEURISTIC = 2#;	24115000
REAL SUBROUTINE FINDSEG;	24115000
BEGIN; STREAM(A+0:T);	24116000
BEGIN SI+LOC T; SI+SI+3;	24117000
5(IF SC="0" THEN JUMP OUT TO L;	24118000
SI+SI+1; TALLY+TALLY+1);	24119000
L: A+TALLY;	24120000
END STREAM;	24121000
FINDSEG + POLISH	24122000
END FINDSEG;	24123000
SUBROUTINE FIND;	24124000
BEGIN POLISH(0);	24125000
T + LOC[INDEX];	24126000
SEG + T,[9:3];	24127000
CNTRS + T,[2:7];	24128000
IF SEG>4 THEN	24129000
L1:  IF (SEG + FINDSEG)=5 THEN GO TO FINAL	24130000
ELSE CNTRS +0;	24131000
IF SIZE+CNTRS>100 THEN GO TO L1;	24132000
P(DEL,(INDEX*256)+SEG*100+CNTRS);	24133000
STREAM(A+0:SEG,T+[T]);	24134000
BEGIN SI+T; SI+SI+3; SI+SI+SEG;	24135000
DI+LOC A; DI+DI+7; SEG+DI;	24136000
T+SI; DS+CHR; TALLY+A;	24137000
TALLY+TALLY+1; A+TALLY;	24138000
SI+SEG; DI+T; DS+CHR;	24139000
END STREAM;	24140000
IF (POLISH=63) OR (CNTRS + CNTRS+SIZE)=100 THEN	24141000
BEGIN CNTRS + 0; SEG + FINDSEG END;	24142000
LOC[INDEX] + T&SEG[9:45:3]&CNTRS[2:41:7];	24143000
FINAL: IF (DISKSPACE + POLISH)≠0 THEN	24144000
BEGIN IF SEG=5 THEN INDEX + 0;	24145000
LOC[0] + LIMIT&INDEX[CTF];	24146000
GO TO EXIT;	24147000
END END FIND;	24148000
\$ SET OMIT = NOT(AUXMEM)	24148999
IF ((AUX OR DATAOLAY) AND NOT AUX,[1:1]) THEN	24149000
IF P(AUXILIARYSPACE(WORDS),DUP) NEG 0 THEN	24149100
BEGIN	24149200
AUXDATA[MIX]:=*P(DUP)+WORDS,[38:6]+1;	24149300



```

P(RTN);
END ELSE P(DEL);
$ POP OMIT
P(DALOC[MIX,*]);
SIZE * (WORDS+29) DIV 30;
IF (LIMIT := LOC[0],[CF])=0 THEN GO TO BADEXIT;
IF (INDEX + LOC[0],[FF])#0 THEN FIND;
INDEX := 2; DO FIND UNTIL (INDEX := INDEX+2)>LIMIT;
BADEXIT;
DISKSPACE * -1;
EXIT;
$ SET OMIT = NOT(STATISTICS)
IF INDEX GEQ OLAYUSED[MIX],[3:15] THEN
OLAYUSED[MIX]:=LOC[INDEX]&INDEX[3:33:15];
$ POP OMIT
STREAM(A+0;L+LIMIT,[41:6],1+[LOC[1]]);
BEGIN SI+T; DI+A;
L(SI+SI+1);
5(IF SC="0" THEN DI+DI+8; SI+SI+1));
A+DI;
END STREAM;
IF (POLISH<HEURISTIC) THEN
IF ((SEG * TWO(MIX)) AND OLAYMASK)#0 THEN
BEGIN OLAYMASK * NOT SEG AND OLAYMASK;
FORK(P(,GETMOREOLAYDISK),MIX,-3,128,1);
IOCOUNT[MIX] * *P(DUP)+1;
END;
END DISKSPACE;
SAVE REAL PROCEDURE GETSPACE(SIZE,TYPE,SAVEF);
VALUE SIZE,TYPE,SAVEF;
REAL SIZE,TYPE;
BOOLEAN SAVEF;
BEGIN REAL MIX,T,MESS;
BOOLEAN BELOW;
LABEL AGAIN,SLEAP,MUSTNOTSWAP;
SUBROUTINE TELLSP0;
BEGIN P(P1MIX); P1MIX:=0;
STREAM(X:=MESS#0, MIX, SIZE, MESS:=MESS:=GETAREA(0));
BEGIN % NOTE THAT 1ST 4 BITS OF MSG ARE ZEROES.
SI:=LOC MIX; DS:=2 DEC;
DS:=8 LIT" NO MEM ";
DS:=5 DEC; DS:=5 LIT" WDS+";
X(DI:=DI-17; DS:=2 LIT"OK");
END;
IOREQUEST(MESS&@274[1:40:8], P(DUP),
[17]&@231[10:40:8]);
P1MIX:=P;
END OF TELLING SPO ABOUT NO MEMS;
BELOW+MEM[MIX+IF TYPE#64 THEN TYPE,[FF] ELSE P1MIX,0]],
[CF] = 0;
$ SET OMIT = NOT(CHECKLINK OR DEBUGGING)
IF CHECK THEN CHECKLINKS(MIX,MEM[MIX,LEFTLIT]);
$ POP OMIT
AGAIN;
WAITSTORE(MIX); STOREDY[MIX]=0;
IF BELOW THEN P(SPACESTACK,STS);
T+ACTSPACE(SIZE,SAVEF,MIX);
IF BELOW THEN P([BELOW],STS);
STOREDY[MIX]=1;
IF T=0 THEN

```

```

24149400
24149500
24149501
24150000
24151000
24152000
24153000
24155000
24155500
24156000
24157000
24157099
24157100
24157200
24157201
24158000
24159000
24160000
24161000
24162000
24163000
24164000
24165000
24166000
24167000
24168000
24169000
24170000
24300000
24301000
24302000
24303000
24304000
24304500
24305000
24305500
24305600
24305700
24305800
24305900
24306000
24306100
24306200
24306300
24306400
24306500
24306600
24306700
24306900
24307000
24307500
24307999
24308000
24308001
24309000
24310000
24311000
24312000
24313000
24314000

```

	BEGIN NOMEMTOG:=1;	24314500
	IF SAVEF.[46:1] THEN P(0,RTN);	24315000
	IF P1MIX=0 OR BELOW THEN	24315100
	BEGIN	24315200
SLEAP:	IF MESS=0 THEN TELLSP0;	24316000
	SLEEP([CLOCK],NOT CLOCK);	24324000
	GO TO AGAIN;	24325000
	END;	24326000
	IF P(PRTROW[P1MIX],DUP)=0 THEN P(DEL) ELSE	24327000
	IF P(DUP).[PSF]=1 OR P(XCH,9,COC)=5 THEN	24327500
MUSTNOTSWAP:	BEGIN MIX:=0; BELOW:=TRUE; GO TO AGAIN;	24328000
	END; %SPACE IS GOT BELOW FENCE IF EOJ	24329000
	IF CANEXPAND[MIX] THEN	24330000
	IF (BELOW:=BELOW+2) GTR 5 THEN	24331000
	BEGIN	24332000
	TERMINATE(MIX&81[CTF]);	24333000
	GO TO INITIATE;	24334000
	END ELSE GO TO SLEAP;	24350000
	IF TAR[P1MIX] NEQ 0 THEN GO MUSTNOTSWAP;	24350500
	EXPAND[MIX]+3;	24351000
	SWAP(FORCESWAP.3);	24352000
	GO TO AGAIN;	24353000
	END;	24354000
	M[GETSPACE+T]+(*P(DUP))&TYPE[3:42:6]&MIX[9:42:6];	24400000
	IF MESS=0 THEN TELLSP0;	24401000
\$ SET OMIT =	NOT(CHECKLINK OR DEBUGGING)	24404999
	IF CHECK THEN CHECKLINKS(MIX,MEM[MIX,LEFTLIT]);	24405000
\$ POP OMIT		24405001
	IF [MEM[MIX,0]],[CF]>T OR MEM[MIX,0],[FF]<T THEN	24406000
	PUNT(4); % INVALID LINK	24407000
	END GETSPACE;	24408000
\$ SET OMIT =	NOT(SHAREDISK)	24599999
PROCEDURE FINDFREEADDRESS(N);	VALUE N; REAL N;	24600000
BEGIN		24601000
	REAL RCW=+0,T=RCW+1,A=T+1,S=A+1,U=S+1,I=U+1;	24602000
	LABEL LOOK;	24603000
	P(@2000,0,0,0,0);	24609000
	IF (FINDFREECTR:=FINDFREECTR+1) GEQ 32 THEN% THIS PROCEDURE IS	24610000
	% CALLED FREQUENTLY SO MAKE IT A SAVE PROCEDURE	24611000
	M>(*P(.,FINDFREEADDRESS))INX 0=2],[2:1]:=1;	24612000
LOOK: T=@2000; A=0;		24613000
	P(WAITIO(IT) INX @4000000000,0,18),DEL); % REPORT FREE ADDRESS	24614000
	IF A#0 THEN % THERE WAS A FREE ADDRESS	24615000
	BEGIN	24616000
	FOR I:=0 STEP 1 UNTIL LQAVAIL=1 DO	24617000
	IF (LQUE[I],[8:40] EQV A)=NOT 0 THEN	24618000
	IF STASUS[LOCATQUE[S+LQUE[I]],[1:7]],[3:5]]#RUNNING THEN	24618100
	BEGIN	24619000
\$ SET OMIT =	DFX OR OMIT	24619999
	U+(LOCATQUE[S]+(*P(DUP))&(NOT 0)[CTF]],[12:6];	24620000
	IOCOUNT[LOCATQUE[S],[3:5]]+*P(DUP)+1;	24620500
	IF UNIT[U],[FF]>1023 THEN	24621000
	BEGIN	24622000
	UNIT[U]:=(*P(DUP))&S[CTF]&S[CTC];	24623000
	STARTIO(U);	24624000
	END ELSE	24625000
	BEGIN	24626000
	LOCATQUE[UNIT[U],[CF]],[FF]:=S;	24627000
	UNIT[U],[CF]:=S;	24628000
	END;	24629000

```

$ POP OMIT
$ SET OMIT = NOT(CFX) OR OMIT
T=IOQUE[S]&6[3:43:5];
RETURNIOSPACE(S);
P1MIX=LOCATQUE[S],[3:5];
IOREQUEST(FINALQUE[S],T,LOCATQUE[S]&18[12:42:6]);
P1MIXI=0;
$ POP OMIT
IF I LSS (LQAVAIL:=LQAVAIL-1) THEN
BEGIN
STREAM(A:=LQAVAIL-I,B:=[LQUE[I]]);
BEGIN SI:=B;SI:=SI+8;DS:=A WDS END;
I:=I-1;
END;
END ELSE LQUE[I],[12:1]+1;
IF LQAVAIL#0 THEN GO LOOK;
END;
FINDINGADDRESS:=0;
IF N THEN KILL([RCW] INX NOT 2);
END; % OF FINDFREEADDRESS
$ POP OMIT
$ SET OMIT = NOT SEPTICTANK
SAVE PROCEDURE DISPOSAL(L,I,R);
VALUE L,I,R; REAL L,I,R;
BEGIN REAL J,K,YECH;
DEFINE
SEEPAGE = ARGH[60]#, % DISK ADDRESS FOR NEXT I/O,
XXX = ARGH[62]#, % INDEX OF BASE OF CURRENT BUFFER (0 OR 64)
STINK = ARGH[124]#, % BASE ADDRESS OF SEPTIC TANK,
YECHH = ARGH[125]#, % INDEX INTO CURRENT BUFFER (0-63),
TUBUFF = ARGH[126]#, % IF ZERO, PUT EVERYTHING INTO TANK,
% IF NOT ZERO, TU/BUFF OF ADAPTER TO STORE,
NTLOC = ARGH[128]#, % STORAGE FOR NT VARIABLES DURING DISKIO,
COMMENT
THE REST OF ARGH IS USED AS FOLLOWS:
ARGH[-1] % DISK ADDRESS FOR FIRST BUFFER,
ARGH[0-59] % FIRST BUFFER,
ARGH[61] % CELL FOR I/O COMPLETES,
ARGH[63] % DISK ADDRESS FOR SECOND BUFFER,
ARGH[64-123] % SECOND BUFFER,
ARGH[127] % FILE ID OF CURRENT SEPTIC TANK,
NOTE THAT SEEPAGE AND STINK ARE WRITTEN OUT AT THE END OF THE
BUFFERS, THE FACT THAT 3:15 IS ZERO MARKS THE END OF THE TANK;
%
REAL SUBROUTINE WRITEDISK;
BEGIN
MOVE(4,P(,NT1),[NTLOC]); % SAVE THE NT-S STOMPED BY IOREQUEST,
DISKIO(ARGH[61],(ARGH INX XXX)-1,61,SEEPAGE);
MOVE(4,[NTLOC],P(,NT1));
XXX:=((P(DUP))+64) AND 64;
IF P(((SEEPAGE:=P(DUP))+2)=STINK) GEQ SEPTICSIZE,DUP) THEN
BEGIN SEEPAGE:=STINK;
ARGH[XXX]:=3; % TSSMCP AND WRAPAROUND BITS,
END;
WRITEDISK:=P;
END;
%
IF SEPTICTANKING THEN
IF TUBUFF=0 OR (R,[9:4]=TUBUFF,[39:4] AND
(TUBUFF,[1:1] OR (R,[14:4]=TUBUFF,[44:4]))) THEN

```

```

24629001
24629099
24629100
24629200
24629350
24629400
24629500
24629501
24630000
24631000
24632000
24633000
24634000
24635000
24636000
24637000
24638000
24639000
24640000
24641000
24641001
25999990
26000000
26001000
26002000
26003000
26004000
26005000
26006000
26007000
26008000
26009000
26010000
26011000
26012000
26013000
26014000
26015000
26016000
26017000
26018000
26019000
26020000
26021000
26022000
26023000
26024000
26025000
26026000
26027000
26028000
26029000
26030000
26031000
26032000
26033000
26034000
26035000
26036000
26036500

```

```

BEGIN IF R=0 THEN J:=4 ELSE                                26037000
  BEGIN J:=IF R,[18:1] THEN I,[9:9] NEQ 0 ELSE I,[24:1]+2; 26038000
    K:=R,[CF] - I,[CF];                                     26039000
    IF R,[25:1] AND J=2 THEN K:=K-1;                       26040000
  END;                                                      26041000
  IF P((YECH:=YECHH)+K+2,DUP) GTR 60 THEN % LEAVES INDEX OF END 26042000
    P(P=YECH+(YECH:=WRITEDISK)); % OF DATA ON STACK, 26043000
  ARGH[YECH:=XXX+YECH]:=R&K[CTC]&J[3:42:6];                26044000
  ARGH[YECH+1]:=X(CLOCK+P(RTR))&L[12:42:6];               26045000
  STREAM(Q:=P(DUP) NEQ 60: K, I:=I,[CF], R:=[ARGH[YECH+2]]); 26046000
  BEGIN S1:=I; DS:=K WDS; Q(DS:=LIT " "); END;            26047000
  IF NOT P THEN P(DEL,WRITEDISK); % 60 WORDS, WRITE IT, 26048000
  P([YECHH],STD);                                         26049000
END;                                                       26050000
END DISPOSAL;                                           26051000
PROCEDURE RUNSEPTIC(BUFF);                               26060000
VALUE BUFF; REAL BUFF;                                  26061000
BEGIN LABEL SPIT,SEP;                                    26062000
  REAL TU,I,T,B,FID;                                     26063000
  INTEGER RC=FID;                                        26064000
  DEFINE SEEPAGE = ARGH[60]#,                             26065000
    XXX = ARGH[62]#,                                     26066000
    STINK = ARGH[124]#,                                  26067000
    YECHH = ARGH[125]#,                                  26068000
    TUBUFF = ARGH[126]#;                                  26069000
%                                                         26070000
  BI:=IF BUFF=0 THEN SPACE(10) ELSE BUFF,[15:15]-1;    26071000
  IF BUFF GEQ 0 THEN % OPEN SEPTIC FILE                 26072000
  BEGIN IF SEPTICTANKING THEN GO SPIT;                   26073000
    IF BUFF NEQ 0 THEN % CHECK FOR SPECIFIC TU BUFF    26074000
    BEGIN STREAM(A:=100, B:=100: X:=0, BUFF);           26075000
      BEGIN S1:=BUFF; D1:=LOC A;                        26076000
        2(                                              26077000
          S1: IF SC=ALPHA THEN                            26078000
            BEGIN IF SC LSS "0" THEN JUMP OUT;          26079000
              TALLY:=1; X:=TALLY;                       26079500
              S1:=S1+1;                                   26080000
              IF SC LEQ "Z" THEN GO TO ONE;             26081000
              IF SC LEQ "9" THEN                        26082000
                BEGIN S1:=S1-1; DS:=2 OCT END ELSE     26083000
                BEGIN S1:=S1-1; DS:=OCT END;           26084000
            END ELSE                                     26085000
            IF SC="*" THEN JUMP OUT ELSE                 26086000
            IF SC = "=" THEN                            26086100
              BEGIN X(DS:=BLIT"0000001N");             26086200
                JUMP OUT;                                26086300
            END ELSE                                     26086400
            BEGIN S1:=S1+1; GO TO S1 END;              26087000
          );                                              26088000
        END;                                             26089000
      T:=P;                                               26090000
      IF ((I=P) OR T) GTR 15 THEN                       26091000
        IF (I=100) AND (T=100) THEN TU:=0 ELSE         26092000
        IF T=100 THEN IF I GTR STAMAX THEN GO SPIT ELSE 26092100
        IF (TU:=LINETABLE[                               26092200
          $ SET OMIT = TWXONLY OR OMIT                   26092299
          IF I GTR LMAX THEN                              26092300
            STABLE[I],LEENKER ELSE                      26092400
          $ POP OMIT                                     26092401
          I],[9:9])=0 THEN                               26092500

```

RUN SEPTIC

```

                                GO SPIT ELSE ELSE                26092600
                                IF (I LSS 16) AND (I GTR 0) AND (T=101) THEN 26092700
                                TU:=(=0)&I[39:44:4] ELSE GO SPIT ELSE 26092800
                                IF (TU:=T&I[39:44:4]) LSS 32 THEN GO SPIT; 26093000
                                END;                                26094000
                                T:=SPACE(30);                       26095000
                                MOVE(30,T=1,T);                     26096000
                                M[T]:=@0007400074000102;           26097000
                                M[T+4]:=@4000000001040;           26098000
                                M[T+5]:=M[T+6]:=@14;               26099000
                                STREAM(DATE,X:=T+3);               26100000
                                BEGIN SI:=LOC DATE; DS:=8 OCT;    26101000
                                DI:=X; DS:=2 LIT "+#";             26102000
                                SII=X; SI:=SI+5; DS:=3 CHR;        26103000
                                END;                                26104000
                                I:=M[T+10]:=GETUSERDISK(M[T+8]:=SEPTICSIZE+1); 26105000
                                M[T+7]:=SEPTICEOF;                 26106000
                                M[T+9]:=1&TWO(3-SYSNO)[5:44:4];    % OPEN FOR OUTPUT 26107000
                                RC:=(NT1:=XCLOCK/3600) MOD 60;    26108000
                                NT1:=NT1 DIV 60;                  26109000
                                STREAM(NT1,RC,F:=FID);             26110000
                                BEGIN SI:=LOC NT1; DS:=LIT"0";    26111000
                                DS:=2 DEC; DS:=2 DEC; DS:=3 LIT" "; 26112000
                                END;                                26113000
                                ARGH:=((GETSPACE(132,5,1)+2) INX M)&132[8:38:10]; 26114000
                                SEEPAGE:=STINK:=I;                 26115000
                                XXX:=ARGH[1]:=0;                   % WORD 1 MARKS END OF TANK AND 0 26116000
                                YECHH:=ARGH[0]:=1;                 % MARKS TSSMCP WITHOUT WRAPAROUND 26117000
                                TUBUFF:=TU;                        26118000
                                ARGH[127]:=FID;                     26119000
                                DISKWAIT(ARGH,[CF],2,1);           26120000
                                ENTERUSERFILE(P(SEP),FID,T=1);    26121000
                                FORGETSPACE(T);                     26122000
                                SEPTICTANKING:=TRUE;               26123000
                                END ELSE % CLOSE SEPTIC FILE     26124000
                                BEGIN IF NOT SEPTICTANKING THEN GO SPIT; 26125000
                                SEPTICTANKING:=FALSE;             26126000
                                IF YECHH NEQ 0 THEN                26127000
                                DISKWAIT(ARGH INX XXX,61,SEEPAGE); 26128000
                                P(DIRECTORYSEARCH(=P(SEP),ARGH[127],12),DEL); 26129000
                                FORGETSPACE(ARGH);                 26130000
                                END;                                26131000
                                STREAM(A:=P(SEP), B:=ARGH[127], C:=TUBUFF,[39:4], D:=TUBUFF,[44:4], 26132000
                                X:=TUBUFF,[1:1],                    26132500
                                E:=BUFF GEQ 0, F:=B);              26133000
                                BEGIN SI:=LOC A; SII:=SI+1; DS:=LIT" "; 26134000
                                DS:=7 CHR; SII:=SI+1; DS:=LIT"/"; DS:=7 CHR; 26135000
                                C(DS:=5 LIT" FOR "; DS:=2 DEC; DS:=LIT"/"; 26136000
                                DS:=2 DEC; JUMP OUT);               26137000
                                X(DI:=DI-2; DS:=2LIT"= ");         26137500
                                DS:=2 LIT" C"; CII:=CI+E; GO TO L1; 26138000
                                DS:=4 LIT"REAT"; GO TO L2;        26139000
                                L1: DS:=3 LIT"LOS";                 26140000
                                L2: DS:=3 LIT"ED+";                 26141000
                                END;                                26142000
                                SPIT;                                26143000
                                SPOUT(B);                            26144000
                                P(XIT);                              26145000
                                SEP:== @62254763312360;           26146000
                                END SEPTIC RUNNER;                 26147000

```

\$ POP OMIT	26150010
\$ SET OMIT = NOT(B6500LOAD)	27990099
BOOLEAN PROCEDURE B6500FORMATTER(CT,I,X,XX,BCL,H,OPTION);	27990100
VALUE OPTION; REAL CT,I,OPTION; ARRAY X[*],XX[*],BCL[*],H[*];	27990200
BEGIN	27990300
REAL NT1,T,HDRTYPE,ROWSZ;	27990400
INTEGER NT2;	27990425
LABEL OK,BADFIX,GOODFIX;	27990450
%	27990490
BOOLEAN SUBROUTINE UNSCREW;	27990500
BEGIN	27990600
NT1:=CT*8;	27990700
STREAM(AD:=(XX INX(NT1 DIV 48))&(NT1 DIV 6)[30:45:3],	27990800
BITS:=((NT1-CT*6)MOD 6)DIV 1,N1:=0,N2:=0,N3:=0,BCL);	27990900
BEGIN	27991000
SI:=AD;SKIP BITS SB;	27991100
4(DI:=DI+6;SKIP 4 DB;8(IF SB THEN DS:=SET ELSE DS:=RESET;SKIP SB));	27991200
SI:=BCL;SI:=SI+24;DI:=LOC N2;DS:=WDS;	27991300
SI:=AD;SKIP BITS SB;SKIP 32 SB;N2(SKIP 8 SB);	27991400
DI:=BCL;DI:=DI+38;SKIP 4 DB;	27991500
8(IF SB THEN DS:=SET ELSE DS:=RESET;SKIP SB);N3:=DI;SI:=N3;	27991600
SI:=SI-8;DI:=LOC N3;DS:=WDS;DI:=BCL;DI:=DI+40;SI:=AD;	27991700
SKIP BITS SB;SKIP 32 SB;	27991800
N2(DI:=DI+7;SKIP 2 SB;6(IF SB THEN DS:=SET ELSE DS:=RESET;SKIP SB));	27991900
SKIP 8 SB;	27992000
N3(DI:=DI+7;SKIP 2 SB;6(IF SB THEN DS:=SET ELSE DS:=RESET;SKIP SB));	27992100
END OF EBCDIC FORMATTING;	27992200
NT2:=BCL[3]+BCL[4]-1;	27992300
FOR NT1:=0 STEP 1 UNTIL NT2 DO BCL[NT1+5]:=P(DUP,LOD)+@20;	27992400
CT:=CT+BCL[0];	27992500
STREAM(N1:=IF BCL[3] >7 THEN 7 ELSE BCL[3],	27992600
N2:=IF BCL[4] >7 THEN 7 ELSE BCL[4],	27992650
N4:=IF BCL[3] >7 THEN BCL[3]-7 ELSE 0,	27992700
N3:=BCL INX 5, N5:=[X[I]]);	27992750
BEGIN	27992800
DS+16 LIT "0";	27992900
DI:=DI-15; SI:=N3;	27993000
N1(SI:=SI+7;DS:=CHR); DI:=N5; DI:=DI+9;	27993100
N4(SI:=SI+8);	27993150
N2(SI:=SI+7;DS:=CHR);	27993200
END;	27993300
IF BCL[2]=1 THEN BEGIN X[I+1]:=X[I];X[I]:=0 END;	27993400
UNSCREW:=BCL[0]=0 OR (I:=I+2)>1024;	27993500
END OF TAPE DIRECTORY CONVERSION;	27993600
%	27993690
REAL SUBROUTINE FIXHDR;	27993700
BEGIN	27993800
BCL:=[M[CT:=SPACE(36)]]&36[8:38:10];	27993900
MOVE(30,CT-1,CT);	27994000
STREAM(SI:=0; T:=H[5]);	27994010
BEGIN SI:=LOC T;DI:=LOC S;DI:=DI+7;SKIP 2 DB; 4(IF SB THEN	27994020
DS:=SET ELSE DS:=RESET; SKIP SB); END;%HDR FMT 10 IN [0:4],	27994030
HDRTYPE:=P;	27994040
NT2*(IF HDRTYPE=0 THEN H[5],[18:10] ELSE H[5],[10:14])+14;	27994050
IF NT2 GTR 34 THEN	27994100
BEGIN IF NT2 GTR 900 THEN GO TO BADFIX;	27994110
FOR NT1:=36 STEP 1 UNTIL NT2+1 DO	27994120
IF XX[NT1] NEQ 0 THEN NT1:=1023;	27994130
IF NT1=1023 THEN GO TO BADFIX ELSE	27994140
BEGIN NT2+34;	27994150

```

        IF HDRTYPE=0 THEN H[5],[23:5]+20 ELSE                27994160
                H[5],[19:5]+20;                                27994170
    END;                                                       27994180
END;                                                           27994190
FOR NT1:=15 STEP 1 UNTIL NT2 DO BCL[NT1-5]:=H[NT1]; % PASS ROWS 27994200
STREAM(R+0,B+0;W+H[3]);                                       27994300
BEGIN                                                         27994400
    % B = BLOCK UNITS
    % R = MAX RECORD UNITS
    SI:=LOC W;                                                 27994500
    DI:=LOC B; DI:=DI+5; SKIP 2 DB;                             27994600
    16(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP 1 SB);          27994700
    SKIP 16 SB;                                                27994800
    DI:=LOC R; DI:=DI+5; SKIP 2 DB;                             27994900
    16(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP 1 SB);          27995000
END;                                                           27995100
NT2:=P; % UNITS PER BLOCK                                     27995200
NT1:=P; % UNITS PER RECORD                                    27995300
NT3:=IF NT1 = 0 THEN 1 ELSE NT2 DIV NT1; % RECORDS PER BLOCK 27995400
IF H[2],[8:1] THEN % UNITS = CHARACTERS 27995500
IF (NT4:=H[2],[13:3] = 2) THEN % DECIMAL (4-BIT) 27995600
BEGIN NT1:=NT1 DIV 12; NT2:=NT2 DIV 12; END ELSE 27995700
IF NT4 = 4 THEN % EBCDIC 27995800
BEGIN NT1:=NT1 DIV 6; NT2:=NT2 DIV 6; END ELSE 27995900
BEGIN NT1:=NT1 DIV 8; NT2:=NT2 DIV 8; END; % BCL 27996000
IF NT1 GTR 1023 THEN GO TO BADFIX; % WORDS PER RECORD 27996100
IF NT2 GTR 1023 THEN GO TO BADFIX; % WORDS PER BLOCK 27996200
BCL[0]:=(NT2 DIV 30 + (NT2 MOD 30 NEQ 0))& % SEGMENTS/BLOCK 27996300
                NT3[30:36:12]& % RECORDS/BLOCK 27996400
                NT2[15:33:15]& % WORDS/BLOCK 27996500
                NT1[1:34:14]; % WORDS/RECORD 27996600
STREAM(A:=H[6],[10:18]; H:=H[6]; X:=0; DATE); 27996625
BEGIN DI:=LOC X; SI:=LOC DATE; DS:=8 OCT; 27996650
    DI:=LOC A; SI:=LOC H; SKIP 2 DB; 27996675
    10(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB); 27996700
    SI:=SI+11; DS:=3 CHR; 27996725
END; 27996750
P([BCL[3]],STD); 27996775
BCL[7]:=H[4]; 27996800
BCL[8]:=IF HDRTYPE=0 THEN H[5],[28:20] ELSE ROWSZ:=H[5],[24:24]; 27996825
BCL[9]:=IF HDRTYPE=0 THEN H[5],[23:05] ELSE H[5],[19:05]; 27996850
IF HDRTYPE#0 THEN % CALCULATE END FOR NEWLIB OPTION 27996880
BEGIN NT1:=H[14],[20:28]+(P(H[14],TOP)=0 OR P(XCH).[1:19]#0); 27996900
    IF (NT2:=NT1 MOD ROWSZ)=0 THEN NT2:=ROWSZ; 27996920
    T:=(NT1+ROWSZ-1) DIV ROWSZ; 27996940
    IF T GTR BCL[9] THEN 27996960
    BEGIN T:=BCL[9]; 27996980
        NT2:=ROWSZ; 27997000
    END; 27997020
    IF BCL[T+9]=0 THEN 27997040
    BEGIN NT2:=BCL[8]; 27997060
        WHILE T GTR 1 DO 27997080
        IF BCL[(T:=T-1)+9]#0 THEN GO TO OK; 27997100
    END; 27997120
OK; T,[9:24]:=NT2; 27997140
END; % T=0 FOR OLD HEADERS 27997160
MOVE(30,BCL,H); 27997180
GO TO GOODFIX; 27997200
BADFIX; 27997220
T:=1; 27997240
GOODFIX; 27997260
FORGETSPACE(CT); 27997280

```

```

FIXHDR:=T;
END OF FIXHDR;
B6500FORMATTER:=IF OPTION = 0 THEN UNSCREW ELSE FIXHDR;
END OF B6500FORMATTER;
$ POP OMIT
PROCEDURE LIBRARYLOADSPECIALCASE(Z); VALUE Z; REAL Z;
BEGIN COMMENT LIBRARYLOAD HAS BEEN BROKEN-UP TO PREVENT SIZE OVERFLOW
AND THE TIE UP OF CORE BY CODE NOT OFTEN USED, THIS
PROCEDURE DOES INITIAL SET-UP AND OTHER SPECIAL FUNCTIONS
FOR LIBRARYLOAD, IT REFERENCES THE LIBRARYLOAD LOCALS
BY F=RELATIVE DECLARATIONS AND CHANGES TO LOCAL
DECLARATIONS IN LIBRARYLOAD SHOULD BE MADE WITH THE
CORRESPONDING CHANGES HERE TO LINE UP THE STACK
CORRECTLY, ADDITIONS SHOULD BE MADE BEFORE DECLARATIONS
OF LOCAL VARIABLES FOR B6500LOAD;
REAL COMMON=-4, MSCW=-2, RCW=+0;
REAL ALPHA=+1, EADD=ALPHA+1,
FID=EADD+1, FN=FID+1,
I=FN+1, IC=I+1,
J=IC+1, K=J+1,
LAST=K+1, LOADING=LAST+1,
MID=LOADING+1, N=MID+1,
N1=N+1, N2=N1+1,
Q=N2+1, REEL=Q+1,
SEG=REEL+1, SIZE=SEG+1,
SN=SIZE+1, T=SN+1,
TYPE=T+1, U=TYPE+1,
UNITNO=U+1, W=UNITNO+1,
Y=W+1;
BOOLEAN BB=Y+1, B6500=BB+1,
LATEST=B6500+1, TOGS=LATEST+1;
ARRAY AROW=TOGS+1[*], H=AROW+1[*],
IOD=H+1[*], LAB=IOD+1[*],
LBL=LAB+1[*], S=LBL+1[*],
X=S+1[*];
$ SET OMIT = NOT(B6500LOAD)
REAL CCT=X+1, LASTROW=CCT+1,
NT1=LASTROW+1, NT2=NT1+1;
BOOLEAN REELSW=NT2+1;
ARRAY BCL=REELSW+1[*], XX=BCL+1[*];
$ POP OMIT
LABEL TRYNEXT, BAC, FINDIT, TRYAGN, BACK, LOADEM, FINDFILENAMES, EXIT;
LABEL XXIT;
LABEL CASE0, CASE1, CASE2;
SWITCH SWIT:= CASE0, CASE1, CASE2;
DEFINE SKIPDIR=TOGS, [47:1]#, REEL1START=TOGS, [46:1]#,
SPACITSW=TOGS, [45:1]#, CHKLBL=TOGS, [44:1]#,
DSED=(TERMSET(P1MIX))#, SPACER=5&@1400[CTF]#,
SPOUTUNIT=0#,
MM=@37700040#, SM=@37700000#;
*****
SUBROUTINE GETASEGMENT;
BEGIN
SEG:=S[29];
DISKWAIT('S, [CF], 30, SEG);
FORGETESPDISK(SEG);
I:=2;
END; % OF GETASEGMENT
*****
SUBROUTINE ABORT;

```

```

27997300
27997350
27997400
27997500
27997501
27997600
27997605
27997610
27997615
27997620
27997625
27997630
27997635
27997640
27997645
27997650
27997656
27997658
27997660
27997662
27997664
27997666
27997668
27997670
27997672
27997674
27997676
27997678
27997680
27997682
27997684
27997686
27997688
27997690
27997692
27997694
27997696
27997698
27997700
27997702
27997704
27997710
27997712
27997715
27997720
27997725
27997730
27997735
27997737
27997740
27997745
27997750
27997755
27997760
27997765
27997770
27997775
27997780
27997785
27997790

```



```

BEGIN
IF LOADING THEN P(DIRECTORYSEARCH(X[J],X[J+1],5+LOADING),DEL);
IF U≥0 THEN
BEGIN
STOPTIMING(5,1023);
STOPTIMING(0,1023);
BLASTQ(U);
SETNOTINUSE(U,0);
END;
WHILE S[29]≠0 DO GETASEGMENT;
$ SET OMIT = PACKETS
IF UNITNO≠0 AND LABELTABLE[UNITNO]=@214 THEN
LABELTABLE [UNITNO]←@114;
$ POP OMIT
STREAM(T:=T:=SPACE(5));
BEGIN DS←21LIT"LIBRARY LOAD ABORTED←"; END;
SPOUT (T);
GO INITIATE;
END; % ABORT
*****
BOOLEAN SUBROUTINE LABELCHECK;
BEGIN
TRYNEXT;
P(WAITIO(LAB INX @120540000000,0,U),DEL);
$ SET OMIT = NOT(B6500LOAD)
IF B6500 THEN
BEGIN IF REELSW AND(NFLAG(LAB[1]),[1:17] EQV "000")≠NOT 0
THEN BEGIN DO P(WAITIO(LAB INX @120540000000,@40,U),DEL)
UNTIL (NFLAG(LAB[1]),[1:17] EQV "001")≠NOT 0;
REELSW←FALSE;
END; P(WAITIO(SPACER,MM,U),DEL);
END;
$ POP OMIT
IF @40≠WAITIO(SPACER,@40,U) THEN
P(WAITIO(@4740000005,0,U),DEL);
IF DSED THEN ABORT;
IF (NOT B6500 AND ((NFLAG(LAB[0]),[6:42] EQV "LABEL ")≠NOT 0
OR (NFLAG(LAB[2]),[6:24] EQV "FILE")≠NOT 0))
$ SET OMIT = NOT(B6500LOAD)
AND (((NT1:=NFLAG(LAB[0]),[1:23] EQV "HDR1")≠NOT 0 OR
(NT1,[24:24] EQV "FILE")≠NOT 0)
$ POP OMIT
THEN BEGIN
STREAM(A:=TINU[U], T:=T:=SPACE(10));
BEGIN SI←A;SI←SI+5;DS←LIT",";DS←3 CHR;
DS←21 LIT" NOT A LIBRARY TAPE←";
END;
SPOUT(T); T←1;
END ELSE T←0;
IF T≠0 AND NOT B6500 THEN
IF NFLAG(LAB[2]),[30:18]=0 AND SKIPDIR THEN
BEGIN
SPACITSW←1; CHKLBL←FALSE;
GO TO BACK; %BRANCH INTO SPACIT,
BAC:
SPACITSW←0; CHKLBL←TRUE;
GO TO TRYNEXT;
END;
LABELCHECK←T;
END;

```

```

27997795
27997800
27997805
27997810
27997815
27997820
27997825
27997830
27997835
27997840
27997845
27997850
27997855
27997856
27997860
27997865
27997875
27997890
27997895
27998050
27998055
27998060
27998065
27998070
27998075
27998080
27998085
27998090
27998095
27998100
27998105
27998110
27998111
27998115
27998120
27998125
27998130
27998135
27998140
27998145
27998150
27998151
27998155
27998160
27998165
27998170
27998175
27998185
27998200
27998205
27998210
27998215
27998220
27998225
27998230
27998235
27998240
27998245
27998250
27998255

```

```

*****
SUBROUTINE FINDTHETAPE;
BEGIN
FINDIT:
  IF (U*FINDINPUT(MID,FID,REEL,=0,0,0,0,1,5)) < 0 THEN ABORT;
  REEL*RDCTABLE[U],[14:10]; %FORCE REEL CONTINUITY IF IL=ED,
  RRRMECH:=TWO(U) OR RRRMECH;
$ SET OMIT = NOT(B6500LOAD)
  IF B6500 THEN P(WAITIO(SPACER,MM,U),DEL);
$ POP OMIT
  IF CHKLBL THEN IF LABELCHECK THEN
  BEGIN
    SETNOTINUSE(U,1);
    GO FINDIT;
  END;
  STARTIMING(5,U);
  M[PRT[P1MIX,3] INX (5*REEL+3)],[23:1] := 1;
  RDCTABLE[U],[8:6]:=P1MIX;
  STREAM (S*PRNTABLE[U],[18:30],T+[T]);
  BEGIN SI*LOC S; DS*8DEC; DI*DI-7; DS*6FILL; END;
  FILEMESSAGE(" IN "&TINU[U][6:30:18],T,
    MID,FID,REEL,0,0,OPNMESS OR OPENK);
END; % OF FINDTHETAPE
*****
BOOLEAN SUBROUTINE ENDOFREEL;
BEGIN
  BLASTQ(U);
  IF P(WAITIO(LAB INX @120540000000,@2000040,U),DUP)=@20 THEN
  BEGIN % PAR ON ENDING LABEL:TEST FOR LAST FILE ON TAPE(EOF)
    LAB[4]:=(P(DUP))&(WAITIO(SPACER,@40,U)=@40)[47:47:11];
    P(WAITIO(5&@3400[CTF],@377,U),DEL);
  END;
$ SET OMIT = NOT(B6500LOAD)
  IF B6500 THEN
  BEGIN IF (NFLAG(LAB[0]),[1:23] EQV "EOV1")#NOT 0 THEN
    BEGIN P(WAITIO(SPACER,MM,U),DEL);
      NT1*WAITIO(LAB INX @120540000000,@40,U);
    END; P(DEL);
  END ELSE
$ POP OMIT
  NT1:=P;
  IF DSED THEN ABORT;
  IF ((NOT B6500) AND NFLAG(LAB[4]) AND NT1#@40)
$ SET OMIT = NOT(B6500LOAD)
  OR(( NFLAG(LAB[0]),[1:23] EQV "EOV1")=NOT 0)
$ POP OMIT
  THEN BEGIN
    STOPTIMING(5,1023);%
    SETNOTINUSE(U,0);
    REEL*REEL+1;
$ SET OMIT = NOT(B6500LOAD)
    IF B6500 THEN BEGIN REELSW*TRUE;
      STREAM(S*LAB INX 0,D+[FID]);
      BEGIN SI*S; SI*SI+4;DI*DI+1; DS*7 CHR;END;
    END ELSE
$ POP OMIT
    FID:=LAB[2];
    FINDTHETAPE;
    ENDOFREEL*TRUE;
  END ELSE ENDOFREEL*FALSE;

```

```

27998260
27998265
27998270
27998275
27998280
27998285
27998290
27998295
27998300
27998301
27998305
27998310
27998315
27998320
27998325
27998330
27998335
27998340
27998345
27998350
27998365
27998370
27998390
27998395
27998400
27998405
27998410
27998415
27998420
27998425
27998430
27998435
27998440
27998445
27998450
27998455
27998460
27998465
27998467
27998468
27998470
27998475
27998480
27998485
27998490
27998491
27998495
27998500
27998505
27998510
27998515
27998520
27998525
27998530
27998535
27998536
27998540
27998545
27998550
27998555

```

```

END;          % OF SUBROUTINE ENDOFREEL                                27998560
*****
SUBROUTINE SPACIT;%                                                27998570
BEGIN                                                    27998575
BACK: WHILE WAITIO(SPACER,MM,U)#@40 DO                               27998580
    BEGIN                                                27998585
        IF DSED THEN ABORT;                                    27998590
        IF(T:=PRTROW[P1MIX],[PSF]) NEQ 0 THEN%CHKFORSWAP        27998595
        BEGIN IF T=2 THEN BEGIN IOCOUNT[P1MIX]:=1; STOPM END  27998601
            ELSE IF T NEQ 1 THEN SWAP(FORCESWAP,1);            27998602
        END;                                                27998603
        IF ELAPSEDLIMIT[P1MIX] GTR IOTIME[P1MIX] THEN          27998604
        IF PROCLIMIT[P1MIX] GTR PROCTIME[P1MIX]+CLOCK+P(RTR) THEN 27998605
            GO XXIT;                                           27998606
        FOR T:=SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO             27998607
        IF ACTIVE[T] GTR 1 THEN                                27998608
        BEGIN IOCOUNT[P1MIX]:=1; SWAP(TIMEND,1);              27998609
            ELAPSEDLIMIT[P1MIX]:=P(DUP)+128;                  27998610
            PROCLIMIT[P1MIX]:=P(DUP)+64;                       27998611
            GO XXIT;                                           27998612
        END;                                                27998613
        XXIT;                                                27998614
    END;                                                    27998615
    IF ENDOFREEL AND NOT SPACITSW THEN GO BACK;               27998616
    IF SPACITSW THEN GO TO BAC; %BRANCH TO LABELCHECK ELSE EXIT 27998617
END;                                                    27998618
*****
BOOLEAN SUBROUTINE NOTLOADINGFROMREEL1;                        27998619
BEGIN %SKIP LAST PORTION OF FILE FROM PREVIOUS REEL           27998620
    SPACIT;                                                  27998625
    IF LABELCHECK THEN P(0) ELSE                               27998630
        IF (NFLAG(LAB[2]) EQV "FILE000") = NOT 0 THEN        27998635
        BEGIN REEL1START+FALSE; P(1) END ELSE P(0);           27998640
    NOTLOADINGFROMREEL1:=P;                                   27998645
END;                                                    27998650
*****
P(Z,RCW,MSCW,STF); RCW:=RCW&P(XCH)[CTC];                     27998655
GO TO SWIT[P];                                               27998660
CASE 0:                                                    27998665
    SI=[M[SPACE(30)]]&30[8:38:10];                             27998730
    TYPE:=COMMON,[FF];                                         27998740
    S[29]:=COMMON,[CF];                                        27998750
$ SET OMIT = NOT(B6500LOAD)                                    27998760
    B6500:=COMMON,[15:1];                                       27998770
$ POP OMIT                                                    27998780
    LATEST:=COMMON,[4:1];                                       27998790
    COMMON + IF COMMON,[9:6]#0 THEN =COMMON,[9:6] OR M ELSE  27998809
        IF COMMON,[3:1] THEN 1 OR M ELSE                       27998810
        IF COMMON,[2:1] THEN 2 OR M ELSE 0;                    27998811
    REEL:=1;                                                  27998815
    GETASEGMENT;                                              27998820
    STREAM(MID:=MID:=S[1],B:=PRT[P1MIX,3]);                   27998830
    BEGIN DS:=16 LIT"ODIRECTRYODISK " ;DS:=25 LIT"0";SII=LOC MID; 27998840
        SII=SI+1;DS:=7 CHR;DS:=8 LIT"0FILE000";DS:=24 LIT"0"; 27998850
    END;                                                    27998860
    UNITNO:=S[0],[2:6];                                       27998870
    LAB:=[M[SPACE(15)]]&15[8:38:10];                           27998880
    MID:=S[1];                                                27998890
    FID:="FILE000";                                           27998900
    REEL1START+TRUE; CHKLBL+TRUE;                              27998910

```

TRYAGN: FINDTHETAPE;	27998960
\$ SET OMIT = NOT(B6500LOAD)	27998969
IF NOT B6500 THEN	27998980
\$ POP OMIT	27998981
IF FID#LAB[2] OR REEL#1 THEN	27998990
IF NOT NOTLOADINGFROMREEL1 THEN	27999000
BEGIN STREAM(A+[TINU[U]],T+T+SPACE(10));	27999010
BEGIN SI+A;SI+SI+5;DS=LIT",";DS+3CHR;	27999020
DS+20 LIT" NOT A LIBRARY TAPE";	27999030
DS=LIT"+";	27999040
END;	27999050
SPOUT(T); SETNOTINUSE(U,1);	27999060
REEL+1;	27999070
GO TO TRYAGN;	27999080
END;	27999090
MID=LAB[1];	27999100
SKIPDIR:=TRUE;	27999110
X=[M[T:=SPACE(1024)]]&1023[8:38:10];	27999120
IF NOT B6500 THEN MID=LAB[1];	27999130
STARTIMING(0,18);	27999140
P(WAIT10((	27999150
\$ SET OMIT = NOT(B6500LOAD)	27999169
IF B6500 THEN (XX:=X&(GETSPACE(1024,0,1)+2)[CTC]) ELSE	27999170
\$ POP OMIT	27999171
X)&@5400[CTF],0,U),DEL);	27999180
\$ SET OMIT = NOT(B6500LOAD)	27999199
IF B6500 THEN	27999200
BEGIN	27999210
BCL:=XX&(GETSPACE(327,0,1)+2)[CTC];	27999220
CCT:=12;I:=0;MOVE(327,BCL,[CF]=1,BCL);	27999230
DO UNTIL B6500FORMATTER(CCT,I,X,XX,BCL,H,0);	27999240
FORGETSPACE(XX);FORGETSPACE(BCL);	27999250
M[T-1]:=I;X[I-2]:=@14;	27999260
END;	27999270
\$ POP OMIT	27999271
IF DSED THEN ABORT;	27999280
IF (N:=M[T-1]) LSS 900 THEN	27999290
BEGIN % GET RID OF EXTRA MEMORY SPACE IF NOT NEEDED	27999300
X=[M[SPACE(N)]]&N[8:38:10];	27999310
MOVE(N,T,X);	27999320
FORGETSPACE(T);	27999330
END;	27999340
FINDFILENAMES:	27999350
FOR I:=2 STEP 2 UNTIL 26 DO	27999360
BEGIN	27999370
J:=IF X[0]=@114 AND NOT REEL1START THEN X[1]=2 ELSE -2;	27999380
IF (FN+S[I])=@14 THEN GO LOADEM;	27999390
SN=S[I+1];W+1;	27999400
WHILE X[J+J+2]#@14 DO % MARK FILES TO BE LOADED	27999410
IF FN<0 OR (FN EQV X[J])=NOT 0 THEN	27999420
IF SN<0 OR (SN EQV X[J+1])=NOT 0 THEN W+X[J]+X[J];	27999430
IF W GTR 0 THEN LBMESS(FN,SN,-1,17,TINU[U],SPOUTUNIT,1);	27999440
END;	27999470
IF S[28]=@14 THEN GO LOADEM;	27999480
GETASEGMENT;	27999490
GO FINDFILENAMES;	27999500
LOADEM:	27999510
W+J+2;	27999520
WHILE X[J+J+2]#@14 DO IF X[J],[1:1] THEN W+J;	27999530
IF W<0 THEN ABORT;	27999540

```

X[W+2]=@14;
IF TYPE=ADDV THEN
FOR W=W STEP =2 UNTIL 0 DO
IF X[W],[1:1] THEN
IF DIRECTORYSEARCH(X[W],X[W+1],5)≠0 THEN X[W]=@14 ELSE
W:=0 ELSE X[W]=0;
CHKLBL=FALSE;
J=0;
IF @40=WAITIO(LAB INX @120540000000,@40,U) THEN
IF B6500 THEN P(WAITIO(LAB INX @120540000000,0,U),DEL) ELSE
J=ENDOFREEL;
IF NOT J THEN% CHECK ENDING LABEL IF NOT LAST FILE OR B6500LOAD
IF ((NOT B6500) AND (NFLAG(LAB[1])EQV MID)≠NOT 0 OR
(NFLAG(LAB[2]) EQV "FILE000")≠NOT 0)
$ SET OMIT = NOT(B6500LOAD)
AND ((NFLAG(LAB[0]),[24:24] EQV "FILE")≠NOT 0 AND
(NFLAG(LAB[1]),[1:17] EQV FID,[30:18])≠NOT 0)
$ POP OMIT
THEN BEGIN STREAM(A:=[TINU[U]],J:=J:=SPACE(10));
BEGIN SI ← A; SI ← SI+5; DS ← LIT", "; DS ← 3 CHR;
DS ← 29 LIT " BAD FILE000 ON LIBRARY TAPE";
END; SPOUT (J); ABORT;
END;
CHKLBL=TRUE;
J=IF X[0]=@114 AND NOT REEL1START THEN X[1] ELSE 0;
H=[M[SPACE(31+6×B6500)]]&36[8:38:10];
AROW=[M[SPACE(2)]]&2[8:38:10];
AROW[0]=[M[SPACE(902)]]&901[8:38:10];
AROW[1]=AROW[0]&SPACE(902)[CTC];
IOD=[M[SPACE(2)]]&2[8:38:10];
$ SET OMIT = NOT(B6500LOAD)
IF B6500 THEN BEGIN P(WAITIO(SPACER,MM,U),DEL);
P(WAITIO(SPACER,MM,U),DEL) END;
$ POP OMIT
GO TO EXIT;
CASE1: FINDTHETAPE; GO TO EXIT;
CASE2: ABORT;
EXIT: P(0,RDS,0,XCH,P&P[CTF],STF);
END OF LIBRARYLOADSPECIALCASE;
PROCEDURE LIBRARYLOAD;
BEGIN REAL COMMON=4;
REAL ALPHA, EADD, FID, FN,
I, IC, J, K,
LAST, LOADING, MID, N,
N1, N2, Q, REEL,
SEG, SIZE, SN, T,
TYPE, U, UNITNO, W,
Y;
BOOLEAN BB, B6500, LATEST,
TOGS;
ARRAY AROW[*], H[*], IOD[*],
LAB[*], LBL[*], S[*],
X[*];
$ SET OMIT = NOT(B6500LOAD)
REAL CCT, LASTROW, NT1, NT2;
BOOLEAN REELSW;
ARRAY BCL[*], XX[*];
$ POP OMIT
DEFINE DSED=(TERMSET(P1MIX))#,
SPOUTUNIT=0#,

```

```

27999550
27999560
27999570
27999580
27999590
27999600
27999610
27999620
27999630
27999640
27999650
27999660
27999670
27999680
27999699
27999700
27999710
27999711
27999720
27999730
27999740
27999760
27999790
27999800
27999810
27999820
27999830
27999840
27999850
27999860
27999879
27999880
27999890
27999891
27999900
27999910
27999920
27999930
27999940
28000000
28001100
28001200
28001300
28001400
28001500
28001600
28001700
28001800
28001900
28002000
28002100
28002200
28002300
28002400
28002500
28002600
28002700
28002800
28006000
28006100

```

SPACER=5&@1400[CTF]#;	28007000
MM=@37700040#;	28008000
SM=@37700000#;	28009000
LABEL EXIT,TRYNEXT,BAC,PARERR,EXT,LOOP,WATE,BACK,	28010000
BADHEADER,OK,WY,BADLOAD,LAY,SKIPPER,FALLOUT,ENDLOOP;	28010100
DEFINE SKIPDIR=TOGS,[47:1]#,REEL1START=TOGS,[46:1]#;	28011050
SPACITSW=TOGS,[45:1]#,CHKLBL=TOGS,[44:1]#;	28011060
DEFINE INITIALSETUP = LIBRARYLOADSPECIALCASE(0)#;	28011070
FINDTHETAPE = LIBRARYLOADSPECIALCASE(1)#;	28011080
ABORT = LIBRARYLOADSPECIALCASE(2)#;	28011090
*****	28012000
DEFINE NOTLOADED(NOTLOADED1) =	28013000
BEGIN NT3:=NOTLOADED1; NOLOADMESS; END#;	28014000
SUBROUTINE NOLOADMESS;	28015000
LBMESS(ABS(X[J]),X[J+1],-1,NT3,TINU[U],SPOUTUNIT,1);	28016000
*****	28020000
SUBROUTINE CHECKFORSWAP;	28021000
BEGIN	28022000
IF (T:=PRTROW[P1MIX],[PSF])#0 THEN	28023000
BEGIN	28024000
IF T=2 THEN	28025000
BEGIN	28026000
IOCOUNT[P1MIX]:=-1;	28027000
STOPM;	28028000
END ELSE	28029000
IF T#1 THEN SWAP(FORCESWAP,1);	28030000
END;	28031000
IF ELAPSEDLIMIT[P1MIX] GTR IOTIME[P1MIX] THEN	28032000
IF PROCLIMIT[P1MIX]>PROCTIME[P1MIX]+CLOCK+P(RTR) THEN GO EXIT;	28033000
FOR T:=SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO	28034000
IF ACTIVE[T]>1 THEN	28035000
BEGIN	28036000
IOCOUNT[P1MIX]:=-1;	28037000
SWAP(TIMEND,1);	28038000
ELAPSEDLIMIT[P1MIX]:=*P(DUP)+128;	28039000
PROCLIMIT[P1MIX]:=*P(DUP)+64;	28040000
GO TO EXIT;	28041000
END;	28042000
EXIT;	28043000
END; % OF CHECKFORSWAP	28044000
*****	28061000
BOOLEAN SUBROUTINE LABELCHECK;	28062000
BEGIN	28063000
TRYNEXT;	28063100
IF WAITIO(LAB INX @120540000000,@40&@20[CTF],U)=@40 AND	28063300
NOT B6500 THEN % MISSING LABEL " FAKE IT,	28063400
BEGIN STREAM(A:=1, B:=[LAB[2]]);	28063500
BEGIN SI:=LOC A; SI:=SI+5;	28063600
DI:=DI+5; DS:=3 ADD;	28063700
END;	28063800
P(WAITIO(@340000005,@377,U),DEL);	28063900
END;	28064000
% SET OMIT = NOT(B6500LOAD)	28064099
IF B6500 THEN	28064100
BEGIN IF REELSW AND(NFLAG(LAB[1]),[1:17] EQV "000")=NOT 0	28064200
THEN BEGIN DO P(WAITIO(LAB INX @120540000000,@40,U),DEL)	28064300
UNTIL (NFLAG(LAB[1]),[1:17] EQV "001")=NOT 0;	28064400
REELSW=FALSE;	28064500
END; P(WAITIO(SPACER,MM,U),DEL);	28064600
END;	28064700

\$ POP OMIT	28064701
IF @40#WAITIO(SPACER,@40,U) THEN	28065000
P(WAITIO(@4740000005,0,U),DEL);	28065610
IF DSED THEN ABORT;	28066000
IF (NOT B6500 AND ((NFLAG(LAB[0]),[6:42] EQV "LABEL ")#NOT 0	28067000
OR (NFLAG(LAB[2]),[6:24] EQV "FILE")#NOT 0))	28067100
\$ SET OMIT = NOT(B6500LOAD)	28068999
AND (((NT1:=NFLAG(LAB[0]),[1:23] EQV "HDR1")#NOT 0 OR	28069000
(NT1,[24:24] EQV "FILE")#NOT 0)	28069100
\$ POP OMIT	28069101
THEN BEGIN	28070000
STREAM(A1=[TINU[U]], T:=T:=SPACE(10));	28071000
BEGIN SI+A;SI+SI+5;DS+LIT",";DS+3 CHR;	28072000
DS+21 LIT" NOT A LIBRARY TAPE*";	28073000
END;	28074000
SPOUT(T); T+1;	28075000
END ELSE T+0;	28076000
IF T=0 AND NOT B6500 THEN	28076100
IF NFLAG(LAB[2]),[30:18]=0 AND SKIPDIR THEN	28076110
BEGIN	28076200
SPACITSW+1; CHKLBL+FALSE;	28076300
GO TO BACK; %BRANCH INTO SPACIT.	28076400
BAC:	28076500
SPACITSW+0; CHKLBL+TRUE;	28076600
GO TO TRYNEXT;	28076750
END;	28076800
LABELCHECK+T;	28077000
END;	28078000
*****	28079000
BOOLEAN SUBROUTINE ENDOFREEL;	28095000
BEGIN	28096000
BLASTQ(U);	28097000
IF P(WAITIO(LAB INX @120540000000,@2000040,U),DUP)=@20 THEN	28098000
BEGIN % PAR ON ENDING LABEL;TEST FOR LAST FILE ON TAPE(EOF)	28098010
LAB[4]:=(P(DUP))&(WAITIO(SPACER,@40,U)=@40)[47:47:1];	28098020
P(WAITIO(5&@3400[CTF],@377,U),DEL);	28098030
END;	28098040
\$ SET OMIT = NOT(B6500LOAD)	28098099
IF B6500 THEN	28098100
IF (NFLAG(LAB[0]),[1:23] EQV "EOV1")#NOT 0 THEN	28098200
BEGIN P(WAITIO(SPACER,MM,U),DEL);	28098300
NT1+WAITIO(LAB INX @120540000000,@40,U);	28098400
END;	28098500
\$ POP OMIT	28098501
IF B6500 THEN P(DEL) ELSE NT1:=P;	28098600
IF DSED THEN ABORT;	28099000
IF ((NOT B6500) AND NFLAG(LAB[4]) AND NT1#@40)	28101000
\$ SET OMIT = NOT(B6500LOAD)	28101099
OR(( NFLAG(LAB[0]),[1:23] EQV "EOV1")=NOT 0)	28101100
\$ POP OMIT	28101101
THEN BEGIN	28102000
STOPTIMING(5,1023);%	28103000
SETNOTINUSE(U,0);	28104000
REEL+REEL+1;	28105000
\$ SET OMIT = NOT(B6500LOAD)	28105949
IF B6500 THEN BEGIN REELSW+TRUE;	28105950
STREAM(S+LAB INX 0,D+[FID]);	28105960
BEGIN SI+S; SI+SI+4;DI+DI+1; DS+7 CHR;END;	28105970
END ELSE	28105980
\$ POP OMIT	28105981

```

        FID:=LAB[2];
        FINDTHETAPE;
        ENDOFREEL+TRUE;
    END ELSE ENDOFREEL+FALSE;
END; % OF SUBROUTINE ENDOFREEL
*****
BOOLEAN SUBROUTINE CHECK;
BEGIN
    IF (Y:=IOD[W]).[27:2]#0 THEN % PARITY ERROR OR EOF
    BEGIN
        IF Y.[7:1] AND Y.[27:1] THEN % END OF REEL
        BEGIN
            IF NOT ENDOFREEL THEN
            BEGIN
                P(WAITIO(@4740000020,@377,U),DEL);
                NOTLOADED(33);
                Y + 1;
                GO TO EXT;
            END;
            IF WAITIO(IOD[W] INX @1604054000000,SM,U)#0 THEN
                GO PARERR;
            IF IOD[1=W].[7:1] THEN % ANOTHER TAPE IO IN PROGRESS
            BEGIN
                IF WAITIO(IOD[1=W],SM,U)#0 THEN GO PARERR;
                IOD[1=W]:=(*P(DUP)) OR IOMASK;
            END;
            Y+0;
        END ELSE % PARITY ERROR
    PARERR: BEGIN
        NOTLOADED(IF Y.[7:1] THEN 27 ELSE 35);
        Y:=1;
    END;
    END ELSE % CHECK RECORD SIZE
    IF Y.[7:1] THEN
    BEGIN
        IF (Y:=((M[AROW[W] INX NOT 0]#900+B6500) AND N<N1)) THEN
            NOTLOADED(39);
        END ELSE Y:=0;
    EXT: CHECK + Y;
    END; % OF CHECK
    *****
    SUBROUTINE IO;
    BEGIN
        IF IC THEN
        BEGIN
            IOREQUEST(=(IOD[W]:=AROW[W] INX @540000000)
                &(SIZE+B6500)[8:38:10]&TINU[U][3:3:5])OR @2017700000,
                IOD[W],[IOD[W]]&U[12:42:6]);
            N:=N+30;
        END ELSE
        BEGIN
            DISKIO(IOD[W],AROW[W] INX B6500=1,(T:=IF(T=LAST-Q+1)
                LSS 30 THEN 30*T ELSE 900),Q);
            Q:=Q+30;
        $ SET OMIT = NOT(STATISTICS)
            COUNTUP(18,T DIV 30);
        $ POP OMIT
        END;
    END; % OF IO
    *****

```

```

28106000
28107000
28108000
28109000
28110000
28111000
28112000
28113000
28114000
28115000
28116000
28117000
28118000
28119000
28120000
28121000
28121500
28121600
28122000
28123000
28124000
28125000
28126000
28127000
28128000
28129000
28129100
28130000
28131000
28132000
28135000
28136000
28137000
28138000
28139000
28140000
28141000
28143000
28144000
28145000
28146000
28147000
28148000
28149000
28150000
28151000
28152000
28153000
28154000
28155000
28156000
28157000
28158000
28159000
28159099
28159100
28159101
28160000
28161000
28162000

```



BOOLEAN SUBROUTINE LOADAROW;	28163000
BEGIN	28164000
SIZE:=900;	28164500
N1:=	28165000
\$ SET OMIT = NOT B6500LOAD	28165490
IF K=LASTROW,[CF] THEN LASTROW,[9:24] ELSE	28165500
\$ POP OMIT	28165510
H[8];	28166000
LAST:=(Q:=H[K+9])+N1-1;	28166500
IOD[1]:=N2:=W:=N1=0;	28167000
IC:=1;	28168000
IO; % FIRE UP FIRST TAPE READ	28169000
W +1;% SWAP BUFFERS	28170000
IF N<N1 THEN% CANNOT DO ROW WITH ONE READ	28171000
LOOP:	28172000
WATE:	28173000
M[P(.,.LIBRARYLOAD)],[CF]+ALPHA;	28173500
COMPLEXSLEEP((((IOD[0] OR IOD[1]) AND IOMASK)#0) OR DSED);	28174000
IF DSED THEN ABORT;	28175000
N2=N2+15;% COUNT NUMBER OF OPERATIONS COMPLETED	28176000
W=IF (IOD[0] AND IOD[1] AND IOMASK)#0 THEN 1=W ELSE	28177000
((IOD[1] AND IOMASK)#0);	28178000
IF NOT(Y*CHECK) THEN% NO ERRORS WERE DETECTED	28179000
BEGIN% KEEP GOING	28180000
IC+1=IOD[W],[7:1];% SWAP UNITS	28181000
IF N<N1 THEN GO TO LOOP;% ROW IS NOT FINISHED	28182000
IF N2+30>N THEN SIZE+T; % CHANGE SIZE FOR LAST RECORD	28183000
IF IOD[W],[24:1] THEN GO TO LOOP;% MORE WRITING TO DO	28184000
IF N2<N THEN BEGIN IOD[W]+0; GO TO WATE END;%	28185000
END;%	28186000
LOADAROW:=Y;	28187000
END; % OF LOADAROW	28188000
*****	28189000
SUBROUTINE SPACIT;%	28190000
BEGIN %	28191000
BACK: WHILE WAITIO(SPACER,MM,U)#@40 DO	28192000
BEGIN	28193000
IF DSED THEN ABORT;	28194000
CHECKFORSWAP;	28195000
END;	28196000
IF ENDOFREEL AND NOT SPACITSW THEN GO BACK;	28196100
IF SPACITSW THEN GO TO BAC; %BRANCH TO LABELCHECK ELSE EXIT.	28197000
END; % SPACIT	28198000
*****	28198500
ALPHA>(*P(.,.LIBRARYLOAD)),[CF];	28200000
INITIALSETUP;	28255000
DO	28256000
BEGIN	28257000
IF LABELCHECK THEN ABORT;	28258000
IF WAITIO((*[AROW[0]])&@5400[CTF],@2000000,U)=@20 THEN	28258050
GO TO BADHEADER; % RD HDR CKING FOR PARITY	28258100
MOVE(30+5*B6500,AROW[0],[CF]+B6500,H);	28258149
\$ SET OMIT = NOT(B6500LOAD)	28258150
IF B6500 THEN	28258160
BEGIN XX:=AROW[0];	28258170
IF (LASTROW:=B6500FORMATTER(CCT,I,X,XX,BCL,H,1))<0 THEN	28258180
GO TO BADHEADER;	28258190
END;	28258191
\$ POP OMIT	28259000
IF DSED THEN ABORT;	28259900
T:=1;	

IF (NOT B6500) AND (M[AROW[0] INX NOT 0] NEQ 30) THEN	28260000
GO TO BADHEADER ELSE	28260010
BEGIN	28260020
STREAM(A:=0;D:=H);	28260030
BEGIN SI:=D; 30(IF SB THEN BEGIN TALLY:=1; JUMP OUT END	28260040
ELSE SI:=SI+8); A:=TALLY;	28260050
END;	28260060
IF P THEN P(1) ELSE	28260070
IF (NT1:=H[9],[43:5])>20 OR NT1=0 THEN P(1) ELSE	28260080
BEGIN I:=0;	28260090
FOR W:=10 STEP 1 UNTIL 29 DO	28260100
BEGIN	28260110
I:=I+(NT2:=(H[W] NEQ 0));	28260120
IF W GEQ NT1 +10 THEN IF NT2 THEN W:=31;	28260130
END;	28260140
IF ((W=31) OR (I GTR NT1) OR((I NEQ 0) AND (H[8]=0)))	28260150
THEN P(1) ELSE P(0);	28260160
END END;	28260180
IF P THEN	28260190
BEGIN	28260200
BADHEADER:	28260250
NOTLOADED(43);	28260300
H[2] ← LAB[2];	28260500
SPACIT;	28260600
IF H[2]≠LAB[2] THEN ABORT; & FOR WE ARE LOST	28260700
GO TO ENDLOOP;	28260800
END ELSE	28260900
IF X[J],[1:1] THEN	28261000
IF (X[J],[2:4] NEQ 0 OR X[J+1],[1:5] NEQ 0 OR ABS(X[J])=0114)	28261100
THEN NOTLOADED(37) ELSE	28261110
IF (T←DIRECTORYSEARCH(X[J]&(3+4*(TYPE=ADDV)))[1:45:3],X[J+1],	28262000
4+(TYPE=ADDV))) GEQ 2 THEN	28263000
IF T=2 THEN	28264000
NOTLOADED(25) ELSE	28265000
BEGIN	28267000
LOADING:=9;	28268000
IF DSED THEN ABORT;	28269000
IF (I:=TYPE NEQ ADDV AND M[T+2] NEQ 0 AND	28269500
((USERCODE[P1MIX] EQV ABS(MCP)) NEQ NOT 0) AND	28270000
((USERCODE[P1MIX] EQV ABS(M[T+2])) NEQ NOT 0)) OR	28271000
(LATEST AND M[T+3],[30:18] GTR H[3],[30:18])) THEN	28272000
BEGIN	28273000
HEADERUNLOCK(ABS(X[J]),X[J+1],T);	28274000
T:=-1;	28275000
NOTLOADED(64-I×23);	28276000
END;	28278000
END ELSE	28279000
IF T=1 THEN & IT WAS "QT=ED"	28280000
BEGIN	28281000
T:=-1;	28282000
NOTLOADED(45);	28283000
END ELSE IF DSED THEN ABORT;	28285000
IF T=0 OR (T GEQ 64 AND TYPE NEQ ADDV) THEN	28286000
BEGIN	28287000
IF T GEQ 64 THEN	28288000
IF M[T+8]≠H[8] THEN	28289000
BEGIN	28290000
FORGETSPACE(T);	28291000
P(DIRECTORYSEARCH(X[J],X[J+1],6),DEL);	28292000
T←0;	28293000

END;	28294000
IF T=0 THEN	28295000
BEGIN	28296000
T:=GETSPACE(30,0,1)+2;#FIXES POSSIBLE PROBLEM	28297000
MOVE(30,T-1,T);	28298000
M[T+4]*=0&SYSNO[4:46:2]&1[2:47:1];	28299000
END ELSE	28302000
EADD*T,[18:15];	28303000
LBL ← [M[T]] & 30[8:38:10];	28304000
FOR W:=H[9],[43:5]+10 STEP 1 UNTIL 29 DO H[W]:=0;	28305000
IF (LBL[9]:=(P(DUP)) AND 31) = 0 THEN LBL[7]:=-1;	28305500
FOR W:=LBL[9]+10 STEP 1 UNTIL 29 DO LBL[W]:=0;	28306000
W:=0;	28306500
WHILE (W+W+1) LEQ H[9],[43:5] DO	28307000
IF H[9+W]≠0 THEN	28308000
IF (H[9+W]:=LBL[9+W]) LEQ 0 THEN	28309000
IF (H[9+W]:=PETUSERDISK(H[8] OR M,COMMON)) LSS 1 THEN	28310000
BEGIN	28311000
I←SPACE(10);	28311200
STREAM(J←JARROW[P1MIX],P1MIX,H←H[8],M←X[J],F←X[J+1],	28311210
I);	28311220
BEGIN DS←14 LIT "#NO USER DISK!"; SI←J;SI←SI+1;	28311230
DS←7CHR;DS←LIT"/";SI←SI+1;DS←7CHR;	28311240
DS←LIT"=";SI←LOC P1MIX;DS←2DEC;J←DI;DI←DI-2;DS←FILL;	28311250
DI←J;DS←LIT"(";SI←LOC M;SI←SI+1;DS←7CHR;SI←SI+1;	28311260
DS←LIT"/";DS←7CHR;DS←2LIT"=";SI←LOC H;DS←8 DEC;	28311270
DS←7LIT" SEGS,.";DI←DI-15;DS←7FILL;	28311280
END;	28311290
SPOUT(I);	28311300
REPLY[P1MIX] := "(I:=VIF&VWY[36:42:6]&	28311400
VOF[30:42:6]&VOK[24:42:6]);	28311410
SWAP(WAITSWAP,1);	28311500
IF NOT WHYSLEEP(I) THEN GO TO WY;	28311700
IF REPLY[P1MIX]=VOK THEN GO TO OK;	28311800
IF REPLY[P1MIX]=VOF THEN	28311910
BEGIN COMMON ← COMMON AND NOT M; GO TO OK; END;	28311920
FOR W:=W STEP -1 UNTIL 1 DO	28314000
IF H[9+W]≠0 THEN	28315000
IF LBL[9+W]=0 THEN	28316000
FORGETUSERDISK(H[9+W],H[8]);	28317000
FORGETSPACE(T);	28318000
IF DSED THEN ABORT;	28318200
IF LBL[9]≠0 THEN	28318600
P(DIRECTORYSEARCH(X[J],X[J+1],14),DEL);	28318800
NOTLOADED(31);	28319000
IF X[J+2]≠@14 THEN SPACIT;	28321000
GO TO ENDL00P;	28322000
END;	28323000
STREAM(A←[H[1]],D←DATE);	28328000
BEGIN SI←LOC D;DI←LOC D;DS←8 OCT;	28329000
SI←SI-4;DI←A;DS←4 CHR;	28330000
END;	28331000
H[4]:=M[T+4]&H[4][8:8:3]&0[11:47:1]&H[4][36:36:6]	28332000
&H[4][43:43:1];	28332100
H[1],[25:23]+XCLOCK+P(RTR);	28333000
IF LBL[9]=0 THEN	28333100
ENTERUSERFILE(ABS(X[J]),X[J+1],H,[CF]-1)	28333200
ELSE	28333300
BEGIN W:=IF H[9] LSS LBL[9] THEN LBL[9] ELSE H[9];	28333400
FOR W:=W+9 STEP -1 UNTIL 10 DO	28333500
OK;	
WY;	

	IF H[W]=0 THEN	28333600
	IF LBL[W]≠0 THEN       % EXTRA ROW IN DISK FILE	28333700
	FORGETUSERDISK(LBL[W],LBL[8]) ELSE ELSE	28333800
\$ SET OMIT = NOT SHAREDISK		28333890
	IF LBL[W]=0 THEN       % NEW ROW	28333900
	SCRATCHDIRECTORYDELETE(H[W],H[8]);	28334000
\$ POP OMIT		28334010
	DISKWAIT(H INX 0,30,EADD);	28334100
	END;	28334200
	FORGETSPACE(T);	28334500
	LOADING:=TRUE;	28335000
	FOR K:=1 STEP 1 UNTIL H[9],[43:5] DO	28336000
	IF H[K+9]≠0 THEN % ROW IS ASSIGNED	28337000
	IF LOADAROW THEN % THERE WAS AN ERROR	28338000
	BEGIN MIP(.,,LIBRARYLOAD)),[CF]:=ALPHA;	28339000
	COMPLEXSLEEP((IOD[0]=0 OR (IOD[0] AND IOMASK)≠0)	28339100
	AND(IOD[1]=0 OR (IOD[1] AND IOMASK)≠0	28339110
	) OR DSED);	28339200
BADLOAD:		28339500
	BLASTQ(U);	28340000
	P(WAITIO(@004740000020,@377,U),DEL);	28340100
	LOADING:=FALSE;	28341000
	P(DIRECTORYSEARCH(X[J],X[J+1],6),DEL);	28342000
	IF X[J+2]≠@14 THEN SPACIT;	28343000
	GO ENDLOOP;	28344000
	END ELSE CHECKFORSWAP;	28345000
\$ SET OMIT = NOT B6500LOAD		28345040
	IF NOT B6500 THEN	28345050
\$ POP OMIT		28345060
	IF WAITIO(SPACER,MM,U) ≠ @40 THEN	28345100
	BEGIN NOTLOADED(56);	28345200
	GO TO BADLOAD;	28345400
	END;	28345500
	P(DIRECTORYSEARCH(X[J],X[J+1],14),DEL);	28346000
	BB ← TRUE; % MARK THAT A FILE HAS BEEN LOADED	28346100
	LBMESS(ABS(X[J]),X[J+1],1,0,TINU[U],SPOUTUNIT,LIBMSG);	28348000
	LOADING:=FALSE;	28349000
\$ SET OMIT = NOT B6500LOAD		28349890
	IF B6500 THEN GO TO SKIPPER ELSE	28349900
\$ POP OMIT		28349910
	GO FALLOUT;	28350000
	END;	28351000
	IF X[J+2]≠@14 THEN GO ENDLOOP ELSE	28352000
	FOR K:=1 STEP 1 UNTIL H[9],[43:5] DO	28353000
	IF H[K+9]≠0 THEN	28354000
	BEGIN	28355000
	CHECKFORSWAP;	28356000
	N:=W:=SIZE:=LOADING:=0;	28357000
	IC:=1;	28358000
	IO; % READ HEADER	28359000
	W:=1;	28360000
	FOR N1←H[8] DIV 30 + ((H[8] MOD 30)≠0) STEP -1 UNTIL 2 DO	28361000
	BEGIN % SPACE OVER FILE	28362000
	IO;	28363000
LAY:	MIP(.,,LIBRARYLOAD)),[CF]←ALPHA;	28364000
	COMPLEXSLEEP(((IOD[0] OR IOD[1]) AND IOMASK)≠0	28364500
	OR DSED);	28365000
	IF DSED THEN ABORT;	28366000
	W←((IOD[1] AND IOMASK)≠0);	28367000
	IF IOD[W],[27:2]≠0 THEN	28368000

```

                IF IOD[W],[27:1] THEN % END OF REEL                28369000
                BEGIN                                             28370000
                    IF NOT ENDOFREEL THEN                          28371000
                        BEGIN                                       28371100
                            $ SET OMIT = NOT B6500LOAD            28371120
                            IF LASTROW,[CF]=K THEN                28371140
                                IF LASTROW,[9:24] LEQ N THEN GO ENDL00P; 28371160
                            $ POP OMIT                             28371180
                                P(WAITIO(@4740000020,@377,U),DEL, 28371200
                                    WAITIO(@4740000020,@377,U),DEL); 28371250
                                SPACIT;                            28371300
                                GO TO ENDL00P;                    28371350
                            END;                                    28371400
                                IO;                                28372000
                                W:=1-W;                            28373000
                                IF (IOD[W] AND IOMASK)=0 OR IOD[W],[27:1] THEN 28374000
                                    N1:=N1+1;                      28375000
                                END ELSE % PARITY ERROR           28376000
                                    BEGIN                           28377000
                                        SPACIT;                    28378000
                                        GO ENDL00P;                28379000
                                    END;                             28380000
                                IF N1>=0 THEN % WAIT FOR LAST READ TO FINISH 28381000
                                    BEGIN                           28382000
                                        N1:=(-1);                 28383000
                                        IOD[W]:=0;                 28384000
                                        GO LAY;                     28385000
                                    END;                             28386000
                                END;                               28387000
                            END;                                   28388000
                        SKIPPER; DO UNTIL WAITIO(SPACER,MM,U)=@40; 28388500
                        FALLOUT; WHILE ENDOFREEL DO P(WAITIO(SPACER,@40,U),DEL); 28389000
                        ENDL00P; LOADING:=FALSE;                   28390000
                            IF DSED THEN ABORT;                   28391000
                        END UNTIL X[J:=J+2]=@14;                   28392000
                            IF NOT BB THEN                         28392200
                                BEGIN                               28392300
                                    STREAM(BB:=BB:=SPACE(5));    28392400
                                    BEGIN DS*18LIT*NULL LIBRARY LOAD*"; END; 28392500
                                    SPOUT (BB);                    28392600
                                END;                                 28392700
                            STOPTIMING(0,1023);                   28393000
                            STOPTIMING(5,1023);                   28394000
                            SETNOTINUSE(U,0);                     28395000
                            $ SET OMIT = PACKETS                  28395999
                                IF UNITNO#0 AND LABELTABLE[UNITNO]=@214 THEN 28396000
                                    LABELTABLE [UNITNO]=@114;    28397000
                            $ POP OMIT                             28397001
                                GO INITIATE;                       28398000
                        END; % OF LIBRARY LOAD PROCEDURE           28399000
                        PROCEDURE LIBRARYDUMP;                     28400000
                        BEGIN REAL ALPHA;                           28401000
                            REAL COMMON=-4;                       28402000
                            REAL I,J,T,U,UNITNO,DUMPING,W,Y,TM,REEL,IC,N,N1,LAST,N2; 28403000
                            REAL Q,MID,FID,MAX,EXP,GTRMAX,K,K1,SEG,MIDCTR,SIZE,TYPE; 28404000
                            REAL RC,B; % ONE IO                   28404001
                            ARRAY X[*],S[*],AROW[*],H[*],IOD[*],LBL[*]; 28405000
                        REAL TOGS;                                   28405100
                        DEFINE DISKPARITY = TOGS,[47:1]#;         28405200
                            DUMPDIR = TOGS,[46:1]#;             28405300

```

NOLBL	= TOGS,[45;1]#;	28405400
TAPEPARITY	= TOGS,[44;1]#;	28405500
SPOUTUNIT	= 0#;	28406000
DSED	= (TERMSET(P1MIX))#;	28406100
SPACER	= 5&@3400[CTF]#;	28406200
MM	= @37700040#;	28406300
LABEL TAPEPAR,PARERR,LOOP,WATE,DISPAR,GETONE,NEXTNAME,GETMORE,		28407000
WRITIT,BACK,BADHDR,NEXT;		28407100
LABEL EXIT;		28407500
%*****		28408000
DEFINE NOTDUMPED(NOTDUMPED1) =		28408100
BEGIN NT1:=NOTDUMPED1; NODUMPMESS; END#;		28408200
SUBROUTINE NODUMPMESS;		28408300
LBMESS(X[J],X[J+1],-3,NT1,IF DUMPING THEN TINU[U] ELSE 0,		28408400
SPOUTUNIT,1);		28408500
%*****		28408600
SUBROUTINE GETASEGMENT;		28409000
BEGIN		28410000
SEG:=S[29];		28411000
DISKWAIT(-S,[CF],30,SEG);		28412000
FORGETESPDISK(SEG);		28413000
I:=2;		28414000
END; % OF GETASEGMENT		28415000
%*****		28416000
SUBROUTINE CHECKFORSWAP;		28417000
BEGIN		28418000
IF (T:=PRTRROW[P1MIX],[PSF])#0 THEN		28419000
BEGIN		28420000
IF T=2 THEN		28421000
BEGIN		28422000
IOCOUNT[P1MIX]:=-1;		28423000
STOPM;		28424000
END ELSE		28425000
IF T#1 THEN SWAP(FORCESWAP,1);		28426000
END;		28427000
IF ELAPSEDLIMIT[P1MIX] GTR IOTIME[P1MIX] THEN		28428000
IF PROCLIMIT[P1MIX]>PROCTIME[P1MIX]+CLOCK+P(RTR) THEN GO EXIT;		28429000
FOR T:=SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO		28430000
IF ACTIVE[T]>1 THEN		28431000
BEGIN		28432000
IOCOUNT[P1MIX]:=-1;		28433000
SWAP(TIMEND,1);		28434000
ELAPSEDLIMIT[P1MIX]:=*P(DUP)+128;		28435000
PROCLIMIT[P1MIX]:=*P(DUP)+64;		28436000
GO TO EXIT;		28437000
END;		28438000
EXIT;		28439000
END; % OF CHECKFORSWAP		28440000
%*****		28441000
SUBROUTINE ABORT;		28442000
BEGIN		28442500
IF DUMPING THEN J:=J-2 ELSE		28443000
BEGIN X[J]:=@14;		28443500
J:=-2;		28444000
END;		28444500
WHILE X[J:=J+2]#@14 AND J LSS 1022 DO		28445000
P(DIRECTORYSEARCH(-X[J],X[J+1],13),DEL);		28446000
IF U GEQ 0 THEN		28447000
BEGIN		28448000
STOPTIMING(0,1023);		28449000

```

                STOPTIMING(5,1023);                28450000
                BLASTQ(U);                          28451000
                SETNOTINUSE(U,0);                   28452000
            END;                                     28453000
            WHILE S[29]#0 DO GETASEGMENT;           28454000
$ SET OMIT = PACKETS                               28454999
            IF UNITNO#0 AND LABELTABLE[UNITNO]=@214 THEN 28455000
                LABELTABLE[UNITNO]!=@114;         28456000
$ POP OMIT                                         28456001
            GO TO INITIATE;                         28457000
        END; % OF ABORT                            28458000
        %*****28458500
        SUBROUTINE SPACIT;                          28458600
        BEGIN                                       28458700
            WHILE WAITIO(SPACER,MM,U) # @40 DO     28458800
                BEGIN                               28458900
                    IF DSED THEN ABORT;           28459000
                    CHECKFORSWAP;                 28459100
                END;                                28459200
                P(WAITIO([TM],@40,U),DEL);        28459300
                % WRITE THE TM BACK
            END; % OF SPACIT                        28459400
        %*****28459500
        SUBROUTINE WRITENDINGLABEL;                28460000
        BEGIN                                       28461000
            P(WAITIO([TM],@40,U),DEL);            28462000
            IF DSED THEN ABORT;                   28463000
            P(WAITIO(LBL&@5000[CTF],@40,U),DEL); 28464000
            IF DSED THEN ABORT;                   28465000
        END; % OF WRITE ENDING LABEL              28466000
        %*****28467000
        SUBROUTINE CHECK;                          28468000
        BEGIN                                       28469000
            IF (Y:=IOD[W]).[27:2]#0 THEN % PARITY ERROR OR EOT 28470000
                BEGIN                               28471000
                    IF Y.[7:1] AND Y.[27:1] THEN % END OF TAPE 28472000
                        BEGIN                         28473000
                            IF IOD[1-W].[7:1] THEN 28474000
                                BEGIN                 28475000
                                    SLEEP([IOD[1-W]],IOMASK); 28476000
                                    IF IOD[1-W].[28:1] THEN GO PARERR; 28477000
                                    IOD[1-W].[27:1]!#0; 28478000
                                END;                   28479000
                            END;
                        END;
                    LBL[4]!=(+P(DUP)) OR 1;        28480000
                    IF LBL[2].[30:18]=0 THEN %FILE000 LAST FILE 28480100
                        STREAM(A+(J+4) DIV 2,B+[LBL[2]]); 28480200
                    BEGIN SI=LOC A; DI=DI+5; DS=3 DEC END; 28480300
                    P(WAITIO([TM],@40,U),DEL);    28481000
                    P(WAITIO(LBL&@5000[CTF],@40,U),DEL); 28482000
                    P(WAITIO([TM],@40,U),DEL);    28483000
                    SETNOTINUSE(U,1);            28484000
                    STOPTIMING(5,1023);          28485000
                    LBL[4]!=(+P(DUP)) AND NOT(1); 28486000
                    STREAM(REEL:=REEL:=REEL+1,LBL); 28487000
                    BEGIN SII=LOG REEL;          28488000
                        DI:=DI+24; DS:=3 DEC;     28489000
                    END;                          28490000
                    IF (U:=LABELASCATCH(LBL)) LSS 0 THEN ABORT; 28491000
                    DUMPDIR=TRUE; %DUMP DIRECTORY 28491100
                    STARTIMING(5,U);             28492000
                END ELSE % PARITY ERROR           28493000

```

TAPEPAR:

```

PARERR:      BEGIN
              IF Y,[7:1] THEN      % TAPE
              BEGIN
                SPACIT;
                P(WAITIO(H&@5000[CTF],@40,U),DEL);
                TAPEPARITY:=TRUE;
                GO TO TAPEPAR;
              END;
              DISKPARITY:=TRUE;
            END;
          END;
        END; % OF SUBROUTINE CHECK
        %*****
        SUBROUTINE IO;
        BEGIN
          IF IC THEN
            IOREQUEST(-(IOD[W]-(AROW[W] INX @500000000))&
              SIZE[8:38:10]&TINU[U][3:3:5]) OR @2017700000,
            IOD[W],[IOD[W]]&U[12:42:6])
          ELSE
            BEGIN
              DISKIO(IOD[W],-(AROW[W] INX 0=1),(T+IF (T+LAST=Q=N+1) LSS
                30 THEN 30*T ELSE 900),Q+N);
              N:=N+30;
            $ SET OMIT = NOT(STATISTICS)
              COUNTUP(18,T DIV 30);
            $ POP OMIT
              END;
            END; % OF IO
            %*****
            SUBROUTINE DUMPAROW;
            BEGIN
              N1 ← H[8]; SIZE ← 900;%
              LAST←(Q+H[K+9])+H[8]-1;%
              IOD[1]:=N2:=W:=N:=IC:=RC:=0;
              IO; % FIRE UP FIRST DISK READ
              W ←1;% SWAP BUFFERS
              IF B←(N<N1) THEN% CANNOT DO ROW WITH ONE READ
            LOOP:  IO;
            WATE:  M(P(.,LIBRARYDUMP)),[CF]*ALPHA;
              IF B      THEN COMPLEXSLEEP((((IOD[1]=W) AND IOMASK)≠0) OR DSED)
            ELSE % 1 IO
              COMPLEXSLEEP((((IOD[RC])AND IOMASK)≠0) OR DSED);
              IF DSED THEN ABORT;
              N2←N2+15;% COUNT NUMBER OF OPERATIONS COMPLETED
              W←IF (IOD[0] AND IOD[1] AND IOMASK)≠0 THEN 1=W ELSE
                ((IOD[1] AND IOMASK)≠0);
              CHECK;
              IF DISKPARITY OR TAPEPARITY THEN GO TO DISPAR;
              IC←1=IOD[W],[7:1];% SWAP UNITS
              IF N<N1 THEN GO TO LOOP;% ROW IS NOT FINISHED
              IF N2+30≥N THEN SIZE←T; % CHANGE SIZE FOR LAST RECORD
              IF IOD[W],[24:1] THEN GO TO LOOP;% MORE WRITING TO DO
            IF N2<N THEN BEGIN IOD[W]:=0;RC:=1=W;B:=0;GO TO WATE;END;%FIX ODD#
            DISPAR:  END;%OF DUMPAROW
            %*****
              ALPHA←(*P(.,LIBRARYDUMP)),[CF];
              S1=[M[SPACE(30)]]&30[8:38:10];
              TYPE:=COMMON,[FF];
              S[29]:=COMMON,[CF];

```

```

28494000
28495000
28495500
28496000
28496500
28497000
28497500
28498000
28499000
28500000
28501000
28502000
28503000
28504000
28505000
28506000
28507000
28508000
28509000
28510000
28511000
28512000
28513000
28514000
28514099
28514100
28514101
28515000
28516000
28517000
28518000
28519000
28520000
28521000
28522000
28523000
28524000
28525000
28526000
28527000
28527500
28527510
28527520
28528000
28529000
28530000
28531000
28532000
28532500
28533000
28534000
28535000
28536000
28537000
28538000
28539000
28539500
28540000
28541000
28542000

```



GETASEGMENT;	28543000
STREAM(MIDI:=MIDI:=S[1],BI=PRT[P1MIX,3]);	28544000
BEGIN DSI=16 LIT"ODIRCTRYODISK " ;DSI:=25 LIT"0" ;SI:=LOC MIDI;	28545000
SI:=SI+1 ;DSI:=7 CHR)DSI:=8 LIT"OF FILE000" ;DSI:=24 LIT"0" ;	28546000
END;	28547000
UNITNO:=S[0],[2:6];	28548000
X:=M[SPACE(1023)]&1023[8:38:10];	28549000
MAX=S[0],[CF];	28550000
EXP:=S[0],[8:2];	28551000
GTRMAX=S[0]<0;	28552000
X[0]:=014;	28553000
MOVE(1022,X,[X[1]]);	28554000
UI:=1;	28554500
GETONE;	28555000
IF DSED THEN ABORT;	28555500
IF I>26 THEN GETASEGMENT;	28556000
IF (S[I] OR S[I+1])<0 THEN SEEKNAM(S[I],S[I+1],W,X[J],X[J+1],Y) ELSE	28557000
BEGIN	28558000
X[J]:=S[I];	28559000
X[J+1]:=S[I+1];	28560000
W:=1;	28561000
END;	28562000
IF W#0 THEN	28563000
BEGIN	28564000
T=0;	28565000
K=1;	28566000
FOR N=J+2 STEP=2 UNTIL 0 DO	28567000
IF (X[J] EQV X[N])=NOT 0 THEN	28568000
IF (X[J+1] EQV X[N+1])=NOT 0 THEN GO TO NEXTNAME;	28569000
IF GTRMAX THEN	28569200
BEGIN	28569300
J:=J+2;	28569400
GO TO NEXTNAME;	28569500
END ELSE	28569600
IF NOT SYSTEMFILE(X[J],X[J+1]) THEN	28570000
IF (T:=DIRECTORYSEARCH(X[J]&1[3:47:1],X[J+1] OR M,3))	28571000
LSS 64 THEN	28571002
BEGIN	28572000
IF DSED THEN ABORT;	28573000
IF T=1 THEN NOTDUMPED(45) ELSE IF T NEQ 2 THEN K:=0	28574000
ELSE NOTDUMPED(25);	28576000
GO TO NEXTNAME;	28577000
END;	28578000
IF T GEQ 64 THEN	28579000
BEGIN	28580000
IF M[T+2]#0 THEN	28581000
IF (USERCODE[P1MIX] EQV ABS(MCP))#NOT 0 THEN	28582000
IF (USERCODE[P1MIX] EQV ABS(M[T+2])) NEQ NOT 0 THEN *	28583000
BEGIN	28584000
P(DIRECTORYSEARCH(-X[J],X[J+1],13),DEL);	28585000
NOTDUMPED(41);	28586000
GO TO NEXTNAME;	28588000
END;	28589000
IF EXP NEQ 0 THEN	28590000
BEGIN	28591000
IF EXP THEN	28592000
BEGIN	28593000
STREAM(T:=0 ;A:=CALCULATEPURGE(-M[T+3],[2:10]));	28594000
BEGIN SI:=LOC A ;DI:=LOC T ;DSI:=8 OCT END;	28595000
IF P GTR M[T+3],[12:18] THEN J:=J+2 ELSE	28596000

```

                P(DIRECTORYSEARCH(=X[J],X[J+1],13),DEL); 28597000
            END ELSE 28598000
            BEGIN 28599000
                IF MIT+4],[11;1] THEN J:=J+2 ELSE 28600000
                P(DIRECTORYSEARCH(=X[J],X[J+1],13),DEL); 28601000
            END; 28602000
        END ELSE J:=J+2; 28603000
    END; 28605000
NEXTNAME: 28607000
    IF (S[I] OR S[I+1])<0 THEN I:=I-2 ELSE W:=0; 28608000
    IF T GEQ 64 THEN FORGETSPACE(T); 28609000
END; 28610000
IF K LSS 1 THEN LBMESS(S[I],S[I+1],-3,15,0,SPOUTUNIT,1); 28612000
IF S[I+1+2]#@14 THEN 28614000
IF J<(2*MAX) OR GTRMAX THEN GO GETONE ELSE 28615000
BEGIN 28616000
    LBL:=M[SPACE(30)]&30[8:38:10]; 28617000
    J:=2; 28618000
    LBL[0]:=MAX; 28619000
    LBL[1]=MID; 28620000
    LBL[28]=0; 28621000
    STREAM(A=MIDCTR:ONE+1,MID+(MID)); 28622000
    BEGIN SI=LOC A;SI=SI+7;IF SC="0" THEN 28623000
        BEGIN TALLY+2;SI=MID;SI=SI+2;5(IF SC=" " THEN JUMP OUT; 28624000
            SI=SI+1;TALLY=TALLY+1);A=TALLY;DI=DI+A;DS=LIT"1"; 28625000
        END ELSE BEGIN DI=DI+A;SI=SI+16;DS=ADD;END; 28626000
    END; 28627000
    MIDCTR=P; 28628000
    COMMON=SEG+GETESPDISK; 28629000
    COMMON,[FF]=TYPE; 28630000
GETMORE: 28631000
    FOR K=2 STEP 2 UNTIL 26 DO 28632000
    BEGIN 28633000
        LBL[K]=X[J+J+2]; 28634000
        IF LBL[K]#@14 THEN GO WRITIT; 28635000
        LBL[K+1]=X[J+1]; 28636000
    END; 28637000
    28638000
    28639000
    28640000
WRITIT: 28641000
    LBL[29]=IF K# 28 THEN 0 ELSE GETESPDISK; 28642000
    DISKWAIT(LBL INX 0,30,SEG); 28643000
    IF K=28 THEN 28644000
    BEGIN 28645000
        SEG=LBL[29]; 28646000
        GO GETMORE; 28647000
    END; 28648000
    FORGETSPACE(LBL); 28649000
    LBMESS("LIBMAIN","DISK ",50,0,0,SPOUTUNIT,1); 28650000
    T:=GETSPACE(12,64,5)+4; 28651000
    IF (J=USERCODE[P1MIX])=ABS(NOT 0) THEN 28652000
    BEGIN 28653000
        J=0; 28654000
        K=31; 28655000
    END ELSE K=26; 28656000
    STREAM(J,COMMON,T); 28657000
    BEGIN 28658000
        DS=8 LIT"CC USER=";SI=LOC J;SI=SI+1;DS=7 CHR; 28659000
        DS=29 LIT";EXECUTE LIBMAIN/DISK;COMMON="; 28660000
        DS=8 DEC; DS=6 LIT";END,=";

```

```

END;
$ SET OMIT = NOT(PACKETS)
IF PSEUDOMIX[P1MIX] GEQ 32 THEN
    NYLONZIPPER[P1MIX],[2:1]+0;
$ POP OMIT
T←T&P1MIX[18:42:6]&K[3:43:5];
CCARD(T);
$ SET OMIT = NOT(PACKETS)
IF PSEUDOMIX[P1MIX] GEQ 32 THEN
    IF MEMROW[P1MIX],[CF] GEQ FENCE THEN
        DO SWAP(WAITSWAP,1) UNTIL NYLONZIPPER[P1MIX],[2:1] ELSE
        SLEEP([NYLONZIPPER[P1MIX]],@1000000000000000);
$ POP OMIT
J←T+0;
GO GETONE;
END;
X[J]←@14; % MARK END OF DIRECTORY
IF J=0 THEN
BEGIN
    STREAM(MID,D:=I:=SPACE(10));
    BEGIN DS:=14 LIT"-NULL LIBRARY ";SI:=LOC MID;
        SI:=SI+1;DS:=7 CHR; DS:=LIT"+";
    END;
    SPOUT(I);
    GO TO INITIATE;
END;
TM←0&"≥+"[1:37:11];%
IF J LSS 900 THEN
BEGIN
    T:=SPACE(J+1);
    MOVE(J+1,X,T);
    FORGETSPACE(X);
    X:=[M[T]]&(J+1)[8:38:10];
END;
REEL:=1;
LBL:=[M[TAPELABEL(MID,FID:="FILE000",1,1,100)]]&10[8:38:10];
IF (U:=LABELASCRATCH(LBL)) LSS 0 THEN ABORT;
STARTIMING(0,18);
STARTIMING(5,U);
P(WAITIO(X&(J+1)[8:38:10]&@5000[CTF],@40,U),DEL);
WRITENDINGLABEL;
AROW:= [M[SPACE(2)]]&2[8:38:10];
AROW[0]:=[M[SPACE(900)]]&900[8:38:10];
AROW[1]:=[M[SPACE(900)]]&900[8:38:10];
IOD:= [M[SPACE(2)]]&2[8:38:10];
J:=2;
DUMPING:=TRUE;
WHILE X[J+J+2]#@14 DO %
BEGIN;
    STREAM(A+(J+2) DIV 2,B←[LBL[2]]);
    BEGIN SI:=LOC A;DI:=DI+5;DS:=3 DEC END;
    LABELTABLE[U]←LBL[2]; % ENTER FILE ID FOR OL MESSAGE
    HI=[M[DIRECTORYSEARCH(X[J],X[J+1],5)]]&30[8:38:10];
    H[9]:=(P(DUP)) AND 31;
    IF NOLBL THEN NOLBL←FALSE ELSE
    BEGIN
        P(WAITIO(LBL&@5000[CTF],@40,U),DEL);
        IF DSED THEN ABORT;
        P(WAITIO([TM],@40,U),DEL);
        IF DSED THEN ABORT;

```

```

28661000
28661099
28661100
28661200
28661201
28661300
28662000
28662099
28662100
28662200
28662300
28662400
28662401
28663000
28664000
28665000
28666000
28667000
28668000
28669000
28670000
28671000
28672000
28673000
28674000
28675000
28676000
28677000
28678000
28679000
28680000
28681000
28682000
28683000
28684000
28685000
28686000
28687000
28688000
28689000
28690000
28691000
28692000
28693000
28694000
28695000
28696000
28697000
28698000
28699000
28700000
28701000
28702000
28702100
28702200
28702300
28703000
28704000
28705000
28706000

```

END;	28706100
IF P([H[9]],LOD,DUP)=0 OR P(XCH)>20 THEN	28706500
GO TO BADHDR;	28706600
P(WAITIO(H&@5000[CTF],@40,U),DEL);	28707000
BACK: IF DSED THEN ABORT;	28708000
FOR K+1 STEP 1 UNTIL H[9] DO% WRITE OUT FILE, ROW BY ROW	28709000
IF H[K+9]#0 THEN	28710000
BEGIN	28711000
DUMPAROW;	28711200
IF TAPEPARITY THEN	28711300
BEGIN	28711400
TAPEPARITY:=FALSE;	28711500
GO BACK;	28711600
END;	28711700
IF DISKPARITY THEN	28711800
BEGIN	28711900
SPACIT;	28712000
IF DSED THEN ABORT;	28712190
BADHDR:	28712195
H:=H&20[8:38:10];	28712200
P(WAITIO(H&@5000[CTF],@40,U),DEL);	28712210
WRITENDINGLABEL;	28712220
P(DIRECTORYSEARCH(-X[J],X[J+1],13),DEL);	28712225
NOTDUMPED(IF DISKPARITY THEN 35 ELSE 43);	28712228
FORGETSPACE(H);	28712230
DISKPARITY:=FALSE;	28712233
GO TO NEXT;	28712235
END;	28712500
CHECKFORSWAP;	28713000
END;	28714000
FORGETSPACE(H);	28715000
WRITENDINGLABEL;	28716000
LBMESS(X[J],X[J+1],3,0,TINU[U],SPOUTUNIT,LIBMSG);	28717100
P(DIRECTORYSEARCH(-X[J],X[J+1],13),DEL);	28718000
IF DUMPDIR THEN	28718100
BEGIN X[0]+@114; X[1]+J+2; %FLAG X[0] AND OFFSET INTO X[1]	28718110
LBL[2],[30:18]:=0; %FILE000	28718120
P(WAITIO(LBL&@5000[CTF],@40,U),DEL);	28718140
IF DSED THEN ABORT;	28718150
P(WAITIO([TM],@40,U),DEL);	28718160
IF DSED THEN ABORT;	28718170
IOD[0]+0; W+1; SIZE+X,[8:10];	28718180
IOREQUEST(-(IOD[W]+(X INX @5000000000)&	28718182
SIZE[8:38:10]&TINU[U][3:3:5]) OR @2017700000,	28718184
IOD[W],[IOD[W]]&U[12:42:16]);	28718186
M[P(.,LIBRARYDUMP)],[CF]:=ALPHA;	28718187
COMPLEXSLEEP(((IOD[W]) AND IOMASK)#0) OR DSED);	28718188
IF DSED THEN ABORT;	28718190
CHECK;	28718192
IF NOT IOD[W],[27:1] THEN	28718194
BEGIN WRITENDINGLABEL; DUMPDIR+FALSE END ELSE NOLBL+TRUE;	28718200
END;	28718210
NEXT: END;%ALL FILES NOW WRITTEN	28719000
P(WAITIO([TM],@40,U),DEL);	28720000
STOPTIMING(0,1023);	28721000
STOPTIMING(5,1023);	28722000
SETNOTINUSE(U,1);	28723000
IF TYPE=UNLOAD THEN	28724000
FOR J+0 STEP 2 WHILE X[J]#@14 DO	28725000
IF DIRECTORYSEARCH(-X[J],X[J+1],7)=3 THEN X[J+2]:=@14;	28726000

```

IF UNITNO#0 AND LABELTABLE[UNITNO]=@214 THEN 28727000
    LABELTABLE [UNITNO]=@114; 28728000
GO INITIATE; 28729000
END; % LIBRARY MAINT. PROCEDURE 28730000
PROCEDURE LIBRARYZERO; 28800000
BEGIN 28801000
    REAL COMMON=-4; 28802000
    REAL TYPE,SEG,I,J,K,N1,Q,N,W,T,THING,ZEROING; 28803000
    ARRAY S[*],X[*],RESULT[*],BUFFADR[*],IOD[*],H[*]; 28804000
    LABEL GETONE,LOOP,WATE,EXIT,ARD; 28806000
    DEFINE DSED=(TERMSET(P1MIX))#; 28807000
    %***** 28807100
    SUBROUTINE CHECKFORSWAP; 28807120
    BEGIN 28807140
        IF (T:=PRTROW[P1MIX],[PSF])#0 THEN 28807160
            BEGIN 28807180
                IF T=2 THEN 28807200
                    BEGIN 28807220
                        IOCOUNT[P1MIX]:=-1; 28807240
                        STOPM; 28807260
                    END ELSE 28807280
                        IF T#1 THEN SWAP(FORCESWAP,1); 28807300
                END; 28807320
                IF ELAPSEDLIMIT[P1MIX] GTR IOTIME[P1MIX] THEN 28807340
                IF PROCLIMIT[P1MIX]>PROCTIME[P1MIX]+CLOCK+P(RTR) THEN GO EXIT; 28807360
                FOR T:=SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO 28807380
                IF ACTIVE[T]>1 THEN 28807400
                    BEGIN 28807420
                        IOCOUNT[P1MIX]:=-1; 28807440
                        SWAP(TIMEND,1); 28807460
                        ELAPSEDLIMIT[P1MIX]:=*P(DUP)+128; 28807480
                        PROCLIMIT[P1MIX]:=*P(DUP)+64; 28807500
                        GO TO EXIT; 28807520
                    END; 28807540
                EXIT; 28807560
            END; % OF CHECKFORSWAP 28807580
            %***** 28808000
            SUBROUTINE GETASEGMENT; 28809000
            BEGIN 28810000
                SEG:=S[29]; 28811000
                DISKWAIT(=S,[CF],30,SEG); 28812000
                FORGETESPDISK(SEG); 28813000
                I:=2; 28814000
            END; % OF GETASEGMENT 28815000
            %***** 28816000
            SUBROUTINE ABORT; 28817000
            BEGIN 28818000
                IF ZEROING THEN 28821000
                    BEGIN 28821500
                        H[4],[43:2]:=1; 28822000
                        H[4],[2:1]:=0; 28822500
                        DISKWAIT(THING,[CF],30,THING,[FF]); 28823000
                        FORGETSPACE(H); 28823500
                    END; 28824000
                    WHILE S[29] NEQ 0 DO GETASEGMENT; 28824250
                    GO INITIATE; 28824500
                END; % OF ABORT 28827000
                %***** 28828000
                SUBROUTINE IO; 28829000
                BEGIN 28830000

```

```

STREAM(DSKADR:=Q+N,DI=(BUFFADR INX (2*W)));
  BEGIN SI:=LOC DSKADR; DS:=8DEC; END;
RESULT[W]:=0;
IOREQUEST(=IOD[W]&@377[25:40:8],
  IOD[W]&(IF (T:=N1=N) LSS 63 THEN 512+T ELSE 512+63)
  [CTF],(W INX RESULT));
  N:=N+63;
END; % OF IO
*****
SUBROUTINE ZEROAROW;
BEGIN
  N1:=H[8];          %NO, OF SEGMENTS/ROW
  Q:=H[K+9];        %DISK ADDR OF ROW
  W:=0;             %BUFFER NO,
  N:=0;            %INDEX OF SEGMENTS
  IO;
  W:=1;            %SWAP BUFFERS
  IF N GEQ N1 THEN RESULT[1]:=RESULT[1] OR IOMASK ELSE
LOOP: IO;
WATE: COMPLEXSLEEP((((RESULT[1-W]) AND IOMASK)#0) OR DSED);
  IF DSED THEN ABORT;
  W+IF (RESULT[0] AND RESULT[1] AND IOMASK)#0 THEN 1=W ELSE
    ((RESULT[1] AND IOMASK)#0);
  IF N<N1 THEN GO TO LOOP;% ROW IS NOT FINISHED
  COMPLEXSLEEP((((RESULT[1-W]) AND IOMASK) NEQ 0) OR DSED);
  IF DSED THEN ABORT;
END;%OF ZEROAROW
*****
SI=[M[SPACE(30)]]&30[8:38:10];
XI=[M[SPACE(1023)]]&1023[8:38:10];
TYPE:=COMMON,[FF];
S[29]:=COMMON,[CF];
GETASEGMENT;
X[0]:=@14;
MOVE(1022,X,[X[1]]);
GETONE:
  IF DSED THEN ABORT;
  IF I>26 THEN GETASEGMENT;
  X[J]:=S[I];
  X[J+1]:=S[I+1];
  J:=J+2;
  IF S[I+I+2]#@14 THEN GO GETONE;
  IOD=[M[SPACE(8)]]&2[8:38:10];
  RESULT:=(2 INX IOD)&18[8:38:10];
  BUFFADR:=(4 INX IOD)&4[8:38:10];
  IOD[0]:=(BUFFADR INX 0)&1[8:38:10]&3[5:46:2];
  IOD[1]:=(BUFFADR INX 2)&1[8:38:10]&3[5:46:2];
  J:=2;
  ZEROING:=1;
  WHILE X[J+J+2]#@14 DO %
  BEGIN
    HI=[M[THING:=DIRECTORYSEARCH(X[J],X[J+1],5)]]&30[8:38:10];
    IF DSED THEN ABORT;
    IF THING=0 OR M[THING+4],[43:2]=3 THEN GO ARD;
    H[4]:=(+P(DUP))&3[43:46:2]&1[2:47:1]&SYSNO[4:46:2];
    DISKWAIT(THING,[CF],30,THING,[FF]);
    LBMESS(X[J],X[J+1],62,0,0,0,1);
    FOR K+1 STEP 1 UNTIL H[9],[43:5] DO% WRITE OUT FILE, ROW BY ROW
    IF H[K+9]#@0 THEN BEGIN ZEROAROW; CHECKFORSWAP; END;
    H[4],[43:2]:=0; % NO LONGER SENSITIVE OR BEING ZEROED

```

```

28831000
28831500
28831600
28832000
28832500
28833000
28833500
28834000
28834500
28835000
28835500
28836000
28836500
28837000
28837500
28838000
28838500
28839000
28847000
28848000
28849000
28850000
28851000
28852000
28852100
28852200
28853000
28854000
28855000
28856000
28857000
28858000
28859000
28860000
28861000
28862000
28863000
28864000
28865000
28866000
28867000
28868000
28869000
28870000
28871000
28872000
28873000
28877000
28878000
28879000
28880000
28881000
28882000
28882100
28882200
28882400
28883000
28884000
28885000
28886000

```

```

DISKWAIT(THING,[CF],30,THING,[FF]);
FORGETSPACE(H);
P(DIRECTORYSEARCH(X[J],X[J+1],6),DEL);
ARD:
END;
GO INITIATE;
END; % OF LIBRARYZERO
$ SET OMIT = NOT(AUXMEM)
PROCEDURE FILLORKILL(A, START, SIZE, TYPE);
VALUE A, START, SIZE, TYPE;
ARRAY A[*];
INTEGER START, SIZE;
BOOLEAN TYPE;
BEGIN COMMENT START IS A BIT INDEX IN ARRAY A
(WHICH IS EITHER A FINE TABLE OR A
COARSE TABLE), SIZE IS THE NUMBER
OF BITS TO TURN ON OR OFF, TYPE IS
USED TO TELL YOU WHICH TO DO, THIS
ROUTINE ALSO COMPUTES THE NEW "LONGRUN"
VALUE AFTER ALLOCATION OR RETURN,
;
INTEGER I, % CURRENT WORD INDEX IN ARRAY A
N, % LAST INDEX EXCEPT FOR OVERFLOW
X; % BIT INDEX OF LAST BIT IN LAST WORD
REAL J, % TEMPORARY STORAGE FOR A[I]
T, % TEMPORARY USED FOR RUN SEARCH
RUN, % SIZE OF LONG RUN ALREADY FOUND
RMASK, % MASK FOR NEXT LARGER RUN
MASK; % CONTAINS "AND NOT" OR "OR" MASK

DEFINE RETURNING = TYPE#; % =0 IF TO ALLOCATE
% =1 IF TO RETURN
DEFINE STOP = SIZE#; % LAST BIT TO ASSIGN OR RETURN
LABEL TRYNEXT, % SEARCH FOR LONGER RUN
FOUNDRUN, % STUFF RUN LENGTH IN
FINISHUP, % INCREMENT INDEX AND EXIT
WHOLEBUNCHOFBITS; % 2*32-1, USED IN MASKING

SUBROUTINE WHATEVERTURNSYOUON;
BEGIN T := J := A[I],[9:39];
IF RETURNING THEN J := J OR MASK
ELSE J := J AND NOT MASK;
COMMENT TURN BITS ON OR OFF--THIS IS THE
GUTS OF THE ALLOCATION/RETURN MECHANISM;
IF T=J THEN GO TO FINISHUP;
COMMENT IF T=J, NO BITS WERE CHANGED SO THE
PREVIOUS RUN COUNT IS CORRECT, THIS
MAY RESULT FROM "ALLOCATION" OF COARSE
ENTRIES IN THE FINE ALLOCATION PROCESS;
IF (T:=J)=(RUN:=0) THEN GO TO FOUNDRUN;
COMMENT IF J IS ZERO, ITS RUN IS OBVIOUSLY ZERO, TOO;
RMASK := 1;
COMMENT NOW TO LOOP AROUND FINDING THE LONGEST
XRUNX OF BITS IN THE WORD, NOTE THAT
ALTHOUGH THERE ARE REALLY TWO LOOPS,
THEY ARE MUTUALLY EXCLUSIVE, THE
METHOD USED IS TO TEST THE LOW ORDER
BITS AGAINST A MASK FOR EQUALITY,
INCREASING THE LENGTH OF THE MASK BY

```

```

28887000
28888000
28889000
28889550
28890000
28891000
28892000
28999999
29000000
29001000
29002000
29003000
29004000
29005000
29006000
29007000
29008000
29009000
29010000
29011000
29012000
29013000
29014000
29015000
29016000
29017000
29018000
29019000
29020000
29021000
29022000
29023000
29024000
29025000
29026000
29027000
29028000
29029000
29030000
29031000
29032000
29033000
29034000
29034100
29034200
29035000
29035100
29035200
29035300
29035400
29036000
29036100
29037000
29037100
29037200
29037300
29037400
29037500
29037600
29037700

```

```

                ONE BIT WHEN SUCCESSFUL AND SLIDING THE
                TEST WORD ONE BIT RIGHT OTHERWISE;
TRYNEXT:: IF (T AND RMASK)=RMASK THEN
    % IF EQUAL, THE RUN IS THERE
    IF (RUN:=RUN+1)=8 THEN GO TO FOUNDRUN
    % WE ONLY CARE ABOUT RUNS OF EIGHT OR LESS,
    % BECAUSE OF THE ALLOCATION STRATEGY AND
    % THE USE OF COARSE/FINE TABLES,
    ELSE RMASK := RMASK+P(DUP)+1
    % THIS INCREASES THE MASK LENGTH BY ONE BIT
ELSE T := T,[9:38];
    % IF THE RUN WAS NOT THERE, SLIDE TEST WORD RIGHT
    IF RMASK LEQ T THEN GO TO TRYNEXT;
    COMMENT IF RMASK IS GREATER THAN T, YOU OBVIOUSLY
    CANNOT FIND A LONGER RUN;
FOUNDRUN:: A[I] := J & RUN[3:42:6];
FINISHUP: I := I+1;
    MASK := P(WHOLEBUNCHOFBITS) & MASK[9:41:7];
    COMMENT BECAUSE OF THE SEVEN-BIT OVERLAP, YOU
    MUST MOVE THE SEVEN LOW-ORDER BITS
    OF THE MASK TO THE HIGH-ORDER BIT
    POSITIONS OF THE NEXT MASK, THIS CAUSES
    THE PROPER PROPAGATION OF "CARRIES,;"
END DOING YOUR OWN THING WHAT EVER IT IS;

I := START,[37:6]; N := ( STOP := START+SIZE-1),[37:6];
COMMENT I IS THE STARTING WORD INDEX IN A, AND
      N IS THE ENDING INDEX (EXCLUSIVE OF ANY
      POSSIBLE CARRYOVER);
MASK := TWO(32-START,[43:5])-1;
COMMENT TURN ON ALL BITS BELOW THE START POINT;
WHILE I<N DO WHATEVERTURNSYOUON;
COMMENT WHILE I IS LESS THAN N, TURN ON (OR OFF)
      ALL BITS CLEAR TO THE END, THE SUBROUTINE
      WILL REBUILD THE MASK AND INCREMENT I;
MASK := MASK-TWO(31-(X:=STOP,[43:5]))+1;
COMMENT THIS CODE TURNS OFF THOSE BITS FOLLOWING
      THE LAST TO THE END OF THE WORD, LEAVING
      THE HIGH ORDER PART UNCHANGED, NOTE THAT
      THE ORDER OF ARITHMETIC OPERATIONS IS
      IMPORTANT (TO AVOID NORMALISATION);
WHATEVERTURNSYOUON;
IF X>24 THEN % IT OVERFLOWS INTO THE NEXT WORD
BEGIN MASK := MASK AND NOT P(WHOLEBUNCHOFBITS);
      WHATEVERTURNSYOUON
END;
P(XIT);
WHOLEBUNCHOFBITS:: @377777777777;
END FILL OR KILL A FEW BITS;
INTEGER PROCEDURE AUXILIARYSPACE(SIZE);
VALUE SIZE;
INTEGER SIZE;
BEGIN COMMENT
    IF SUFFICIENT AUXILIARY MEMORY SPACE EXISTS
    (CONTIGUOUSLY, OF COURSE), AUXSPACE WILL
    ALLOCATE IT AND REMOVE IT FROM THE BIT TABLES,
    THE SEARCH HAS BEEN OPTIMIZED TO DETERMINE
    AVAILABILITY, AND WILL RETURN A ZERO IF THERE
    IS NO SUCH SPACE AVAILABLE, THE VALUE
    RETURNED IF SPACE IS ALLOCATED IS A PSEUDO-

```

```

29037800
29037900
29038000
29038100
29039000
29039100
29039200
29039300
29040000
29040100
29041000
29041100
29042000
29042100
29042200
29043000
29044000
29045000
29045100
29045200
29045300
29045400
29045500
29046000
29047000
29048000
29048100
29048200
29048300
29049000
29049100
29050000
29050100
29050200
29050300
29051000
29051100
29051200
29051300
29051400
29051500
29052000
29053000
29054000
29055000
29056000
29057000
29058000
29059000
29100000
29101000
29102000
29103000
29104000
29105000
29106000
29107000
29108000
29109000
29110000

```



```

DALOC ENTRY, OF A FORM NICE FOR STUFFING INTO
ABSENT DESCRIPTORS AND FOR CALLING UPON
ACTUALOVERLAYADDRESS WITH,
;
INTEGER CFRONT = +1;% PSEUDONYM FOR AUXILIARYSPACE
INTEGER CBITS, % NUMBER OF COARSE BITS NEEDED
MASK, % MASKS OFF PROPER RUN LENGTH
INDEX, % CTABLE OR FTABLE INDEX
I, N;
REAL J;
ARRAY COARSETABLE[*],
FINETABLE [*];
DEFINE LONGRUN = [3;6]#,
ALLOCATE(ALLOCATE1,ALLOCATE2,ALLOCATE3) =
FILLORKILL(ALLOCATE1,ALLOCATE2,ALLOCATE3,0)#;
SUBROUTINE FINESEARCH;
BEGIN COMMENT ATTEMPT TO ALLOCATE FROM ONE
TO SEVEN CONTIGUOUS BITS
FROM THE BACK OF THE FINE TABLE;
FOR I:=63 STEP -1 UNTIL 0 DO
IF FINETABLE[I],LONGRUN GEQ SIZE THEN
BEGIN J := FINETABLE[I],[9;39];
COMMENT AT LAST WE GET TO USE THE "LONGRUN" INDICATORS
WHICH WE HAVE SO PATIENTLY BUILT EVERY TIME
THROUGH FILLORKILL, THE MECHANISM IS
USED SO THAT THE WORST CASE (AUXILIARY
MEMORY NOT AVAILABLE) CAN BE HANDLED
WITH THE UTMOST DISPATCH, AND WILL CAUSE ONLY
MINOR DEGRADATION IN THE DISK ALLOCATION PROCESS;
MASK := TWO(SIZE)-1; N := 32-SIZE;
WHILE (J AND MASK)#MASK DO
BEGIN MASK := MASK+P(DUP); N := N-1 END;
COMMENT STARTING AT THE LOW END OF THE WORD,
SLIDE THE MASK LEFT UNTIL WE FIND
THE FIRST ACCEPTABLE RUN, BECAUSE OF
THE "LONGRUN" MECHANISM, WE KNOW
WE WILL FIND SUCH A SET OF BITS;
ALLOCATE(FINETABLE, CFRONT:=32*I+N, SIZE);
COMMENT ALLOCATE "SIZE" BITS FROM THE FINE TABLE;
ALLOCATE(COARSETABLE, CFRONT,[36;9],
((CFRONT,[45;3]+SIZE)>8)+1);
COMMENT ALLOCATE FROM THE COARSE TABLE EITHER
ONE OR TWO BITS, DEPENDING ON WHETHER
WE OVERLAP THE END OF A SET OF EIGHT
FINE BITS CORRESPONDING TO A GIVEN
COARSE BIT, NOTE THAT ONE (OR BOTH)
OF THESE BITS MAY ALREADY BE OFF, BUT
IT IS QUICKER TO "RE-ALLOCATE" THEM
THAN TO TEST FOR IT SPECIALLY;
I := 0
END
END;
SUBROUTINE COARSESEARCH;
BEGIN COMMENT SEARCHES FOR AND ALLOCATES
SPACE FROM COARSE AND FINE

```

```

29111000
29112000
29113000
29114000
29115000
29116000
29117000
29117100
29117200
29118000
29118100
29119000
29120000
29121000
29121100
29122000
29122100
29122200
29123000
29124000
29125000
29126000
29127000
29128000
29129000
29129100
29129150
29129200
29129300
29129400
29129500
29129600
29129700
29130000
29131000
29132000
29132100
29132200
29132300
29132400
29132500
29133000
29133100
29134000
29135000
29135100
29135200
29135300
29135400
29135500
29135600
29135700
29135800
29136000
29137000
29138000
29139000
29140000
29141000
29142000

```

```

TABLES, USING COARSE TABLE
TO FIND THE AREA, AND FINE
TABLE TO SQUEEZE OUT GAPS;
FOR I:=0 STEP 1 UNTIL 7 DO
IF (J := COARSETABLE[I]), LONGRUN GEQ CBITS THEN
BEGIN COMMENT THERE IS SUFFICIENT SPACE IN THIS WORD;
MASK := (NOT 0), [9:39]=TWO(39-CBITS)+1; N := -7;
WHILE (J AND MASK)≠MASK DO
BEGIN MASK := MASK, [9:38]; N := N+1 END;
COMMENT SLIDE MASK RIGHT UNTIL YOU FIND THE FIRST
SUFFICIENTLY LONG RUN, NOTE THAT BIT
SIXTEEN IS THE ZERO POINT, SINCE THE
PRECEDING BITS ARE "CARRIES";
CFRONT := 32×I+N;
COMMENT NOW CHECK FINE TABLE TO SEE
HOW FAR YOU CAN BACK UP
THIS ENTRY, USING THE
"NORMALISED" CFRONT TO
GIVE THE FINE INDEX,
NOTE THAT, SINCE WE TOOK
THE FIRST AVAILABLE SPACE
OF SUFFICIENT SIZE FROM
THE COARSE TABLE, WE CAN
BACK UP SEVEN OR LESS BITS
BY USING THE FINE TABLE, NOTE THAT
THE SEVEN-BIT OVERLAP IN EACH
WORD IS BOTH NECESSARY AND
SUFFICIENT.
;
STREAM(N:=0 & CFRONT[43:46:2]; % 8×(CFRONT MOD 4)
T:=FINETABLE[CFRONT, [39:7]]);
BEGIN SI:=LOG T; SI:=SI+1;
SKIP 3 SB; SKIP N SB;
7(IF SB THEN TALLY:=TALLY+1
ELSE TALLY:=0; SKIP SB);
NI=TALLY
END;
IF P(DUP)>(SIZE-1), [45:3]
THEN CBITS := CBITS-1;
COMMENT IF TEST WAS PASSED, WE SAVED A COARSE BIT;
ALLOCATE(COARSETABLE, CFRONT, CBITS);
CFRONT := P(CHS)+8×CFRONT;
ALLOCATE(FINETABLE, CFRONT, SIZE);
I := 7
END
END;

SUBROUTINE SEARCH;
BEGIN
IF (((INDEX=1) AND P(RRR), [30:1]) OR
((INDEX=0) AND P(RRR), [31:1])) THEN
IF NOT(UNIT[16+INDEX], [14:1]) THEN % CHECK FOR AUXMEM ERRORS
IF (COARSETABLE := CTABLE[INDEX]), [CF]≠0 THEN
BEGIN FINETABLE := FTABLE[INDEX];
IF SIZE<8 THEN FINESEARCH
ELSE COARSESEARCH
END;
END;

```

```

29143000
29144000
29145000
29146000
29147000
29148000
29149000
29150000
29151000
29152000
29153000
29154000
29155000
29156000
29157000
29158000
29159000
29160000
29161000
29162000
29163000
29164000
29165000
29166000
29167000
29168000
29169000
29170000
29171000
29172000
29173000
29174000
29175000
29176000
29177000
29178000
29179000
29180000
29181000
29182000
29183000
29184000
29185000
29186000
29187000
29188000
29189000
29190000
29190100
29190150
29190160
29190200
29191000
29192000
29193000
29194000
29195000
29195100
29196000
29197000

```

```

IF SIZE >1022 THEN P(0,RTN);
IF (SIZE := SIZE.[38:6]+1)<8 THEN INDEX := 1
    ELSE CBITS := (SIZE+7).[39:6];
CFRONT := -1; SEARCH;
IF CFRONT<0 THEN
BEGIN INDEX := 1-INDEX; SEARCH END;
IF CFRONT<0 THEN CFRONT := 0
    ELSE CFRONT := CFRONT & (14+INDEX)[33:44:4];
END AUXILIARYSPACE;
PROCEDURE FORGETAUXILIARYSPACE(SIZE,LOC);
VALUE SIZE, LOC;
INTEGER SIZE, LOC;
BEGIN COMMENT RETURNS AUXILIARY MEMORY SPACE
    TO THE FINE AND COARSE TABLES,
    EXAMINES BOUNDARY CONDITIONS TO
    DETERMINE WHETHER COARSE TABLE
    UPDATE IS REQUIRED, MARKS FINE
    TABLE ENTRIES IN ANY EVENT;

INTEGER FIRST,
    LAST,
    TABLE,
    INDEX;

ARRAY FINETABLE[*];

DEFINE RETURN(RETURN1,RETURN2,RETURN3) =
    FILLORKILL(RETURN1,RETURN2,RETURN3,1)#;

BOOLEAN SUBROUTINE NOTALLTHERE;
BEGIN;STREAM(N:=INDEX AND 24;
    T:=NOT FINETABLE[INDEX.[37:6]]);
    BEGIN SI:=LOC T; SI:=SI+2; SKIP 3 SB; SKIP N SB;
    8(SKIP SB; IF SB THEN TALLY:=1); N:=TALLY
    END;
    NOTALLTHERE := POLISH
END;

RETURN(FINETABLE:=FTABLE[TABLE:=LOC,[36:1]],
    LOC:=LOC.[37:11]+(SIZE:=SIZE,[38:6]+1));
FIRST := (INDEX := LOC).[37:8];
IF NOTALLTHERE THEN FIRST := FIRST+1;
IF (LAST := (INDEX := LOC+SIZE).[37:8])<FIRST THEN P(XIT);
IF NOTALLTHERE THEN
    IF (LAST:=LAST-1)<FIRST THEN P(XIT);
RETURN(CTABLE[TABLE], FIRST, LAST=FIRST+1);
END FORGETTING AUXILIARY MEMORY SPACE;
PROCEDURE AUXILIARYTABLEINITIALIZE;
BEGIN INTEGER AREA,
    INDEX;

LABEL CON1,
    CON2;

BOOLEAN B;
SUBROUTINE SETUPARRAYROW;
BEGIN COMMENT SETS UP COARSETABLE AND FINETABLE
    ENTRIES FOR AVAILABLE ROW "INDEX" (0 OR 1);
    IF AREA=0 THEN AREA := GETSPACE(74,0,1)+2;
    M[AREA+1] := P(CON2); MOVE(71,AREA+1,P(DUP)+1);

```

```

29198000
29199000
29200000
29201000
29202000
29203000
29204000
29205000
29206000
29207000
29300000
29301000
29302000
29303000
29304000
29305000
29306000
29307000
29308000
29308100
29309000
29310000
29311000
29312000
29312100
29313000
29313100
29314000
29315000
29316000
29317000
29318000
29319000
29320000
29321000
29322000
29323000
29324000
29325000
29326000
29327000
29328000
29329000
29330000
29331000
29332000
29333000
29334000
29400000
29401000
29402000
29403000
29404000
29405000
29406000
29410000
29411000
29412000
29413000
29414000

```

```

M[AREA+9] := M[AREA] := P(CON1); M[AREA+8] := 0;
CTABLE[INDEX] := [M[AREA J]] & 9[8:38:10] & AREA[CTF];
FTABLE[INDEX] := [M[AREA+9]] & 65[8:38:10];
END;

SUBROUTINE WHOLETHING;
BEGIN AREA := CTABLE[INDEX],[FF];
IF NOT B THEN
BEGIN IF AREA#0 AND AREA<PRT,[CF] THEN
BEGIN FORGETSPACE(AREA); AREA := 0 END;
CTABLE[INDEX] := 0 & AREA[CTF];
FTABLE[INDEX] := 0
END ELSE SETUPARRAYROW;
END;

%*****
INDEX := 0; B := USEDRA AND P(RRR).[31:1]; WHOLETHING;
INDEX := 1; B := USEDRA AND P(RRR).[30:1]; WHOLETHING;
P(XIT);
CON1:== @0100037777777777;
CON2:== @0107777777777777;
END;
PROCEDURE AUXILIARYMEMORYCASUALTYRECOVERY;
BEGIN COMMENT RECOVERS "LOST" AUXILIARY MEMORY
USED FOR DATA STORAGE FOR A JOB
WHICH TERMINATED ABNORMALLY;
REAL J, RSLT, IOD;
INTEGER I,
T,
INDEX;

ARRAY A[*];

LABEL LOOP,
EXIT,
DONE;

SUBROUTINE FIRSTAID;
BEGIN IF (A:=FTABLE[INDEX]) = (I:=0) THEN GO TO EXIT;
LOOP: IF I,[36:1] THEN GO TO EXIT;
IF (J := (NOT A[I,[37:6]]).[16:32])=0 THEN
BEGIN I := (I OR 31)+1; GO TO LOOP END;
STREAM(J; SI:=I,[43:5], NI:=P(DUP, 32, XCH, SUB));
BEGIN SI:=LOC J; SI:=SI+2;
SKIP 3 SB; SKIP S SB;
N(SKIP SB; IF SB THEN JUMP OUT;
TALLY:=TALLY+1);
J:=TALLY
END;
IF I,[36:7]#(I:=P(XCH)+1).[36:7] THEN GO TO LOOP;
IOD := T & 1[2:47:1] & (1+INDEX)[4:46:2] &
1[8:38:10] & 1[18:37:11];
IOREQUEST(NABS(IOD),IOD,[RSLT]&(16+INDEX)[12:42:6]);
RSLT := 0;
SLEEP([RSLT],IOMASK);
IF RSLT,[26:7] NEQ 0 THEN % AUXMEM ERROR
BEGIN
AUXDATA[P1MIX] := 0; GO TO DONE;
END;
IF (J:=M[T])>0 AND J,[CF]=P1MIX THEN

```

```

29415000
29416000
29417000
29418000
29419000
29420000
29421000
29422000
29423000
29424000
29425000
29426000
29427000
29428000
29429000
29430000
29431000
29432000
29433000
29434000
29435000
29436000
29500000
29501000
29502000
29503000
29504000
29505000
29507000
29509000
29510000
29511000
29512000
29513000
29514000
29515000
29516000
29517000
29518000
29520000
29521000
29522000
29523000
29524000
29525000
29526000
29527000
29528000
29529000
29530000
29531000
29531100
29531200
29531300
29531400
29531500
29531600
29531700
29531800
29532000

```

```

        BEGIN FORGETAUXILIARYSPACE(J,[FF], I & INDEX[36:47:1]); 29533000
            IF (AUXDATA[P1MIX] := *P(DUP)-J,[23:6]-1) = 0 29534000
                THEN GO TO DONE; 29535000
        END; 29536000
        I := I+J,[23:6]+1; 29537000
        GO TO LOOP; 29538000
EXIT:; 29539000
    END FIRTAID; 29540000
        IF AUXDATA[P1MIX]=0 THEN P(XIT); 29541000
        T := GETSPACE(10,0,5)+3; 29542000
        WAITSTORE(P1MIX); STOREDY(P1MIX):=0; 29543000
        FIRTAID; INDEX := 1; FIRTAID; 29544000
    DONE:; FORGETSPACE(T-1); 29546000
$ SET OMIT = NOT(DEBUGGING) OR OMIT 29547000
    IF AUXDATA[P1MIX]≠0 THEN DDT; 29547049
$ POP OMIT 29547050
    STOREDY(P1MIX):=1; 29547051
    END CASUALTY RECOVERY OF STORAGE THROUGH LINKS IN AUX MEM; 29548000
$ POP OMIT 29549000
COMMENT ERRORMESSER IS CALLED BY ERRORFIXER (IF OPTION 33 IS ON) TO 29549001
    TYPE OUT A PSEUDO-TERMINAL MESSAGE, IT DOES ABOUT THE SAME 30900000
    THING AS THE FIRST PART OF TERMINALMESSAGE; 30901000
PROCEDURE ERRORMESSER(TYPE); VALUE TYPE; REAL TYPE; 30902000
    BEGIN INTEGER S,ADR,BF,SA,N; 30903000
        NAME SD; 30904000
        LABEL L; 30905000
        BF := SPACE(10); 30906000
        SD=PRT[P1MIX,4]; 30907000
        NT1=SD[0]; 30908000
        ADR=M[PRT[P1MIX,8]],[CF]; 30909000
        FOR S+1 STEP 1 UNTIL NT1 DO 30910000
            IF (SA+SD[S],[18:15])>1023 AND SA≤ADR AND SD[S]>0 THEN 30911000
                IF M[SA=1],[18:15]+SA>ADR THEN GO L; 30912000
        S+0; 30913000
L: SD=[M[SPACE(TERMSGSZ)]]; 30914000
        ADR+ADR=SA; 30915000
        DISKWAIT(-(SD INX 0),TERMSGSZ,MESSAGETABLE[1],[22:26]); 30916000
        N+IF TYPE=1 THEN 11 ELSE IF TYPE=2 THEN 9 ELSE IF TYPE=4 THEN 30917000
            7 ELSE IF TYPE=8 THEN 13 ELSE 5; 30918000
        STREAM(M+[SD[N]],J+[JAR[P1MIX,0]],P1MIX,S,ADR,X+S≠0,BF); 30919000
        BEGIN SI+M; SI+SI+2; DS+6 CHR; BF+DI; DI+LOC M; SI+SI+1; 30920000
            DI+DI+7; DS+CHR; DI+BF; DS+M CHR; DS+8 LIT" BRANCH "; 30921000
            SI+J; SI+SI+1; DS+7 CHR; DS+LIT"/"; 30922000
            SI+SI+1; DS+7CHR; DS+LIT"="; SI+LOC P1MIX; 30923000
            DS+2DEC; BF+DI; DI+DI=2; DS+FILL; DI+BF; 30924000
            X(DS+5 LIT", S =" ; SI+LOC S; DS+4 DEC; DS+5 LIT", A =" ; 30925000
                DS+4 DEC; BF+DI; DI+DI=4; DS+3 FILL; 30926000
                DI+BF; DI+DI=13; DS+3 FILL); 30927000
            DI+BF; DS+ LIT"="; 30928000
        END; 30929000
        FORGETSPACE(SD); 30929500
        SPOUTER(BF,0,ERRORMSG); 30930000
    END ERRORMESSER; 30931000
PROCEDURE ERRORFIXER(TYPE); VALUE TYPE; INTEGER TYPE; 31000000
COMMENT LOOKS FOR RUN-TIME-ERROR ACTION LABELS IN ALGOL PROGRAMS, 31001000
    AND HANDLES THEM, RETURNING ONLY IF NO LABEL GIVEN; 31002000
BEGIN ARRAY AIT[*],PRTD[*]; 31003000
    NAME ADDR; 31004000
    REAL I, GOT, ADR=ADDR,LABLE; 31005000

```

```

CHECKSTACKSPACE;
    IF TYPE =2 THEN%OVRFLW
    IF JAR[P1MIX,2],[3:1] THEN
    IF(PRT[P1MIX,@51]AND @20)≠0 THEN
    BEGIN I+M[ADR+PRT[P1MIX,8] INX 0];
    STREAM(I+(I INX 0)&I[30:10:2],GOT+[GOT]);
    BEGIN SI+1;SI+SI-2;DI+DI+6;DS+2 CHR END;
    IF GOT,[45:3]=5 THEN M[ADR=3]+@7777777777777777;
    M[ADR=2]+@7777777777777777;
    PRT[P1MIX,@51],[45:2]+2;
    GO TO INITIATE;
    END;
PRTD ← PRTRW[P1MIX];
    WHILE (AIT+PRTD [AITNDX]),PBIT=0 DO
    MAKEPRESENT([PRTD [AITNDX] INX 0];
    I+AIT[0]+1;
    DO I+I-1 UNTIL((GOT+(ADDR+AIT[I]),OWNBIT AND (ADR,[CF]
    =TYPE)) OR(I≤1)); % LOOK FOR ENTRY
    IF GOT THEN % WILL REINITIATE THE GUY, SO SET HIM UP
    BEGIN IF (LABLE+M[ADR,MOM])≠0 THEN
    IF LABLE≠15 THEN
    IF LABLE,BLKCNT≤(PRTD[16]+(LABLE,MOM≠0))THEN
    BEGIN IF PRTD [CURBLKCNT]>LABLE,BLKCNT THEN
    BEGIN PRTD [CURBLKCNT]+LABLE,BLKCNT+1;
    ASR;
    END; IF(ADDR+LABLE,MOM)=0 THEN
    LABLE,MOM+ADDR+PRTD[10],MOM+2;
    ADDR+ADDR&ADR[33:33:15];
$ SET OMIT = PACKETS
    IF ERRORMSG THEN
$ POP OMIT
    ERRORMESSER(TYPE);
    IF PRTD[LABLE,[CF]],PBIT=0 THEN
    MAKEPRESENT([PRTD[LABLE,[CF]],[CF]]);
    DO UNTIL(*(ADDR+HUNT(ADDR+1)),[1:3]=4);
    ADDR [1]+M[PRTD [8] INX NOT 0];
    ADDR [2]+M[PRTD [8]]&0[10:10:2]&
    (LABLE)[18:18:15]&PRTD [(LABLE),[CF]][33:33:15];
    PRTD [8]+P(DUP,LOD)&(ADDR INX 2)[33:33:15];
    GO INITIATE;
    END; END;
END ERRORFIXER;
PROCEDURE SKIPFILE(U,D); VALUE U,D; REAL U,D;
    BEGIN REAL T1,T2,IOD,K;
    NAME N;
    LABEL L,EXIT;
    N := [T1];
    T2+IOMASK;
    IOD:=SPACE(2)&@1400[CTF]&TINU[U][3:3:5]&D[22:47:1];
    L: IOREQUEST(NABS(IOD)&@377[25:40:8],IOD,[N]&U[12:42:6]);
    K := 1-K;
    N[0] := 0;
    N := K INX [T1];
    WAITORSWAP(U,[N],[CF]);
    IF N[0],[27:1] THEN GO TO EXIT;
    IF PRTRW[P1MIX],[PSF] ≠ 0 THEN
    BEGIN IF (NT3:=PRTRW[P1MIX],[PSF])=1 THEN
    TERMINALMESSAGE(PRTRW[P1MIX],[FF]);
    IF NT3=2 THEN STOPM ELSE SWAP(FORCESWAP,1);
    GO TO L;

```

```

31005010
31005050
31005100
31005200
31005300
31005310
31005320
31005330
31005350
31005400
31005500
31005600
31006100
31007000
31008000
31009000
31010000
31011000
31012000
31013000
31013050
31013100
31014000
31015000
31016000
31017000
31017100
31017200
31017209
31017210
31017211
31017220
31017300
31017400
31018000
31019000
31020000
31021000
31022000
31023000
31024000
31025000
31100000
31101000
31102000
31102100
31102200
31102300
31103000
31104000
31105000
31106000
31107000
31108000
31109000
31110000
31111000
31112000
31113000
31114000

```

END;	31115000
IF MEMROW[P1MIX],[CF]<FENCE THEN GO TO L;	31115100
IF ELAPSEDLIMIT[P1MIX] GTR IOTIME[P1MIX]	%R5931116000
THEN GO TO L;	%R5931116100
FOR NT3 := SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO	31117000
IF ACTIVE[NT3] GTR 1 THEN	%DS31118000
BEGIN SWAP(TIMEND,1); GO TO L; END;	31119000
ELAPSEDLIMIT[P1MIX]:=*P(DUP)+64;	31120000
GO TO L;	31121000
EXIT:BLASTQ(U);	31122000
FORGETSPACE(IOD);	31123000
END;	31124000
REAL PROCEDURE FINDOUTPUT(MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE,KIND);%	37000000
VALUE MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE;%	37001000
REAL MID,FID,TYPE,FORMS,REEL,CDATE,CYCLE,KIND;%	37002000
BEGIN INTEGER GOTL,GOTT,GOTB,GOTP,GOTC;	37003000
REAL U;	37003100
LABEL EXIT,SW,ON,OWT,AROUND,OUKID,X,ROUND,CLAIMT,THERE,SOMEWHERE;	37004000
% SET OMIT = NOT(PACKETS)	37004199
REAL FREEF; LABEL FREEL; % FILE TO BE PRINTED ALONE	37004200
% POP OMIT	37004201
LABEL W1,W2,W3,W4,W5,W6,W7;	%P 37005000
DEFINE DSED=(TERMSET(P1MIX));%	37006000
LABEL CP,MT,SU,PP,CKFM,DOITOVER;	%P 37007000
DEFINE PNTOG=(TYPE=0 OR TYPE GEQ 20);%	37007100
SWITCH TYPESW=CP,ROUND,MT,SU,ROUND,SU,ROUND,PP,PP,MT;	%P 37008000
REAL SUBROUTINE PRINTER;%	37009000
BEGIN IF LABELTABLE[20]=0 THEN BEGIN U+20; P(1) END ELSE%	37010000
IF LABELTABLE[21]=0 THEN BEGIN U+21; P(1) END ELSE P(0);%	37011000
PRINTER+GOTL+P;%	37012000
END PRINTER;%	37013000
REAL SUBROUTINE PTPUNCH;%	37014000
BEGIN IF LABELTABLE[26]=0 THEN BEGIN U+26; P(1) END ELSE%	37015000
IF LABELTABLE[29]=0 THEN BEGIN U+29; P(1) END ELSE P(0);%	37016000
PTPUNCH+GOTP+P;%	37017000
END PTPUNCH;%	37018000
REAL T1,T2,T3;%	37019000
REAL SUBROUTINE PUNCH;%	37019100
BEGIN IF LABELTABLE[22]=0 THEN BEGIN U+22;P(1) END ELSE P(0);	37019200
PUNCH+GOTC+P;	37019300
END PUNCH;	37019400
REAL SUBROUTINE MAGTAPE;%	37020000
BEGIN IF NOT(GOTL OR GOTB OR GOTC) THEN%	37021000
BEGIN IF T1#0 THEN%	37022000
BEGIN FOR U+0 STEP 1 UNTIL 15 DO%	37023000
IF (MULTITABLE[U] EQV T1)#NOT 0 THEN%	37024000
IF LABELTABLE[U]<0 THEN%	37025000
IF RDCTABLE[U],[8:16]=P1MIX THEN%	37026000
IF (T3+PRNTABLE[U])<0 THEN%	37027000
IF M[M[T3],[15:15]=3] INX 5],[41:1] THEN%	37028000
BEGIN P(1); GO OWT END;%	37029000
END;%	37030000
FOR U+0 STEP 1 UNTIL 15 DO%	37031000
IF LABELTABLE[U]=0 AND PRNTABLE[U],[1:1]	%R9137032000
THEN BEGIN P(1); GO OWT; END;	%R9137032100
END;%	37033000
P(0);%	37034000
OWT; MAGTAPE+GOTT+P;%	37035000
END MAGTAPE;%	37036000
SUBROUTINE BADFM; %BUILD AND SPOUT BAD FM MESSAGE %	37036100

BEGIN	%RHR	37036200
T1:=SPACE(10);		37036300
STREAM(A+TINU[U],MX+P1MIX,T1);	%RHR	37036400
BEGIN DS+19 LIT "INVALID INPUT UNIT ";	%RHR	37036500
SI+LOC MX; DS+2 DEC; DS+2 LIT"FM";	%RHR	37036600
SI+LOC A; SI+SI+5; DS+3 CHR;	%RHR	37036800
DS+LIT "+"; DI+DI-8; DS+FILL;	%RHR	37036900
END; SPOUT(T1);	%RHR	37037000
LABELTABLE[U]+@114; READY+READY AND (U+NOT TWO(U));		37037100
RRRMECH+RRRMECH AND U; SAVEWORD+SAVEWORD AND U;	%RHR	37037200
END BADFM SUBROUTIN;	%RHR	37037300
REAL SUBROUTINE BKUPTAPE;%		37038000
BEGIN IF NOT(GOTL OR GOTC) THEN		37039000
FOR U+0 STEP 1 UNTIL 15 DO%		37040000
IF (LABELTABLE[U] EQV T3)=NOT 0 THEN%		37041000
IF (MULTITABLE[U] EQV T2)=NOT 0 THEN%		37042000
BEGIN P(1); GO AROUND END;%		37043000
P(0);%		37044000
AROUND;  BKUPTAPE+GOTB+P;%		37045000
END BKUPTAPE;%		37046000
\$ SET OMIT = NOT(PACKETS)		37046004
FREEF+TYPE.[1;1]; TYPE+ABS(TYPE);		37046005
\$ POP OMIT		37046006
IF TYPE>1 AND TYPE#4 AND TYPE#6 AND TYPE<15 THEN GO SOMEWHERE;		37046020
ROUND;  IF TYPE=1 OR TYPE=4 OR (TYPE>16 AND TYPE<19) THEN		37046040
IF PRINTER THEN BEGIN KIND+1; GO CKFM END;	%P	37046060
IF TYPE=0 OR (TYPE>20 AND TYPE) THEN		37046070
IF PUNCH THEN BEGIN KIND+6; GO CKFM END;		37046075
IF TYPE=4 OR TYPE=6 OR TYPE=16 OR TYPE=18 OR		37046080
(TYPE GEQ 20 AND NOT TYPE,[46;1]) THEN		37046090
BEGIN T1+0; T2+IF TYPE GEQ 20 THEN "PUTMCP " ELSE "PBTMCP ";		37046100
T3+@122212342546447;		37046110
IF BKUPTAPE THEN GO THERE;	%P	37046120
IF MAGTAPE THEN	%P	37046140
CLAIMT;  BEGIN MULTITABLE[U]+T2; LABELTABLE[U]+T3;	%P	37046160
RDCTABLE[U],[8;6]+P1MIX;		37046162
RRRMECH+TWO(U) OR RRRMECH;	%P	37046170
IF REEL=0 THEN REEL+1;		37046175
T1 := SPACE(10)+2;		37046180
STREAM(U:=TINU[U],N:=PRNTABLE[U],[30;18],		37046190
A+REEL,B+DATE,C+CYCLE,D+0,PN+TYPE GEQ 20,		37046192
T+T1-2);		37046194
BEGIN DS+12LIT" NEW PBT ON"; SI+LOC U; SI+SI+5;	%P	37046200
PN(D+DI; DI+DI-6; DS+2LIT"UT"; DI+D);		37046205
DS+3 CHR;DS+25LIT"+ LABEL OPBTMCP OBACK=UP";%P		37046210
PN(D+DI; DI+DI-14; DS+2LIT"UT"; DI+D);		37046212
SI := LOC A; DS := 3 DEC;		37046215
SI+SI+3; DS+5 CHR; SI+SI+6;DS+2 CHR;		37046217
15(DS:=2 LIT"0");DI:=DI-11;SI:=LOC N;		37046220
DS:=5 DEC;		37046221
END;	%P	37046240
P(WAITIO(T1&8[8;38;10]&5[21;45;3],0,U),DEL);	%P	37046260
SPOUT(T1-2);		37046270
T1.[1;11]:=@1737;		37046280
P(WAITIO([T1],0,U),DEL);	%P	37046300
THERE;  LABELTABLE[U],[1;5]+@20; KIND+7; GO EXIT	%P	37046320
END; END;	%P	37046340
IF (TYPE GEQ 15 AND TYPE LEQ 18) OR TYPE GEQ 22 THEN		37046350
BEGIN		37046360
\$ SET OMIT = NOT(PACKETS)		37046369



```

IF (T1:=PSEUDOMIX[P1MIX])≠0 AND TYPE<22 AND NOT FREEF THEN 37046370
BEGIN 37046380
  T1:=T1-32; 37046390
  T2:=PACKETPBD[T1]; 37046400
  T3:=CIDTABLE[T1,6],[6:24]; 37046410
  IF T2=0 OR T3=0 OR (T2+10)>1000 THEN GO FREEF; 37046420
  PACKETPBD[T1]:=T2+10; 37046430
END ELSE 37046440
$ POP OMIT 37046441
BEGIN 37046450
$ SET OMIT = NOT(PACKETS) 37046459
FREEF; 37046460
$ POP OMIT 37046461
  T3:=NEXTCDNUM(1); 37046470
  T2:=001; 37046480
END; 37046490
KIND:=12; 37046500
STREAM(T3,T2,D:=T1:=U:=SPACE(30)); 37046520
BEGIN 37046530
  DS+8 LIT"0@+1,013";DS+24 LIT"0"; 37046540
  DS:=7 LIT"8400000";DS:=10 LIT"0"; 37046560
  SI:=LOC T3;SI:=SI+4; DS:=4 CHR; 37046580
  SI:=LOC T2; DS:=3 DEC; 37046590
  46(DS+4 LIT"0"); 37046600
END; M[T1+1]*M[T1+8]* PBDROWSZ+1; 37046620
$ SET OMIT = NOT(SHAREDISK) 37046624
M[T1+4],[4:2]:=SYSNO; 37046625
$ POP OMIT 37046626
  M[T1+5]*MID&(TYPE GEQ 22)[3:47:1]; % PNCH BK UP TOG 37046630
  GO EXIT 37046640
END; 37046660
W3: FILEMESS("#,.,.,."&(IF TYPE=6 OR TYPE=20 THEN " " ELSE 37046680
  (IF PNTOG THEN "CP" ELSE "LP"))[12:36:12] 37046690
  &(IF TYPE<2 THEN " " ELSE IF TYPE GEQ 20 THEN 37046700
  "PUT" ELSE "PBT")[30:30:18], 37046710
  (IF TYPE<2 THEN "RQD " ELSE " MT RQD"),MID,FID, 37046720
  REEL,CDATE,CYCLE); 37046740
  REPLY[P1MIX]:= -VOK&VWY[36:42:6]&VOU[30:42:6]; 37046742
  IF [MEM[P1MIX,MLINK1]],[CF]≥FENCE THEN SWAP(WAITSWAP,1); 37046750
  COMPLEXSLEEP(((IF (TYPE≠6 AND TYPE≠20) THEN IF PNTOG THEN 37046760
  PUNCH ELSE PRINTER ELSE 0) OR REPLY[P1MIX] 37046770
  >0 OR(IF TYPE>1 THEN BKUPTAPE OR MAGTAPE ELSE 0) OR 37046780
  DSED)); IF DSED THEN GO INITIATE; %P 37046800
  IF NOT(GOTB OR GOTT OR GOTL OR GOTC) THEN 37046820
  BEGIN 37046838
  IF NOT WHYSLEEP(VOK&VWY[36:42:6]&VOU[30:42:6]) THEN GO TO W3; 37046840
  IF REPLY[P1MIX]=VOK THEN GO TO W3; 37046850
  IF PNTOG THEN BEGIN U+REPLY[P1MIX],[FF]; GO CP END; 37046855
  TYPE+IF (U+REPLY[P1MIX],[FF])=1 THEN 4 ELSE %P 37046860
  IF U=2 THEN 1 ELSE IF U=3 THEN 6 ELSE 15; 37046880
  REPLY[P1MIX]+0; GO ROUND; %P 37046900
END; REPLY[P1MIX]+0; %P 37046920
  IF GOTB THEN GO THERE ELSE IF GOTT THEN GO CLAIMT ELSE 37046940
  IF GOTC THEN KIND+6 ELSE KIND+1; 37046950
CKFM: IF FORMS THEN %P 37046960
BEGIN LABELTABLE[U]+FID; MULTITABLE[U]+MID; %P 37046980
  RDCTABLE[U],[8:6]+P1MIX; 37046990
W7: FILEMESS("#,.,., FM"&TINU[U][12:30:18],"RQD ", %P 37047000
  MID,FID,REEL,CDATE,CYCLE); 37047100
  REPLY[P1MIX]+ -VWY&VOK[36:42:6]&VOU[30:42:6]&VFM[24:42:6]; 37047105

```

```

IF [MEM[P1MIX,MLINK1]],[CF]≥FENCE THEN 37047110
  SWAP(WAITSWAP,1) ELSE 37047120
COMPLEXSLEEP((REPLY[P1MIX]>0 OR DSED)); %P 37047200
IF REPLY[P1MIX]=VOK THEN GO EXIT; 37047300
IF NOT WHYSLEEP(VWY&VOK[36:42:6]& 37047400
VOU[30:42:6]&VFM[24:42:6]) THEN GO TO W7; 37047410
KIND+LABELTABLE[U]+MULTITABLE[U]+GOTL+RDCTABLE[U]+U+0; 37047500
IF NOT DSED THEN IF REPLY[P1MIX],[CF]=VFM THEN %RHR 37047600
IF(U+REPLY[P1MIX],[FF])≠20 AND U≠21 THEN %RHR 37047605
BEGIN BADFM; GO ROUND END ELSE %RHR 37047610
BEGIN LABELTABLE[U]+FID; RDCTABLE[U],[8:6]+P1MIX; 37047625
MULTITABLE[U]+MID; KIND+UNIT[U],[1:4]; %RHR 37047650
GO EXIT; %RHR 37047660
END ELSE GO OUKID; %RHR 37047670
END; GO X; %P 37047700
SOMEWHERE; IF FORMS THEN %P 37047800
W1: BEGIN REPLY[P1MIX]+VWY&VFM[36:42:6];% 37048000
FILEMESS("#FM RQD",0,MID,FID,REEL,CDATE,CYCLE);% 37049000
IF [MEM[P1MIX,MLINK1]]≥FENCE THEN SWAP(WAITSWAP,1) ELSE 37049500
COMPLEXSLEEP((REPLY[P1MIX]>0) OR DSED);% 37050000
IF DSED THEN GO TO INITIATE;% 37051000
IF NOT WHYSLEEP(VWY&VFM[36:42:6]) THEN GO TO W1;% 37052000
U+REPLY[P1MIX],[18:15]; REPLY[P1MIX]+0;% 37053000
IF NOT DSED THEN 37053100
IF U LSS 16 THEN 37053200
IF PRNTABLE[U],[1:1] THEN ELSE GO TO SOMEWHERE; 37053300
GO TO X;% 37054000
END;% 37055000
SW: GO TO TYPESW[TYPE];% 37056000
% 37057000
CP: TYPE+IF U=1 THEN 21 ELSE IF U=3 THEN 20 ELSE 37058000
IF U=5 THEN 0 ELSE 22; REPLY[P1MIX]+0; GO ROUND; 37059000
% 37084000
PP: IF NOT PTPUNCH THEN% 37085000
W4: BEGIN FILEMESS("#PP RQD",0,MID,FID,REEL,CDATE,CYCLE);% 37086000
IF OUTWAIT(PTPUNCH) THEN GO TO W4; 37087000
IF NOT PTPUNCH THEN GO TO W4; 37088000
END;% 37093000
GO TO X;% 37094000
% 37095000
SU: T1+FID,[6:18];% 37096000
FOR U=0 STEP 1 UNTIL 31 DO% 37097000
IF TINU[U],[30:18]=T1 THEN GO ON;% 37098000
GO TO MT;% 37099000
ON: IF LABELTABLE[U]≠0 THEN% 37100000
W5: BEGIN FILEMESS("#,.., "&T1[12:30:18],"RQD ",% 37101000
MID,FID,REEL,CDATE,CYCLE);% 37102000
IF OUTWAIT(LABELTABLE[U]=0) THEN GO TO W5; 37103000
IF LABELTABLE[U]≠0 THEN GO TO W5; 37104000
END;% 37109000
GO TO X;% 37110000
% 37111000
MT: T1+MID;% 37112000
IF NOT MAGTAPE THEN% 37113000
W6: BEGIN FILEMESS("#MT RQD",0,MID,FID,REEL,CDATE,CYCLE);% 37114000
IF OUTWAIT(MAGTAPE) THEN GO TO W6; 37115000
IF DSED THEN 37115100
BEGIN 37115200
U+1; 37115300
GO TO EXIT; 37115400

```

```

                END;
                IF NOT MAGTAPE THEN GO TO W6;
            END;
            IF (T1+PRNTABLE[U],[15:15])#0 THEN%
            BEGIN FILECLOSE(T1&3[18:33:15]);%
                M[M[T1=3] INX 5],[39:4]+1;%
            END;%
%
X:  KIND=UNIT[U],[1:4];%
    LABELTABLE[U]+=FID; MULTITABLE[U]+MID;%
    RDCTABLE[U]+P(DUP,LOD)&REEL[14:38:10]&CDATE[24:31:17]
        &CYCLE[41:41:7]&P1MIX[8:42:6];
EXIT:  FINDOUTPUT+U;%
END FINDOUTPUT;%
REAL PROCEDURE FINDINPUT(MID,FID,REEL,CDATE,CYCLE,COBOL,UL,OF,MODE,FN);
    VALUE  MID,FID,REEL,CDATE,CYCLE,COBOL,  OF,MODE,FN;%
    REAL  MID,FID,REEL,CDATE,CYCLE,COBOL,UL,OF,MODE,FN;%
BEGIN REAL T1,T2,U,LO,HI,FIRST,IL;
    REAL A=COBOL;
    INTEGER S,COUNT;
    INTEGER USASI=IL;
    ARRAY FPB=LO[*];
    LABEL LOOK,SEE,SRCHOUT;
    LABEL START,WHY,EXIT,X,Y,READALABEL,REW,EXIT;
    LABEL ONN,DUN,FAIL;
    DEFINE UNLABELED = UL#;

    DEFINE DSED=(TERMSET(P1MIX))#;

    SUBROUTINE CHECKTERMIX;
    BEGIN
        IF DSED THEN
            BEGIN
                IF (JAR[P1MIX,0] EQV "LIBMAIN")=NOT 0 AND
                    (JAR[P1MIX,1] EQV "DISK ")=NOT 0 THEN
                    BEGIN
                        U+1;
                        GO TO EXIT;
                    END ELSE GO TO INITIATE;
                END;
            END;
        % CHECKTERMIX

    REAL SUBROUTINE SEARCH;%
    BEGIN COUNT=0; IF NOT DSED THEN
% SET OMIT = NOT(PACKETS)
    BEGIN IF(LO#(HI#PSEUDOMIX[P1MIX])) NEQ 0 THEN
        BEGIN IF S GEQ 0 THEN
            IF(LABELTABLE[LO] EQV (-@14)=NOT 0) THEN
                COMPLEXSLEEP((LABELTABLE[LO] EQV (-@14)) NEQ NOT 0);
% POP OMIT
% SET OMIT = PACKETS
        BEGIN IF (LO#JAR[P1MIX,6],[2:6]) GEQ 32 THEN HI#=LO ELSE
            BEGIN HI#=LO#=0; GO TO FAIL; END;
% POP OMIT
        LOOK:  FOR U+LO STEP 1 UNTIL HI DO%
            IF (LABELTABLE[U] EQV FID)=NOT 0 THEN%
            IF (MULTITABLE[U] EQV MID)=NOT 0 THEN%
            IF ((T1+RDCTABLE[U]),[14:10]=REEL) OR (REEL=0) THEN%
            IF (T1,[24:17]=CDATE) OR (CDATE=0) THEN%
            IF (T1,[41:7]=CYCLE) OR (CYCLE=0) THEN%

```

```

37115500
37116000
37121000
37122000
37123000
37124000
37125000
37172000
37173000
37174000
37174100
37174200
37175000
37176000
37177000
37178000
37179000
37180000
37180100
37180200
37180300
37180400
37180500
37180600
37180650
37180700
37180990
37181000
37185300
37185310
37185320
37185330
37185340
37185350
37185360
37185370
37185380
37185390
37185400
37185410
37185420
37185990
37186000
37187000
37187099
37187100
37187110
37187120
37187130
37187131
37187999
37188000
37188100
37188101
37189000
37190000
37191000
37192000
37193000
37194000

```

```

BEGIN
$ SET OMIT = NOT(PACKETS)
    IF LO#HI AND LO=32 THEN
    IF PACKETACT[U=32]#0 THEN GO TO FAIL;
$ POP OMIT
    COUNT:=COUNT+1; P(U,XCH);
END;
FAIL:
$ SET OMIT = NOT(PACKETS)
END;
$ POP OMIT
    IF LO = HI THEN IF COUNT = 1 THEN GO SEE ELSE
    IF LO#0 THEN IF (LO:=JAREP1MIX,6J,[2:6J])=23 OR LO#24
    THEN HI:=LO ELSE GO TO ONN ELSE
ONN: BEGIN LO:=32; HI:=35; END ELSE
    IF LO=32 THEN BEGIN LO:=23; HI:=24; END ELSE
    IF LO=23 THEN BEGIN LO:= 0; HI:=15; END ELSE GO TO DUN;
    GO TO LOOK;
DUN: IF CYCLE,[1:1] THEN % PBT
    BEGIN
    IF COUNT=0 THEN IF FID,[1:5]<3 THEN
    BEGIN FID,[1:5]+FID,[1:5]+1;
    LO+0; HI+15; GO LOOK;
    END ELSE FID,[1:5]+1;
    GO SRCHOUT;
    END;
    IF COUNT=0 THEN
    IF MID#0 THEN%
    IF NOT CDATE,[1:1] THEN % NOT LIBMAIN/DISK
    FOR U+0 STEP 1 UNTIL 15 DO%
    IF (MULTITABLE[U] EQV MID)=NOT 0 THEN%
    IF (RDCTABLE[U],[24:17]=CDATE) OR (CDATE=0) THEN
    IF LABELTABLE[U]>0 THEN%
    BEGIN COUNT+COUNT+1;
    P(U,XCH);
    END ELSE%
    IF RDCTABLE[U],[8:6]=P1MIX THEN%
    IF (T1+M[M[PRNTABLE[U],[15:15]-3] INX 5]),[41:1] THEN
    IF T1,[43:1] OR T1,[40:1]=0 THEN%
    BEGIN COUNT+COUNT+1; P(U,XCH) END;
SEE:
    END;
    SRCHOUT:
    SEARCH+S+COUNT>0;
    END SEARCH;%
    REAL SUBROUTINE RESEARCH;
    BEGIN
$ SET OMIT = NOT PACKETS
    S:=-2;
$ POP OMIT
    P(SEARCH);
    DO P(DEL) UNTIL (COUNT:=COUNT-1) LSS 0;
    RESEARCH+S;
    END RESEARCH;
    REAL SUBROUTINE REED;%
    BEGIN IF (HI+WAITIO(T1,LO&@377[18:33:15],U) AND @367)#0 THEN
    IF (HI AND NOT LO)#0 THEN
    BEGIN BLASTQ(U); SETNOTINUSE(U,0); STOPTIMING(FN,1023);

```

```

37195000
37195009
37195010
37195020
37195021
37195030
37195040
37195050
37195099
37195100
37195101
37195200
37195250
37195280
37195300
37195350
37195400
37195450
37195500
37195550
37195600
37195650
37195700
37195750
37195800
37195850
37196200
37197000
37197500
37198000
37199000
37199100
37200000
37201000
37202000
37203000
37204000
37205000
37206000
37207000
37207500
37208000
37208500
37209000
37210000
37210090
37210100
37210150
37210170
37210175
37210180
37210200
37210250
37210300
37210400
37210990
37211000
37212000
37213000
37214000

```

FILEMESS("PARITY ", "ON ... "&TINU[U][24:30:18], %	37215000
MID, FID, REEL, CDATE, CYCLE); %	37216000
END; %	37217000
IF DSED THEN	37218000
BEGIN	37218100
SETNOTINUSE(U, 0);	37218200
STOPTIMING(FN, 1023);	37218300
CHECKTERMIX;	37218400
END;	37219000
REED=HI; %	37220000
END REED; %	37221000
SUBROUTINE SEARCHCOM; % FILE SEARCH FOR COM 30	37221090
BEGIN P(DEL);	37221100
IF NOT SEARCH THEN U:=1 ELSE	37221120
IF COUNT=1 THEN U:=P ELSE	37221140
BEGIN	37221160
S:=COUNT; T1:=0;	37221180
COUNT:=IF COUNT>8 THEN 8 ELSE COUNT;	37221200
WHILE (COUNT:=COUNT-1) GEQ 0 DO	37221220
BEGIN U:=P;	37221240
IF T1 THEN	37221260
BEGIN	37221280
T1:=0; M[A],[30:18]:=TINU[U],[30:18];	37221300
A:=A+1;	37221320
END ELSE	37221340
BEGIN	37221360
T1:=1; M[A],[12:18]:=TINU[U],[30:18];	37221380
END;	37221400
END;	37221420
U:=-S;	37221440
END;	37221460
GO EXIT;	37221480
END;	37221500
START; %	37221520
IF UL<0 THEN SEARCHCOM ELSE	37221990
IF UL THEN GO TO WHY ELSE %	37222000
IF NOT SEARCH THEN %	37222100
WHY: BEGIN FILEMESS("#NO FIL", 0, MID, FID, REEL, CDATE, CYCLE); %	37222500
FIRST:=VOK&VWY[36:42:6]&VUL[30:42:6]&VIL[24:42:6];	37223000
IF COBOL THEN	37224000
FIRST:=FIRST&(VOF×OF)[18:42:6]&(VFR×UL)[12:42:6];	37225000
REPLY[P1MIX]←FIRST&1[2:47:1];	37225050
IF[MEM[P1MIX,MLINK1]], [CF]≥FENCE THEN	37225100
SWAP(WAITSWAP, 1);	37226000
COMPLEXSLEEP(RESEARCH OR (REPLY[P1MIX]>0) OR DSED);	37226100
CHECKTERMIX;	37226200
IF S THEN S←SEARCH ELSE	37227000
BEGIN IF NOT WHYSLEEP(FIRST) THEN GO TO WHY;	37228000
IF (T2:=(T1:=REPLY[P1MIX]), [FF]) GTR 64 THEN % IL	37229000
BEGIN STREAM(T2); % MID/FID	37229500
BEGIN SI:=T2;	37230000
LL: SI:=SI+1; IF SC#"L" THEN GO TO LL;	37230250
SI:=SI+1; T2:=SI;	37230500
END;	37230750
T2:=P;	37231000
NAMEID(HI, T2); MID:=HI; NAMEID(HI, T2);	37231250
NAMEID(HI, T2); FID:=HI;	37231500
FORGETSPACE(T1, [FF]-1);	37232000
	37232250
	37232500

GO TO Y;	37232750
END;	37233000
IF T1=VOK THEN GO TO Y;	% OK 37233250
IF NOT (IL:=T1,[CF]=VIL) THEN	% OF, FR 37233500
BEGIN U:=1;	37233750
REPLY[P1MIX]:=0;	37234000
GO TO EXIT;	37234250
END;	37234500
UNLABELED←=LABELTABLE[U+T1,[18:15]]=@314;%	37235000
P(U);	37235100
COUNT:=1;	37235250
IF LABELTABLE[U]=0 THEN	37235500
BEGIN MULTITABLE[U]:=MID;	37235750
LABELTABLE[U]:=FID;	37236000
END ELSE	37236250
BEGIN MID:=MULTITABLE[U],[6:42];	37236500
FID:=LABELTABLE[U],[6:42];	37236750
END;	37237000
END;	37238000
REPLY[P1MIX]+0;%	37239000
END;%	37240000
IF COUNT>1 THEN	37240100
SXIT: BEGIN FILEMESS("#DUP ", "FIL ", "MID,FID,REEL,CDATE,CYCLE);	37240200
WHILE (COUNT+COUNT-1)≥0 DO	37240300
BEGIN IF (U+P)<16 THEN IF MID≠0 THEN	37240400
IF (T1+PRNTABLE[U],[15:15])≠0 THEN	37240500
FILECLOSE(T1&@12[18:33:15]);	37240600
STREAM(X:=[TINU[U]]; DI:=SI:=SPACE(10));	37240700
BEGIN SI+X; SI+SI+5; DS+8 LIT " DUP ON ";	37240800
DS+3 CHR; DS+LIT " ";	37240900
X+DI;	37240910
END;	37241000
T1+P;	37241010
IF U≥32 THEN IF CIDROW[U -32]≠0 THEN	37241020
STREAM(DK+CIDTABLE[U -32,2],T1);	37241030
BEGIN DI+DI-1; DS+6 LIT " DECK ";	37241040
SI+LOC DK; SI+SI+1; DS+7 CHR;	37241050
END;	37241060
END;	37241200
REPLY[P1MIX]i= -VWY&VOK[36:42:6]&VIL[30:42:6];	37241300
IF[MEM[P1MIX,MLINK1]], [CF]≥FENCE THEN	37241310
SWAP(WAITSWAP,1) ELSE	37241320
COMPLEXSLEEP(DSED OR (REPLY[P1MIX]>0));	37241400
CHECKTERMIX;	37241500
IF (T1+REPLY[P1MIX]), [33:15]=VIL THEN	37241510
BEGIN REPLY[P1MIX]+0;	37241520
IF T1,[FF] > 36 THEN GO SXIT;	37241525
P(T1,[18:15]);	37241530
GO TO X;	37241540
END;	37241550
IF NOT WHYSLEEP(VWY&VOK[36:42:6]&VIL[30:42:6]) THEN	37241600
BEGIN SI:=SEARCH; GO TO SXIT; END;	37241610
REPLY[P1MIX]+0; GO TO START;	37241700
Y:	37241800
END;	37241810
X:	37241900
LABELTABLE[U+P],[1:5]+@20;	37241910
RDCTABLE[U],[8:6]+P1MIX;	37242000
IF NOT UNLABELED THEN	37242100
BEGIN FPB:=PRT[P1MIX,3];	37242200
FPB[FN]:=MID;	

```

        FPB[FN+1]:=FID;
    END;
    IF U LSS 16 THEN
    IF MID#0 THEN
    BEGIN IF (T1+PRNTABLE[U],[15:15])#0 THEN%
        BEGIN FILECLOSE(T1&3[18:33:15]);%
            M[M[T1-3] INX 5],[39:4]+1;%
        END;%
    %
        RRRMECH+TWO(U) OR RRRMECH; STARTIMING(FN,U);
        IF UNLABELED OR IL OR CYCLE,[1:1] THEN GO EXIT;
        T1 := SPACE(11)&10[8:38:10]&MODE[21:47:1]
            &3[23:46:2];%
        LO+@40; FIRST+1;%
    READALABEL: IF REED # 0 THEN IF FIRST THEN%
    REW: BEGIN FIRST+WAITIO(@4200000000,0,U); GO READALABEL END ELSE
        BEGIN SETNOTINUSE(U,1); FORGETSPACE(T1,[33:15]);
            STOPTIMING(FN,1023); GO TO START END;
        STREAM(Y:=0;X:=0,T1);
            BEGIN DI:=LOC X; DS:=8 LIT "VOL1HDR1";
                SI:=T1; DI:=DI-8;
                IF 4 SC=DC THEN TALLY:=1 ELSE
                    BEGIN SI:=T1; IF 4 SC=DC THEN TALLY:=2; END;
                Y:=TALLY;
            END;
            IF (USASI:=P)>0 THEN USASITAPE(T1,[CF],USASI,2,U,0);
            STREAM(M+0,F+0,R+0,D+0,C+0;S+T1 INX 1);%
            BEGIN SI+S; DI+LOC M; DS+2 WDS; DS+3 OCT;%
                DS+5 OCT; DS+2 OCT;%
            END;%
            IF (P=CYCLE OR CYCLE=0) AND (P(XCH)=CDATE OR CDATE=0) AND%
                (P(XCH)=REEL OR REEL=0)AND ((P(XCH) EQV FID)#NOT 0) AND%
                ((P(XCH) EQV MID)#NOT 0) THEN%
            BEGIN FORGETSPACE(T1,[33:15]); T1+@340000005;%
                LO+0;T1+REED; GO TO EXIT;%
            END;%
            IF FIRST THEN GO REW;%
            LO:=@60; DO UNTIL (FIRST:=REED),[42:1]; DO UNTIL REED,[42:1];
            IF USASI>0 THEN DO UNTIL REED,[42:1] ELSE FIRST:=REED;
            LO+@40; GO READALABEL;
        END;%
    EXIT: FINDINPUT+U;%
    END FINDINPUT;%
    PROCEDURE STARTIMING(FN,U); VALUE FN,U; REAL FN,U;%
    BEGIN ARRAY FPB[*]; INTEGER I,J;%
        FPB+PRT[P1MIX,3];%
        IF U<32 THEN
    BEGIN IF FPB[FN+4]#0 THEN
        BEGIN IF (I+FPB[FN+3],[36:6])#0 THEN%
            IF I#U+1 THEN IF (I+FPB,[8:10])#(1023-ETRLNG) THEN%
                BEGIN J+GETSPACE(I+ETRLNG,2,1)+2;%
                    % SET OMIT = SHAREDISK
                        MOVE(I,FPB,J);%
                    % POP OMIT
                    % SET OMIT = NOT SHAREDISK
                        MOVE(I+1,FPB INX NOT 0,J-1);%
                    % POP OMIT
                        MOVE(ETRLNG,[FPB[FN]],J+1);%
                        FORGETSPACE(FPB,[33:15]);%
                        FPBD[P1MIX] +

```

```

37242300
37242400
37242600
37242800
37243000
37244000
37245000
37246000
37247000
37248000
37248500
37249000
37250000
37251000
37252000
37253000
37254000
37255000
37255100
37255200
37255300
37255400
37255500
37255700
37255800
37255900
37256000
37257000
37258000
37259000
37260000
37261000
37262000
37263000
37264000
37265000
37266000
37267000
37267050
37267100
37268000
37269000
37270000
37271000
37272000
37273000
37273100
37274000
37275000
37276000
37277000
37277999
37278000
37278001
37278099
37278100
37278101
37278200
37279000
37279100

```

```

PRT[P1MIX,3]+FPB+[M[J]]+(I+ETRLNG)[8:38:10];% 37280000
FPB[FN+4]+0; FPB[FN+3],[24:12]+0;% 37281000
END;% 37282000
FPB[FN+4]+FPB[FN+4]-CLOCK+P(RTR);% 37283000
FPB[FN+3],[36:6]+U+1;% 37284000
IF U LSS 16 THEN % RDC & PRN LOG ENTRIES 37284100
BEGIN 37284110
STREAM(R:=RDCTABLE[U],[14:10],D:=RDCTABLE[U],[24:17], 37284120
C:=RDCTABLE[U],[41:7],T:=FPB[FN+2]); 37284130
BEGIN SI:=LOC R; DS:=3DEC; DS:=5DEC; DS:=DEC; END; 37284140
FPB[FN+3],[6:17]:=PRNTABLE[U],[31:17]; 37284150
END; 37284310
END END ELSE 37285000
BEGIN IF (I+FPB[FN+4])<=0 THEN 37285100
BEGIN FPB[FN+4]+I+CLOCK+P(RTR); I+FPB[FN+3],[36:6]-1; 37285200
FPB[FN+3],[24:12]+P(DUP),[24:12]+(J+TINU[I],[18:12]); 37285300
IF I<16 THEN 37285305
IF J>0 THEN FILEMESS("# IO"&TINU[I] 37285310
[12:30:18],"ERRORS:"&FPB[FN],FPB[FN+1],J,0,0); 37285320
TINU[I],[18:12]+0; 37285400
END END END TIMING; 37285500
REAL PROCEDURE DISKADDRESS(MID,FID,FPB3,A,H,IO); % (SHM)37286000
VALUE MID,FID,FPB3,A,H,IO; % (SHM)37286100
REAL MID,FID,FPB3,A,IO; % (SHM)37286200
ARRAY H[*]; 37286300
BEGIN LABEL EOF, EOF2; 37287000
INTEGER I; 37287250
REAL T, V; 37287500
IF A>=0 THEN% 37288000
BEGIN T+(A DIV H[0],[30:12])xH[0],[42:6];% 37289000
IF H[9] LEQ I:=(IF H[1]=0 THEN 0 ELSE T DIV H[1]) THEN 37290000
GO TO EOF; 37290100
IF H[I:=I+10]=0 THEN % NEW ROW NEEDED, 37291000
IF IO THEN GO TO EOF ELSE % EOF ON A READ, 37291200
IF IO=2 THEN % CALLED FROM FILEOPEN SO 37291400
BEGIN % DONT EXPAND THE FILE YET, 37291600
T:=1; 37291800
GO TO EOF2; 37292000
END 37292200
ELSE 37292400
IF H[4] THEN % IN DIRECTORY, UPDATE HEADER, 37292600
P(DIRECTORYSEARCH(=MID,FID,=H&I[CTF]),DEL) 37292800
ELSE % NOT IN DIRECTORY, 37293000
BEGIN % (SHM)37293210
IF (V:=FPB3,[18:5]) GTR 0 THEN % EU SPECIFIED % (SHM)37293220
V:=(IF V GTR 20 THEN 0 ELSE =V) ELSE % (SHM)37293230
IF (V:=FPB3,[16:2]) GTR 0 THEN % SPEED SPECIFIED % (SHM)37293240
V:=(IF V GTR 2 THEN 0 ELSE V) ELSE % (SHM)37293250
V:=0; % NO SPEED OR EU SPECIFIED % (SHM)37293260
H[I] := PETUSERDISK(H[8],V); % (SHM)37293270
END; % (SHM)37293330
T+H[I]+I+T MOD H[1];% 37294000
STREAM(D+[T]); BEGIN SI+D; DS+8 DEC END;% 37295000
END ELSE% 37296000
EOF: T+0;% 37297000
EOF2: 37297500
DISKADDRESS+T;% 37298000
END DISKADDRESS;% 37299000
37300000
37301000

```



```

PROCEDURE SETNOTINUSE(U,RWL); VALUE U,RWL; REAL U,RWL; 37302000
BEGIN REAL I,J; 37303000
  IF U<16 THEN P(WAIT10(@4200000000,@377,U),DEL); 37303200
  SLEEP([TOGGLE],STATUSMASK); 37304000
  RRRMECH*((I+TWO(U)) AND SAVEWORD) OR ((I+NOT I) AND RRRMECH);% 37305000
  READY*READY AND I;% 37306000
  IF RWL THEN 37312000
BEGIN 37313000
  STREAM(S+[TINU[U]],M+MULTITABLE[U],F+LABELTABLE[U], 37314000
    N+IF U<16 THEN PRNTABLE[U],[30:18] ELSE 0, 37314100
    T:=MULTITABLE[U]=0, TT:=U GEQ 16, DI:=J:=SPACE(10)); 37314200
  BEGIN SI+S; SI+SI+5; DS=LIT "#"; DS+3 CHR;% 37315000
    DS+6 LIT " RW/L "; SI+LOC M; SI+SI+1; 37316000
    DS+7 CHR; DS=LIT " "; SI+SI+1; DS+7 CHR; 37316100
    T(M+DI;DI+DI-15;DS+7FILL;DI+M); TT(JUMP 37316200
    OUT TO LA); DS=LIT "("; DS+5 DEC; DS=LIT")"; 37316300
    LA; DS=LIT "+"; 37316400
  END;% 37317000
  SPOUT(J); 37318000
  LABELTABLE[U]+@214; 37318100
END ELSE LABELTABLE[U]+@114; 37319000
  MULTITABLE[U]+RDCTABLE[U]+0; 37319010
  IF U<16 THEN PRNTABLE[U]+0; 37319020
END SETNOTINUSE; 37319100
PROCEDURE BLASTQ(U); 37320000
VALUE U; REAL U; 37321000
BEGIN 37322000
  REAL I,X; 37323000
  BOOLEAN SUBROUTINE CHECKIO; 37323100
  BEGIN 37323200
    CHECKIO:=(I:=UNIT[U]),[5:8]#0 OR (I,[14:1] AND I,[13:5]#@31); 37323300
  END; 37323400
  IF CHECKIO THEN COMPLEXSLEEP(NOT CHECKIO); 37324000
  IF I,[16:1] THEN % SKIP I/O IN PROCESS 37326000
  BEGIN I:=NFLAG(LOCATQUE[X:=I,[FF]]); 37327000
    LOCATQUE[X],[FF]:=@77777; 37328000
    UNIT[U],[CF]:=X; 37329000
  END ELSE 37330000
    UNIT[U],[5:43]:=(NOT 0),[18:30]; 37331000
  WHILE (I:=I,[FF])#@77777 DO 37332000
  BEGIN RETURNIOSPACE(I); 37333000
    IOCOUNT[P1MIX]:=(+P(DUP))-1; 37333500
    I:=NFLAG(LOCATQUE[I]); 37334000
  END; 37335000
END BLASTQ;% 37336000
PROCEDURE BUILDLABEL(LABLE,MID,FID,REEL,CDATE,CYCLE,PFACT,PTN,BLKODE,% 37337000
  BSIZE,RSIZE);% 37338000
  VALUE LABLE,MID,FID,REEL,CDATE,CYCLE,PFACT,PTN,BLKODE,% 37339000
  BSIZE,RSIZE;% 37340000
  ARRAY LABLE[*];% 37341000
  REAL MID,FID,REEL,CDATE,CYCLE,PFACT,PTN,BLKODE,% 37342000
  BSIZE,RSIZE;% 37343000
BEGIN;STREAM(D+[PFACT]); BEGIN SI+D; SI+SI+5; DS+3 OCT END;% 37344000
  PFACT+CALCULATEPURGE(PFACT);% 37345000
  STREAM(S+[MID],LABLE);% 37346000
  BEGIN DS+8 LIT " LABEL "; SI+S; DS+2 WDS;% 37347000
    DS+3 DEC; DS+5 DEC; DS+2 DEC; SI+SI+3; DS+5 CHR;% 37348000
    DS+14 LIT "0"; DS+5 DEC; SI+SI+7; DS+CHR;% 37349000
    DS+5 DEC; DS+5 DEC; DS+11 LIT "0"% 37350000
  END;% 37351000

```

```

IF (BSIZE+LABLE,[8:10])>10 THEN% 37352000
STREAM(J+JARROW[P1MIX],D+[LABLE[10]]);% 37353000
BEGIN SI+J; SI+SI+1; DS+LIT " "; DS+7 CHR;% 37354000
SI+SI+1; DS+LIT "/" ; DS+7 CHR; 12(DS+2 LIT " ");% 37355000
END END GUILDLABEL;% 37356000
PROCEDURE FILEMESSAGE(I,K,M,F,R,D,C,TYPE); 37357000
VALUE I,K,M,F,R,D,C,TYPE; 37357100
REAL I,K,M,F,R,D,C,TYPE; 37357200
BEGIN REAL Z,L;% 37359000
L 1= SPACE(12); 37360000
STREAM(Z[I+[1]],J+[JAR[P1MIX,*]],P1MIX,L); 37361000
BEGIN SI+1; 37362000
IF SC="+" THEN BEGIN TALLY+1; DS+LIT "="; SI+SI+1 END ELSE% 37363000
BEGIN SI+SI+1; IF SC="#" THEN DS+LIT " " END;% 37364000
DS+7 CHR; DS+LIT " "; L+DI;% 37365000
2(DI+LOC Z; IF 8 SC#DC THEN BEGIN DI+L; SI+SI-7; DS+7 CHR;% 37366000
DS+LIT " "; L+DI END);% 37367000
DI+L; SI+SI+1; DS+7 CHR; DS+LIT " "; L+DI;% 37368000
3(DI+LOC Z; IF 8 SC#DC THEN BEGIN DI+L; SI+SI-8; DS+7 DEC; 37369000
L+DI; DI+DI-7; DS+6 FILL; 37370000
DI+L; DS+LIT " "; L+DI; 37371000
END); 37372000
37373000
37374000
37375000
DI+L; DI+DI-1; DS+LIT ";";% 37376000
Z+TALLY; SI+LOC Z; SI+SI+7;% 37377000
IF SC="0" THEN BEGIN SI+J; SI+SI+1; DS+7 CHR; DS+LIT "/" ;% 37378000
SI+SI+1; DS+7 CHR; DS+LIT "=" ;% 37379000
SI+LOC P1MIX; DS+2 DEC; 37379500
L+DI; DI+DI-2; DS+FILL; DI+L END; 37380000
DS+LIT ";";% 37381000
END;% 37382000
IF P THEN BEGIN TERMINATE(P1MIX); TERMINALMESSAGE(=L) END;% 37383000
SPOUTER(L,0,TYPE); 37384000
END FILEMESS;% 37385000
PROCEDURE FILLBUFFERS(CURRENT,FINAL,COBOL,NR); 37385500
VALUE CURRENT,FINAL,COBOL,NR; 37386000
REAL CURRENT,FINAL,COBOL,NR; 37387000
BEGIN ARRAY LOCAT[*];% 37388000
INTEGER I,J,K,D;% 37388100
INTEGER FIRSTLOC=J,PREVLOC=K,CURLOC=D; 37388200
REAL T=LOCAT; 37388250
REAL T1; 37388275
REAL NF=T1+1; % MUST BE AT THE TOP OF THE STACK 37388300
LABEL LINK; 37388400
REAL BSIZE=CURRENT,N=FINAL,U=COBOL,ALPHA=NR; 37388500
IF ALPHA<512 THEN 37388600
BEGIN 37388700
P(NR=(COBOL GTR 0)); % INITIALIZE NF 37388800
IF COBOL THEN FINAL:=CURRENT; 37388900
J+FINAL,[33:15]=K+CURRENT,[33:15];% 37390000
D+2&(NOT CURRENT)[1:22:1];% 37391000
LOCAT+M[K+D]; NR+NR-1;% 37392000
FOR I+1 STEP 1 UNTIL NF DO% 37393000
BEGIN IOREQUEST(FLAG(FINAL),CURRENT,LOCAT);% 37394000
M[LOCAT]+M[LOCAT]&0[26:26:7] AND NOT(M OR IOMASK);% 37394025
IF NOT COBOL THEN 37394050
IF I=1 THEN IF P(FINAL,[3:5],DUP)=6 OR P(XCH)=7 THEN 37394100
BEGIN 37394150
SLEEP(LOCAT & 0 [3:3:30],IOMASK);

```

```

STREAM(N=0,L=0;NDIV64=0,BACC=T1*FINAL,[7:1]);
      BUF ← (M[LOCAT] INX T1)-(1-T1));
BEGIN  DI ← LOC N; SI ← BUF; BACC(SI ← SI+4);
      IF 4 SC≠DC THEN GO OWT;
      DI ← LOC N; BACC(SI ← BUF); DS ← 4 OCT;
      SI ← LOC L; DI ← LOC BACC; SI ← SI-2; DI ← DI-1;
      DS ← 1 CHR; SI ← BUF;
      CI ← CI+BACC; GO FWD;
      NDIV64(SI ← SI-32; SI ← SI-32); SI ← SI-N; SI ← SI+4;
      GO ON;
FWD:   NDIV64(SI ← SI+32; SI ← SI+32); SI ← SI+N;
ON:    DI ← LOC L; DS ← 4 OCT;
OWT:
END STREAM;
T1 ← P;
IF P(DUP)=0 OR P(XCH)≠T1 THEN TERMINATE(P1MIX&86 [CTF]);
END;
      IF NR>0 THEN STREAM(NR,T←M[LOCAT],LOCAT);%
          BEGIN SI←LOCAT; SI←SI+8; DS←NR WDS;%
              SI←LOC T; DS←WDS END;%
          CURRENT,[33:15]←K←M[K+D],[18:15];%
          FINAL,[33:15]←K+J;%
      END ELSE
      BEGIN
      T←ALPHA&U[12:42:6] OR M;%
      FOR I←N-1 STEP -1 UNTIL 0 DO%
      BEGIN M[ALPHA+I]←(CURLOC←GETSPACE(BSIZE+4,3,1)+2)+2;
          IF FIRSTLOC=0 THEN FIRSTLOC←CURLOC;%
          M[CURLOC+1]←0; MOVE(BSIZE+1,CURLOC+1,CURLOC+2);
LINK:  M[CURLOC]←FLAG(T)&(PREVLOC+2)[18:33:15];%
          M[CURLOC+BSIZE+3]←FLAG(T)&(PREVLOC+BSIZE+1)[18:33:15];%
          PREVLOC←CURLOC;%
      END;%
      IF I≠(-1) THEN BEGIN CURLOC←FIRSTLOC; GO TO LINK END;%
      END END FILL OR GET BUFFERS;
REAL PROCEDURE FILEHEADER(MID, FID, NROWS, SIZE, BLEN, RLEN, S);
VALUE  MID, FID, NROWS, SIZE, S;
REAL  MID, FID;
INTEGER NROWS, SIZE, BLEN, RLEN, S;
      BEGIN REAL  Q, Z;
      $ SET OMIT = NOT SHAREDISK
          INTEGER HDRSIZE;
      $ POP OMIT
          LABEL T1FILL,EXIT;
          ARRAY  T = Q[*];
          INTEGER LPER,
                  SPER,
                  N1,
                  R1,
                  L1,
                  W;
          SUBROUTINE GOBBLE;
          BEGIN SPER := (BLEN+29) DIV 30;
              IF S=0 THEN RLEN := BLEN;
                  LPER := BLEN DIV RLEN;
          END GOBBLE;
      $ SET OMIT = SHAREDISK
          Q:=S,[13:3];
      $ POP OMIT
      $ SET OMIT = NOT SHAREDISK

```

```

37394200
37394250
37394260
37394280
37394300
37394350
37394360
37394400
37394450
37394460
37394500
37394550
37394560
37394600
37394650
37394700
37394800
37395000
37396000
37397000
37398000
37399000
37400000
37401000
37404000
37405000
37406000
37407000
37408000
37412000
37413000
37414000
37415000
37416000
37417000
37418000
37419000
37420000
37421000
37422000
37422199
37422200
37422201
37422300
37423000
37424000
37424100
37424200
37424300
37424400
37424500
37425000
37426000
37427000
37428000
37429000
37429499
37429500
37429501
37429599

```



```

PROCEDURE PURGEIT(U); VALUE U; INTEGER U;%
BEGIN ARRAY LABEL[*];
REAL EOF;

RDCTABLE[U],[8:6] := 0;%MAKE THE USER THE MCP
P(WAITIO(@4200000000,@377,U),DEL);
LABEL := [MSPACE(10)]&10[8:38:10]&5[21:45:3];
BUILDLABEL(LABEL,0,"X",1,0,1,0,PRNTABLE[U],[30:18],0,0,0);%
P(WAITIO(LABEL,@37700000,U),DEL);%
EOF:=@173700000000000000;%
P(WAITIO([EOF],@37700000,U),DEL);%
FORGETSPACE(LABEL,[33:15]);
SETNOTINUSE(U,0);
KILL([U] INX NOT 1);
END PURGEIT;%

PROCEDURE KRUNCHER(H); ARRAY H[*];
BEGIN DEFINE E=H[7]#,RL=H[1]#,RPB=H[0],[30:12]#,
MAXROWS=H[9],[43:5]#,
BCL=H[0],[42:6]#,BRL=H[8]#;
ARRAY A[*];
LABEL FORGET,EXIT,AGAIN,DONE;
INTEGER NB,NBR;
REAL I,J,K,T;
A:=[MSPACE(41)]&40[8:38:10];
MOVE(41,A,[CF]-1,A);
IF E LSS 0 THEN GO TO EXIT;
NB:=E DIV RPB;
NBR:=RL DIV BCL;
IF RL NEQ BRL THEN
FOR I:=10 STEP 1 UNTIL 29 DO
IF H[I] NEQ 0 THEN
$ SET OMIT = SHAREDISK
FORGETUSERDISK(H[I]+RL,BRL=RL);
$ SET OMIT = NOT SHAREDISK
FORGETUSERDISK(H[I]+RL,(BRL=RL)&(NOT H[4])[1:47:1]);
$ RESET OMIT
BRL:=RL;
IF NB LSS NBR THEN
BEGIN A[0]:=H[NT2:=10];
NT4:=1;
RL:=(NB+1)*BCL;
GO TO FORGET;
END;
T:=(K:=J:=1)+NBR*20;
AGAIN: IF(NT1:=NBR DIV J)=0 THEN GO TO DONE;
IF (NT2:=NB DIV NT1) GTR 19 THEN GO TO DONE;
IF NBR MOD J=0 THEN
BEGIN IF (NT3:=NT1*NT2+NT1) LSS T THEN
BEGIN K:=J; T:=NT3; NT4:=NT2+1 END;
END;
J:=J+1;
GO TO AGAIN;
DONE: IF K=1 THEN GO TO EXIT;
NT2:=NB DIV NBR + 10;
RL:=RL DIV K;
FOR I:=10 STEP 1 UNTIL NT2 DO
BEGIN IF (NT1:=H[I]=RL) GTR 0 THEN
FOR J:=1 STEP 1 UNTIL K DO
A[(I-10)*K+J-1]:=NT1+J*RL;
END;
END;

```

```

37449000
37450000
37451000
37452000
37453000
37453100
37454000
37455000
37456000
37457000
37458000
37463000
37464000
37465000
37466000
%R1737500000
%R1737501000
37501500
%R1737502000
%R1737504000
%R1737505000
%R1737506000
%R1737507000
37508000
%09737509000
%R1737510000
%R1737511000
%R1737512000
%R1737513000
%R1737514000
%R1737515000
37515995
37516000
37516050
37516100
37516105
%R1737517000
%R1737520000
%R1737521000
%R1737521100
%R1737521200
%R1737521300
%R1737521400
%R1737522000
%R1737523000
%R1737524000
%R1737525000
%R1737526000
%R1737527000
%R1737528000
%R1737529000
%R1737530000
%R1737530100
%R1737530200
%R1737531000
%R1737532000
%R1737533000
%R1737534000
%R1737535000
%R1737536000

```

```

FOR K:=NT4 STEP 1 UNTIL 19 DO A[K]:=0; %R1737538000
IF MAXROWS LSS (NT5:=(NT4#20)+NT4) THEN MAXROWS:=NT5; 37538500
FORGET: IF NB+1 NEQ NBR THEN %R1737539000
$ SET OMIT = SHAREDISK 37541995
FORGETUSERDISK(A[NT4-1]+RL,(NT2-9)*BRL-NT4*RL); 37542000
$ SET OMIT = NOT SHAREDISK 37542005
FORGETUSERDISK(A[NT4-1]+RL, 37542010
((NT2-9)*BRL-NT4*RL)&(NOT H[4])[1:47:1]); 37542020
$ RESET OMIT 37542025
MOVE(20,A,[H[10]]); %R1737543000
BRL:=RL; %R1737544000
EXIT: FORGETSPACE(A); %R1737545000
END; %R1737546000
PROCEDURE DISKFILEOPEN(ALPHA); VALUE ALPHA; INTEGER ALPHA;% %R9038000000
BEGIN REAL RCW:=+0, MSCW:=+2; %R9038001000
REAL IOM:=IOMASK, IOMASK:=+1; %R9038002000
INTEGER NBUFS:=+2, FNUM:=+3, BLEN:=+4, TYPE:=+5, REEL:=+6, CDATE:=+7, %R9038003000
CYCLE:=+8, MODE:=+9, IO:=+10, RLEN:=+11, U:=+12, KIND:=+13, %R9038004000
DIREC:=+14, FORMS:=+15, COBOL:=+16, UNLABELED:=+17, 38005000
OPTIONAL:=+18, CNTCTL:=+19; 38006000
REAL MFID:=+20, FID:=+21, T1:=+22, T2:=+23, MASK:=+24, STATE:=+25; 38007000
ARRAY FIB:=+26[*], FPB:=+27[*];% %R9038008000
INTEGER ACCESS:=+28, FIB7:=+29; %R9038009000
ARRAY HEADER:=+30[*];% %R9038010000
LABEL MSG,EXIT; 38010500
SUBROUTINE DISKSETUP;% %R9038011000
BEGIN IF STATE,[42:1] THEN% %R9038012000
BEGIN 38013000
IF (NOT MFID)=(NOT 0)AND NOT USERCODE[P1MIX],[1:1] THEN 38013100
BEGIN 38013110
FPB[FNUM]:=MFID:=FID; 38013120
FPB[FNUM+1]:=FID:=USERCODE[P1MIX]; 38013130
END; 38013140
IF NFLAG(FIB[14])=FLAG(FILEHEADER(MFID 38013200
$ SET OMIT = NOT SHAREDISK 38013299
&(TYPE=26))[1:47:1] 38013300
$ POP OMIT 38013301
,FID&FIB[5][1:45:1],FIB[8],[20:5] 38013400
,FIB[8],[25:23],BLEN,RLEN,STATE)))<6 THEN 38013600
BEGIN P(DEL); 38013800
$ SET OMIT = NOT SHAREDISK 38013899
FIB[5],[45:1]+1; % FOR PAR LABEL ACTION 38013900
FIB[8],[3:5]+FIB[14]; % IOSTATUS UPDATE 38014000
$ POP OMIT 38014001
T1:=1; 38014200
GO TO EXIT; 38014400
END; 38014600
IF FIB[8],[20:28]#0 THEN FPB[FNUM+2],[18:30]+DATE ELSE 38015000
IF CDATE NEQ 0 THEN % LABEL EQUATION DATE SPECIFIED 38015100
BEGIN 38015200
HEADER := FIB[14]; 38015300
IF CDATE NEQ HEADER[3],[30:18] THEN % WRONG DATE 38015400
BEGIN % WRITE DATE CHECK MESSAGE 38015500
STREAM(H:=HEADER[3],[30:18],T2:=[T2]); 38015600
BEGIN SII=LOC H; DS:=8DEC; END; 38015700
MSG: FILEMESS("#DAT CK", " =00000"&T2[18:18:30], 38015800
MFID, FID, REEL, CDATE, CYCLE); 38015900
REPLY[P1MIX]:= -VWY&VOK[36:42:6]&VFM[30:42:6]; 38016000
IF [MEM[P1MIX,MLINK1]],[CF] GEQ FENCE THEN 38016100
SWAP(WAITSWAP,1) ELSE 38016200

```

```

COMPLEXSLEEP( (TERMSET(P1MIX) OR REPLY[P1MIX] GTR 0)); 38016300
IF TERMSET(P1MIX) THEN 38016400
    BEGIN 38016500
        FORGETSPACE(DIRECTORYSEARCH(MFID,FID,
$ SET OMIT = NOT SHAREDISK 38016600
            IF TYPE=26 THEN 22 ELSE 38016649
$ POP OMIT 38016650
                FIB[5],[13:3]+10)); 38016700
                GO TO INITIATE; 38016800
                END; 38016900
                IF NOT WHYSLEEP( 38017000
                VWY&VOK[36:42:6]&VFM[30:42:6]) THEN GO TO MSG; 38017100
                IF (T1:=REPLY[P1MIX],[CF]) NEQ VOK AND 38017200
                T1 NEQ VFM THEN GO TO MSG; 38017300
                T1 := CDATE := 0; 38017400
                FPB[FNUM+2],[18:30] := T2; % USE OLD DATE 38017500
                END; % IF DATE CHECK MESSAGE 38017600
                END; % IF LABEL EQUATION DATE SPECIFIED 38017700
                FIB[18] := RLFN & BLEN[CF] & BLEN[3:33:15]; 38017710
                STATE,[46:2] := BLEN NEQ RLEN; 38017720
                STARTIMING(FNUM,18); 38018000
                FPB:=PRT[P1MIX,3]; % STARTIMING MAY HAVE MOVED IT, 38018500
                END;% %R9038019000
                HEADER+FIB[14];% %R9038020000
                KIND+4; U+18;% %R9038021000
                MODE+0;% %R9038022000
                IF NOT COBOL THEN UNLABELED+1;% %R9038023000
                CNTCTL+BLENS<1023;% %R9038024000
$ SET OMIT = NOT SHAREDISK 38024004
    IF TYPE#26 THEN 38024005
$ POP OMIT 38024006
        IF FIB[8],[20:28]=0 THEN % NOT CREATING 38024010
        IF HEADER[8]<((BLEN+29) DIV 30) THEN %BLKSIZE > ROWSIZE 38024020
        BEGIN BLEN+HEADER[8]*30; FORGETSPACE(HEADER INX 0); 38024030
        P(DIRECTORYSEARCH(=MFID,FID,STATE,[13:3]+10),DEL); 38024040
        FIB[14]+HEADER+FLAG(FILEHEADER(MFID,FID&FIB[5][1:45:1],0,0, 38024050
            BLEN,RLEN,STATE)); 38024060
        FIB[18],[3:15]+BLEN; 38024070
        END; 38024080
        IF COBOL>0 AND (FIB[13],[22:1] OR TYPE=10 OR TYPE=26) THEN 38024100
        BEGIN COBOL:=3; %IF COBOL=10 OR COBOL=RANDOM 38024200
            BLEN := BLEN + RLEN; %THEN CHANGE BUFFSIZE TO 38024300
        END; %BUFFSIZE + RECSIZE 38024400
            GETBUFFERS((IF CNTCTL THEN BLEN% %R9038025000
                ELSE ((BLEN+29) DIV 30)*30)+1,% %R9038026000
                NBUFS,U,ALPHA);% %R9038027000
            IF COBOL = 3 THEN %IF COBOL=10 OR COBOL=RANDOM 38027100
            BEGIN COBOL := 1; %THEN CHANGE BUFFSIZE TO 38027200
                BLEN := BLEN - RLEN; %BUFFSIZE = RECSIZE 38027300
            END; % (SEE ABOVE) 38027400
                FIB[16]+M[ALPHA]&CNTCTL[23:47:1]&I0[24:47:1];% %R9038028000
                &((BLEN+29) DIV 30)[27:42:6];% %R9038029000
                &(IF CNTCTL THEN BLEN ELSE 1023)[8:38:10];% %R9038030000
                &TINU[18][3:3:5] OR M OR IOMASK;% %R9038031000
                FIB[16],[2:1]:=(HEADER,[31:2] AND (10+1))#0; %R9038032000
                FIB[5],[1:1]:= NOT FIB[16],[2:1]; %R9038033000
                IF FIB[5],[1:1] THEN %R9038034000
                FOR MASK:=10 STEP 1 UNTIL 29 DO HEADER[MASK]:=0; %R9038035000
                FIB[19]+(IF DIREC THEN BLEN=RLEN+1 ELSE 1) %R9038036000
                INX FIB[16]&0[27:27:6]; %R9038037000

```

```

IF STATE,[46:2]#0 THEN FIB[19],[8:10]+RLEN;% %R9038038000
FS[P1MIX,(T2:=(FNUM DIV ETRLNG)),[40:4]]+(*P(DUP)) OR %R9038039000
(TWO(O&T2[43:44:4])*((NOT HEADER),[31:2])); %R9038040000
T2+IF COBOL THEN 0 ELSE FIB[19],[33:15]=FIB[16],[33:15];%R9038041000
FIB[10],[3:15]=M[ALPHA]=2; % HEAD OF BUFFER RING 38041100
FOR MASK+0 STEP 1 UNTIL NBUFS=1 DO% %R9038042000
M[ALPHA+MASK]+(P(DUP,LOD)+T2)% %R9038043000
&P(FLAG(FIB[19]=ABS(3*COBOL)),XCH)[CTC]; 38044000
FIB[16]=FIB[16] OR M; %R9038045000
FIB[5],[45:1]+0; 38045100
IF P([FIB[14]],LOD),[FF]=2 THEN FIB[5],[11:2]+1;%INPUT ONLY,38045105
IF HEADER[4],[10:1] AND NOT IO THEN 38045110
FILEMESS("CODE ", "FILE ",MFID,FID,0,0,0); 38045120
$ SET OMIT = NOT(PACKETS) 38045149
IF PSEUDOMIX[P1MIX]#0 THEN 38045150
IF NOT FIB[5],[41:1] THEN 38045155
FILEMESSAGE((IF IO THEN " IN " ELSE " OUT") 38045160
&TINU[U][6:30:18], IF ACCESS=0 THEN " SER " 38045200
ELSE IF ACCESS=1 THEN IF TYPE=26 THEN " PRO " 38045300
ELSE " RDM " ELSE " UPD ", 38045310
MFID,FID,0,0,0,64); 38045400
$ POP OMIT 38045501
END DISKSETUP;% %R9038046000
P(RCW,MSCW,STF); %R9038047000
RCW:=RCW&P(XCH)[CTC]; %R9038048000
DISKSETUP; %R9038049000
IF COBOL<0 THEN % ADJUST UPPER BOUND FOR COBOL 68 38049200
BEGIN MASK + (IF IO AND NOT FIB[13],[22:1] 38049300
THEN HEADER[7] 38049400
ELSE ((HEADER[9] * HEADER[1]) DIV 38049500
HEADER[0],[42:6]) * HEADER[0],[30:12]) - 1);38049600
IF FIB[3]=0 OR FIB[3]>MASK THEN FIB[3]=MASK; %LESSOR OF 2 EVILS38049700
END; 38049800
IF P(TYPE,DUP)=10 OR P(XCH)=26 THEN 38050000
BEGIN %R9038051000
IF COBOL<1 THEN % ALGOL OR COBOL 68 38052000
FOR MASK + 0 STEP 1 UNTIL NBUFS=1 DO 38053000
IF COBOL THEN M[M[ALPHA+MASK] INX NOT 2] + NOT 0 38053500
ELSE M[ALPHA+MASK]+P(DUP,LOD)&1[27:47:1]; 38054000
FIB[6]=FIB[7]+0;% %R9038055000
FIB[17]=IF IO THEN 0 ELSE BLEN;% %R9038056000
END ELSE %R9038057000
BEGIN %R9038058000
T2=(MFID+FIB[16]),[33:15];% %R9038059000
FIB7=FIB[7]; %R9038060000
IF COBOL THEN% %R9038061000
BEGIN IF COBOL>0 THEN 38062000
IF NOT (FIB7=0 OR FIB[13],[22:1]) THEN 38062500
BEGIN FIB7 + FIB7 = 1; 38063000
OPTIONAL + NBUFS = 1; 38063500
END ELSE OPTIONAL + NBUFS = 2 38064000
ELSE BEGIN % COBOL 68 38064200
OPTIONAL + NBUFS = 1; 38064400
IF DIREC THEN FIB7 + FIB[7] + FIB[3]; 38064600
END; 38065000
FID=FIB[16];% %R9038066000
MASK=0;% %R9038067000
END ELSE% %R9038068000
BEGIN OPTIONAL+NBUFS=1;% %R9038069000
MASK=(FID+FIB[19]),[33:15]=T2;% %R9038070000

```



```

END;% %R9038071000
IF STATE,[46:2]#0 OR IO THEN %R9038072000
IF M[ALPHA],[2:1] THEN %R9038073000
FOR T1=0 STEP 1 UNTIL OPTIONAL DO% %R9038074000
BEGIN IF (M[T2])#% 38074500
    DISKADDRESS(FPB[FNUM], FPB[FNUM+1], FPB[FNUM+3], 38075000
    FORMS:=((HEADER[0],[30:12]*T1)&DIREC[1:47:1])+FIB7,38075500
    HEADER, IO&(NOT HEADER[4])[46:47:1]) > 1 THEN 38076000
BEGIN 38076500
    IF (USERCODE[P1MIX] EQV MCP)#NOT 0 THEN 38077000
    IF P(M[MFID],DUP),[3:6]#0 AND 38077500
        P(XCH)<DIRDSK*DSKTOG THEN 38078000
    BEGIN 38078500
        TERMINATE(P1MIX); 38079000
        TERMINALMESSAGE(30); 38079500
    END; 38080000
    IOREQUEST(FLAG(FID),MFID&1[24:47:1],M[T2=2]); 38080500
    M[ALPHA]=FLAG(MFID)&0[26:26:7] AND NOT 38081000
        (M OR IOMASK); 38081250
    END ELSE 38081500
    IF M[T2]#0 THEN % EOF IF INPUT, FULL HDR IF OUTPT 38081750
        M[ALPHA]=P(DUP,LOD)&1[27:47:1] AND NOT M; 38082000
    IF COBOL<0 THEN M[M[ALPHA] INX NOT 2] * 38082400
        (IF FORMS#0 THEN FORMS DIV FIB[11] ELSE NOT 0); 38082500
    STREAM(N=NBUFS-1,T=M[ALPHA],ALPHA);% %R9038083000
    BEGIN SI=ALPHA; SI=SI+8; DS=N WDS;% %R9038084000
        SI=LOC T; DS=WDS;% %R9038085000
    END;% %R9038086000
    MFID,[33:15]+T2+M[T2=2],[18:15];% %R9038087000
    FID,[33:15]+T2+MASK;% %R9038088000
    END;% %R9038089000
    IF (NBUFS-1)#OPTIONAL THEN FIB[16],[33:15]+M[ALPHA] ;% %R9038090000
    FORMS=(FORMS+FIB7 MOD HEADER[0],[30:12])*RLEN; %R9038091000
    SLEEP([M[ALPHA]],IOMASK);% %R9038092000
    IF COBOL # 0 THEN % NOT COBOL 68 38092900
        IF FIB[13],[22:1]THEN M[ALPHA],[33:15]+FIB[16]INX 1 ELSE%R38093000
        M[ALPHA],[33:15]+FIB[16],[33:15]+FORMS+1;% %R9038094000
        IF (NBUFS-1)#OPTIONAL AND IO AND NOT FIB[13],[22:1] THEN%R38095000
            FIB[ 17 ]+0 ELSE %R9038096000
            FIB[17]+IF DIREC THEN FORMS+RLEN% %R9038097000
                ELSE BLEN=FORMS;% %R9038098000
    END; %R9038099000
    T1=0; 38099400
EXIT; 38099500
P(P&RCW[CTC],0,RDS,0,XCH,P&P[CTF],STF); %R9038100000
END DISKFILEOPEN; %R9038101000
PROCEDURE OTHERFILEOPEN(IN ALPHA); VALUE ALPHA; INTEGER ALPHA; 38102000
BEGIN REAL RCW=+0, MSCW=+2; %R9038102100
    REAL IOM=IOMASK, IOMASK=+1; %R9038102200
    INTEGER NBUFS=+2, FNUM=+3, BLEN=+4, TYPE=+5, REEL=+6, CDATE=+7, %R9038102300
        CYCLE=+8, MODE=+9, IO=+10, RLEN=+11, U=+12, KIND=+13, %R9038102400
        DIREC=+14, FORMS=+15, COBOL=+16, UNLABELED=+17, 38102500
        OPTIONAL=+18, CNTCTL=+19; 38102600
    REAL MFID=+20, FID=+21, T1=+22, T2=+23, MASK=+24, STATE=+25; 38102700
    ARRAY FIB=+26[*], FPB=+27[*];% %R9038102800
    INTEGER ACCESS=+28, FIB7=+29; %R9038102900
    ARRAY HEADER=+30[*];% %R9038103000
    REAL USASI=NT1, RHEAD=HEADER; 38103100
    LABEL FIND,DCN; 38103200
    SUBROUTINE TYPEOPEN;% %R9038103400

```

```

BEGIN
    T1:=(COPNMESS AND ((T1:=JAR[P1MIX,0])>0 OR
        COPNMESS AND T1<0)) OR OPENK;
    NT2:=0;
    IF U<16 THEN
        STREAM(S:=PRNTABLE[U],[30:18], D:=[NT2]);
        BEGIN SI:=LOC S; DS:=8 DEC;
            DI:=DI-7; DS:=6 FILL;
        END;
        FILEMESSAGE((" IN ")&
            TINU[U][6:30:18], NT2, FPB[FNUM], FPB[FNUM+1],
            IF KIND=2 OR KIND=9 THEN P(REEL,CDATE) ELSE P(0,0),
            P,CYCLE,T1);
    END;
SUBROUTINE REED;%
    BEGIN IF (T2*WAITIO(T1,(MASK OR @40)&@377[CTF],U) AND @367)≠0 THEN
        IF (T2 AND NOT MASK)≠0 THEN
            BEGIN STOPTIMING(FNUM,1023); SETNOTINUSE(U,0);
                FILEMESS("PARITY ", "ON ... "&TINU[U][24:30:18],%
                    MFID,FID,REEL,CDATE,CYCLE);%
            END;%
            IF TERMSET(P1MIX) THEN
                BEGIN STOPTIMING(FNUM,1023); SETNOTINUSE(U,0);
                    GO TO INITIATE;
                END;
        END REED;%
REAL SUBROUTINE CNTLBITS;%
    CNTLBITS←IOMASK&MODE[21:47:1]&DIREC[22:47:1]&CNTCTL[23:47:1]
        &IO[24:47:1]&(KIND=7 OR KIND>9 AND KIND≤12)[20:47:1]
        &(IF KIND=10R KIND=7OR KIND=12THEN@20ELSE 0)[27:42:6];
SUBROUTINE LABELAREA;%
    M[T1←ALPHA=2]←M OR (GETSPACE((T1←M[T1],[8:10])+4,2,1)+4)
        &T1[8:38:10]&CNTLBITS[18:18:15];%
SUBROUTINE DOCARDLABEL;
    BEGIN NT3 := SPACE(13)+2;
        MOVE(10,T1,NT3);
        FORGETSPACE(T1-2); T1←NT3;
        M[ALPHA=2]←[M[T1]]&10[8:38:10]&1[24:47:1];
        MODE←CNTCTL←DIREC←0;
    END;
P(RCW,MSCW,STF);
RCW:=RCW&P(XCH)[CTC];
IF STATE,[41:1] THEN%
    BEGIN U←FIB[15],[25:5];%
    END ELSE%
    BEGIN IF (U←FINDINPUT(MFID,FID,REEL,CDATE,CYCLE,COBOL,UNLABELED,%
        OPTIONAL,MODE,FNUM))<0 THEN%
        BEGIN FIB[5],[39:4]←9; GO TO FIND END;%
        STARTIMING(FNUM,IF U>31 THEN 18 ELSE U);
        FPB:=PRT[P1MIX,3]; % STARTIMING MAY HAVE MOVED IT,
        KIND:=IF U GTR 31 THEN 11 ELSE UNIT[U],[1:4];
        TYPEOPEN;%
        IF U<16 THEN BEGIN RRRMECH←TWO(U) OR RRRMECH;
            PRNTABLE[U],[15:15]←ALPHA;%
        END;%
        IF (T1←RDCTABLE[U],[14:10])≠0 THEN REEL←T1;
        STATE,[39:4]←0;%
    END;

```

```

%R9038103500
38103600
38103700
38103800
38103900
38104000
38104100
38104200
38104300
38104400
38104500
38104600
38104700
%R9038104800
%R9038105300
38105400
%R9038105500
%R9038105600
%R9038105700
%R9038105800
%R9038105900
%R9038106000
%R9038106100
%R9038106200
%R9038106300
%R9038106400
%R9038106500
38106600
%R38106700
38106800
%R9038106900
%R938107000
%R9038107100
%R9038107200
%R9038107300
%R9038107400
%R9038107500
%R9038107600
%R9038107700
%R9038107800
%R9038107900
%R9038110000
%R9038110500
%R9038111500
%R9038112000
%R9038112500
%R38113000
%R9038113500
%R9038114000
%R9038114500
38115000
38115100
%R9038115500
%R9038116000
%R9038116500
%R9038117000
38117500
%R9038118000
%R9038118500
%R9038119000

```

```

END;%
IF KIND=0 THEN%
BEGIN IF U=23 THEN BEGIN T1=READER; READER=0 END%
      ELSE BEGIN T1=READERB; READERB=0 END;%
      DOCARDLABEL;
      IF BLEN<T1*(MODE+1)*10 THEN BLEN=T1;%
END ELSE%
IF KIND=2 THEN%
BEGIN IF NOT UNLABELED THEN BEGIN%
      IF DIREC AND NOT FIB[16],[22:1] THEN
      BEGIN IF NOT STATE,[40:1] THEN BEGIN%
            T1=5&3[23:46:2] OR M;%
            MASK=0; REED;%
            MASK=@60; DO REED UNTIL T2,[42:1];
            DO REED UNTIL T2,[42:1];
            MASK=0; REED;          END;%
      END;

      CNTCTL+1; LABELAREA;%
      T1=NFLAG(M[ALPHA=2]);
      IF DIREC THEN T1=T1,[8:10]=1 INX T1;
      MASK=@40; REED;
      STREAM(Y:=0;X:=0,X1:=0,X2:=0,Z:=T1);
      BEGIN DI:=LOC X; DS:=24 LIT "VOL1HDR1HDR2EOF1EOF2EOV1";
            DI:=LOC X;
            6(TALLY:=TALLY+1);
            SI:=Z;
            IF 4 SC=DC THEN
            JUMP OUT TO B);
            TALLY:=0;

      B;
      Y:=TALLY;

      END;
      IF (USASI=P)>0 THEN
      USASITAPE(T1,[CF],USASI,4,U,DIREC) ELSE
      IF M[T1 INX 6],[24:6]=1 THEN
      BEGIN
      REED;
      MASK=@60;
      T1=5&3[23:46:2] OR M;
      T2=0;
      END;
      IF T2 NEQ @40 THEN DO REED UNTIL T2,[42:1] ELSE
      FOR CNTCTL+DIREC STEP 1 UNTIL 2 DO% DIREC = 0 OR 1 %DB 38141500
      P(WAITIO(@4740000005&(NOT DIREC)[22:47:1],@377,U),DEL);%DB38142000
      END;%
      IF BLEN = 0 THEN
      BEGIN;STREAM(B:=0,BF:=0,R:=0;L="LABEL ",S=M[ALPHA=2]);%R9038143500
            BEGIN SI:=LOC L; SI:=SI+1; DI:=DI+1;%
            IF 7SC = DC THEN%
            BEGIN SI:=S; SI:=SI+58; DI:=LOC BF; DI:=DI+1;%
                  DS:=CHR; DS:=5 OCT; DS:=5 OCT;
            END%
            END STREAM;
            RLEN := POLISH; BLEN := POLISH;
            STATE := STATE & P(XCH)[46:46:2];
            FIB[18] := RLEN & BLEN[CTF] & BLEN[3:33:15];
      END;
      CNTCTL+BLEN<1023;%

```

```

%R9038119500
%R9038120000
%R9038120500
%R9038121000
%R9038121500
%R9038122000
%R9038122500
%R9038123000
%R9038123500
      38124000
%R9038124500
%R9038125000
%R9038125500
      38126000
%R9038126500
%R9038127000
%R9038127500
%R9038128000
%R9038128500
%R9038129000
      38129500
      38130000
      38130500
      38131000
      38131500
      38132000
      38132500
      38133000
      38133500
      38134000
      38134500
      38135000
      38135500
      38136000
      38136500
      38137000
      38137500
      38138000
      38138500
      38139000
      38139500
      38140000
      38140500
      38141000
      %DB 38141500
      %DB38142000
%R9038142500
%R9038143000
%R9038143500
%R9038144000
%R9038144500
%R9038145000
%R9038145500
%R9038146000
%R9038146500
%R9038147000
%R9038147500
%R9038148000
%R9038148500
%R9038149000

```

END ELSE%	%R9038149500
IF KIND=9 THEN%	%R9038150000
BEGIN UNLABELED+CNTCTL+1;%	%R9038150500
DIREC+0;%	%R9038151000
END ELSE%	%R9038151500
IF KIND=11 THEN	%R9038152000
BEGIN T1=CIDROW[U=32],[18:15];	%R9038152500
CIDROW[U=32],[18:15]+0;	%R9038153000
DOCARDLABEL;	%R9038153500
FIB[13],[1:9]+NBUFS+1; FIB[13],[10:9]+1;	%R9038154000
IF BLEN<10 THEN BLEN+10;	%R9038154500
END ELSE	%R9038155000
DCN:= FILEMESS("I/O ERR",0,MFID,FID,REEL,CDATE,CYCLE);%	%R9038155500
P(1);	%R9038156000
IF BLEN=0 THEN GO TO DCN;%	%R9038156500
IF NOT FIB[18],[1:1] OR P THEN	%R9038157000
GETBUFFERS(BLEN,NBUFS,U,ALPHA);	%R9038157500
#IND:=	%R9038158000
P(P&RCW[CTC],0,RDS,0,XCH,P&P[CTF],STF);	%R9038158500
END OTHER FILE OPEN IN;	38159000
PROCEDURE OTHERFILEOPENOUT(ALPHA); VALUE ALPHA; INTEGER ALPHA;	38200000
BEGIN REAL RCW=+0,MSCW=-2;	%R9038200100
REAL IOM=IOMASK, IOMASK=+1;	%R9038200200
INTEGER NBUFS=+2,FNUM=+3,BLEN=+4,TYPE=+5,REEL=+6,CDATE=+7,	%R9038200300
CYCLE=+8,MODE=+9,IO=+10,RLEN=+11,U=+12,KIND=+13,	%R9038200400
DIREC=+14,FORMS=+15,COBOL=+16,UNLABELED=+17,	38200500
OPTIONAL=+18,CNTCTL=+19;	38200600
REAL MFID=+20,FID=+21,T1=+22,T2=+23,MASK=+24,STATE=+25;	38200700
ARRAY FIB=+26[*],FPB=+27[*];%	%R9038200800
INTEGER ACCESS=+28,FIB7=+29;	%R9038200900
ARRAY HEADER=+30[*];%	%R9038201000
REAL USASI=NT1, RHEAD=HEADER;	38201100
LABEL LPS,FIND,DCN,PBS;	38201200
SUBROUTINE TYPEOPEN;%	%R9038201400
BEGIN	%R9038201500
T1=(COPNMESS AND ((T1:=JAR[P1MIX,0])>0 OR	38201600
COPNMESS AND T1<0)) OR OPENK;	38201700
NT2:=0;	38201800
IF U<16 THEN	38201900
STREAM(S:=PRNTABLE[U],[30:18], DI=[NT2]);	38202000
BEGIN SI:=LOC S; DS:=8 DEC;	38202100
DI:=DI-7; DS:=6 FILL;	38202200
END;	38202300
FILEMESSAGE((" OUT")&	38202400
TINU[U][6:30:18], NT2, FPB[FNUM], FPB[FNUM+1],	38202500
IF KIND=2 OR KIND=9 THEN P(REEL,CDATE) ELSE P(0,0),	38202600
P,CYCLE,T1);	38202700
END;	%R9038202800
SUBROUTINE REED;%	%R9038203300
BEGIN IF (T2=WAITIO(T1,(MASK OR @40)&@377[CTF],U) AND @367)≠0 THEN	38203400
IF (T2 AND NOT MASK)≠0 THEN	%R9038203500
BEGIN STOPTIMING(FNUM,1023); SETNOTINUSE(U,0);	%R9038203600
FILEMESS("PARITY ", "ON ... "&TINU[U][24:30:18],%	%R9038203700
MFID,FID,REEL,CDATE,CYCLE);%	%R9038203800
END;%	%R9038203900
IF TERMSET(P1MIX) THEN	%R9038204000
BEGIN STOPTIMING(FNUM,1023); SETNOTINUSE(U,0);	%R9038204100
GO TO INITIATE;	%R9038204200
END;	%R9038204300
END REED;%	%R9038204400

```

REAL SUBROUTINE CNTLBITS;%                                %R9038204500
  CNTLBITS=IOMASK&MODE[21:47:1]&DIREC[22:47:1]&CNTCTL[23:47:1]38204600
  &IO[24:47:1]&(KIND=7 OR KIND>9 AND KIND<=12)[20:47:1]R38204700
  &(IF KIND=10R KIND=7OR KIND=12THEN@20ELSE 0)[27:42:6]38204800
SUBROUTINE LABELAREA;%                                    %R9038204900
  M[T1+ALPHA-2]*M OR (GETSPACE((T1+M[T1],[8:10])+4,2,1)+4)%R938205000
  &T1[8:38:10]&CNTLBITS[18:18:15];%                      %R9038205100
%                                                         %R9038205900
  P(RCW,MSCW,STF);                                       %R9038210000
  RCW:=RCW&P(XCH)[CTC];                                   %R9038210500
  IF STATE,[41:1] THEN%                                   %R9038211500
  BEGIN U=FIB[15],[25:5];%                                 %R9038212000
  END ELSE%                                               %R9038212500
  BEGIN T2:=FPB[FNUM+3]; % SAVES COPIES FOR BACK UP      38213000
  IF (U:=FINDOUTPUT(MFID,FID,TYPE                          38213500
  $ SET OMIT = NOT PACKETS                                38214000
  &FPB[FNUM+3][1:23:1]                                    38214500
  $ POP OMIT                                              38215000
  ,FORMS,REEL,CDATE,CYCLE,KIND))>40 THEN                38215500
  BEGIN FIB[14],[3:15]+U; %R9038216000
  FPB[FNUM+2],[18:30]+DATE; %R9038216500
  IF MCP#NOT 0 THEN M[U+2]+USERCODE[P1MIX]; %R9038217000
  M[U+3]+XCLOCK+P(RTR); %R9038217500
  T1:=SPACE(30);                                         38218000
  MOVE(30,U,T1);                                         38218500
  STREAM(DATE,B:=T1+3);                                   38219000
  BEGIN SI:=LOC DATE;DS:=8OCT;DI:=DI-8;DS:=2LIT"+2";END; 38219500
  M[T1+1]+(XCLOCK+P(RTR))&(M[T1+3])[6:30:18];           38220000
  M[T1+4]:= 0&SYSNO[4:46:2]&1[2:47:1];                 38220500
  M[T1+5]+(P(DUP))&1[2:47:1]; %ABORTED PBD 10G.         38221000
  M[T1+6]:=0;                                           38221500
  M[U-1]:=EUF(IF TYPE NEQ 0 AND TYPE LSS 20 THEN         38222000
  "PBD " ELSE "PUD " ,M[U+6],T1-1);                     38222500
  FORGETSPACE(T1);                                       38223000
  FILEMESSAGE((IF TYPE GEQ 20 OR TYPE=0 THEN "PUD...", 38223500
  ELSE "PBD...")&M[U+6][24:6:24],                       38224000
  "OUT " &M[U+6][30:30:18],                             38224500
  MFID,FID,0,0,0,                                       38225000
  (PBDREL OR OPNMESS) OR OPENK);                       38225500
  STARTIMING(FNUM,U+18); %R9038226000
  FPB:=PRT[P1MIX,3]; % STARTIMING MAY HAVE MOVED IT, 38226500
  END ELSE %R9038227000
  BEGIN %R9038227500
  STARTIMING(FNUM,U);% %R9038228000
  FPB:=PRT[P1MIX,3]; % WATCH OUT FOR STARTIMING. 38228500
  TYPEOPEN;% %R9038229000
  IF TYPE=5 OR TYPE=8 OR TYPE=9 THEN UNLABELED+1;% %R9038229500
  IF U<16 THEN BEGIN RRRMECH+TWO(U) OR RRRMECH; %R9038230000
  PRNTABLE[U],[15:15]+ALPHA;% %R9038230500
  END; %R9038231000
  END;% %R9038231500
  IF KIND=6 THEN% %R9038232000
  BEGIN BLEN:=10; %R9038232500
  FIB[18]+(P(DUP))&BLEN[CTC]&BLEN[CTF]&BLEN[3:33:15]; 38233000
  MODE+DIREC+CNTCTL+0;% %R9038233500
  END ELSE% %R9038234000
  IF KIND=1 THEN% %R9038234500
  BEGIN MODE+DIREC+CNTCTL+0;% %R9038235000
  LPS: %R9038235500
  IF NOT COBOL THEN M[ALPHA-2]+0&15[8:38:10];% %R9038236000

```

```

END ELSE%                                %R9038236500
IF KIND=12 THEN                            %R9038237000
BEGIN TYPE=IF (TYPE#0 AND TYPE<20) THEN 15 ELSE 22; 38237500
PBS: MODE=DIREC=0; FIB[13],[1:9]*NBUFS=CNCTCL+1; FIB[13],[10:9]+1; 38238000
      BLEN=IF TYPE#20 THEN 10 ELSE IF BLEN#17 THEN 17 ELSE BLEN; 38238500
      M[T1+GETSPACE(92,3,1)+2]*M[T1-1]+[M[ALPHA]]*(T1+2)[CTF]&%R9038239000
      U[12:42:6]; 38239500
      DISKIO(RHEAD,-T1-77,9,JAR[P1MIX,6],[CF]); 38240000
      M[ALPHA]=T1+2; 38240500
      FIB[14]+(*P(DUP))&(T1+2)[CTC]&(T1+56)[CTF]; %R9038241000
      FIB[18]+(*P(DUP))&BLEN[CTC]&BLEN[CTF]&BLEN[03:33:15]; %R9038241500
      STREAM(D+T1+1); 2(36(DS+8 LIT"0")); %R9038242000
      FIB[5],[FF]+(M[T1+91]*FIB[5],[FF]&1[18:47:1])+1; %R9038242500
      SLEEP([RHEAD],IOMASK); 38243000
      HEADER=[M[T1]]&92[8:38:10]; 38243500
      HEADER[74]*MFID; %R9038244000
      HEADER[75]*FID; %R9038244500
      HEADER[87]*FORMS; %R9038245000
      HEADER[88]=T2,[15:8]; % COPIES 38245500
      %R9038246000
      HEADER[76]*ABS(JAR[P1MIX,0]); %R9038246500
      HEADER[77]*ABS(JAR[P1MIX,1]); %R9038247000
      GO TO LPS; %R9038247500
END ELSE %R9038248000
IF KIND=7 THEN% %R9038248500
BEGIN TYPE=IF (TYPE#0 AND TYPE<20) THEN 6 ELSE 20; 38249000
      IF SVPBT THEN SAVEWORD:=TWO(U) OR SAVEWORD; 38249500
      GO TO PBS; %R9038250000
END ELSE% %R9038250500
IF KIND=2 THEN% %R9038251000
BEGIN IF PRNTABLE[U]#0 THEN GO TO DCN;% %R9038251500
      CNCTCL=MODE;% %R9038252000
END ELSE% %R9038252500
IF KIND=8 THEN% %R9038253000
BEGIN UNLABELED=CNCTCL+1;% %R9038253500
      DIREC=0;% %R9038254000
END;% %R9038254500
IF UNLABELED THEN% %R9038255000
BEGIN IF COBOL THEN% %R9038255500
      BEGIN MASK=0;% %R9038256000
      IF KIND=1 THEN BEGIN T1=@4000100000; REED END ELSE%R938256500
      IF KIND=7 OR KIND=12 THEN %R9038257000
      BEGIN 38257500
      IF TYPE < 20 THEN 38258000
      BEGIN 38258500
      HEADER[73]+@1540176000100000&FIB[5][FTC]; 38259000
      FIB[5],[FF]*FIB[5],[FF]+1; %R9038259500
      FIB[14],[FF]=T1+38; 38260000
      END; 38260500
      GO FIND; 38261000
      END; %R9038261500
      END;% %R9038262000
END ELSE% %R9038262500
BEGIN IF COBOL THEN% %R9038263000
      BEGIN M[ALPHA=2]+P(DUP,LOD)&CNCLBITS[18:18:15];% %R9038263500
      IF U<16 THEN% %R9038264000
      STREAM(N+PRNTABLE[U],[30:18],D+M[ALPHA=2]);% %R9038264500
      BEGIN SI=LOC N; DI=DI+53; DS=5 DEC END;% %R9038265000
      END ELSE% %R9038265500
      BEGIN IF REEL=0 THEN REEL+1;% %R9038266000

```

J

4

```

IF CYCLE=0 THEN CYCLE+1;%                                %R9038266500
IF CDATE=0 THEN STREAM(,CD+(CDATE));%                    %R9038267000
    BEGIN SI+LOC DATE; SI+SI+3; DS+5 OCT END; 38267500
LABELAREA;%                                              %R9038268000
BUILDLABEL(M[ALPHA=2],MFID,FID,REEL,CDATE,CYCLE,%        %R9038268500
    FIB[4],(IF U<16 THEN PRNTABLEIU),[30;18]%%R38269000
    ELSE 0),STATE,[46;2],%                                %R9038269500
    BLEN,RLEN);%                                          %R9038270000
END;%                                                    %R9038270500
M[M[ALPHA=2] INX P(DUP),[8;10]]+@3700000000000000;%     %R9038271000
IF (P(KIND,DUP)=7 OR (P(XCH,DUP)=12 OR P(XCH)=1)) THEN 38271500
IF KIND=7 AND FIB[13],[28;10]#COBOL THEN GO FIND ELSE 38272000
BEGIN IF TYPE GEQ 20 THEN                                38272500
BEGIN M[M[ALPHA=2] INX 4]=FLAG(NABS(JAR[P1MIX,0]));        38273000
M[M[ALPHA=2] INX 5]=FLAG(JAR[P1MIX,1]&17[1;43;5]);        38273500
STREAM(A=[M[M[ALPHA=2] INX 6]]);                          38274000
BEGIN DS:=15 LIT" PUNCH BACK-UP "; DS:=LIT"%";          38274500
    2(DS:=8 LIT"%%%%%%%");                                38275000
END;                                                       38275500
END ELSE                                                 38276000
BEGIN T1=M[M[ALPHA=2] INX 3];                              38276500
DISKWAIT(=(M[ALPHA=2] INX 4),10,JAR[P1MIX,6],[CF]);        38277000
M[M[ALPHA=2] INX 13]+FLAG(NABS(JAR[P1MIX,0]))&0[2;47;1]; 38277500
M[M[ALPHA=2] INX 14]+FLAG(JAR[P1MIX,1]&17[1;43;5]);%R9038278000
M[M[ALPHA=2] INX 3]=T1;                                    38278500
END;                                                       38279000
M[M[ALPHA=2] INX 1]=MFID;                                  38279500
M[M[ALPHA=2] INX 2]=FID;                                    %R9038280000
IF KIND=1 THEN M[ALPHA=2]+P(DUP,LOD)&1[27;42;6] ELSE%R38280500
BEGIN HEADER[73]+FIB[5],[FF]&(TYPE<20)[CTF]&              38281000
    15[3;43;5];                                           38281500
    FIB[5]+P(DUP,LOD,0,1,CFX,+);                          %R9038282000
    STREAM(L+M[ALPHA=2],B+[HEADER[56]]);                  %R9038282500
    BEGIN SI=L; DS+17 WDS END;                             %R9038283000
    FIB[14],[FF]+[HEADER[38]]; GO FIND;                   %R9038283500
END; END;                                                 %R9038284000
T1+NFLAG(M[ALPHA=2]);%                                    %R9038284500
MASK+0; REED;%                                           %R9038285000
IF KIND=2 THEN%                                           %R9038285500
BEGIN T2=@1737000000000000;%                              %R9038286000
    T1+NFLAG([T2]);%                                      %R9038286500
    REED;%                                               %R9038287000
END;%                                                    %R9038287500
END;%                                                    %R9038288000
P(0);                                                     %R9038288500
IF BLEN=0 THEN                                           38289000
DCN: FILEMESS(="I/O ERR",0,MFID,FID,REEL,CDATE,CYCLE); 38289250
IF NOT FIB[16],[1;1] OR P THEN                            %R9038289500
    GETBUFFERS(BLEN,NBUFS,U,ALPHA);                      %R9038290000
FIND:                                                     %R9038290500
    P(P&RCW[CTC],0,RDS,0,XCH,P&P[CTF],STF);             %R9038291000
END OTHER FILE OPEN OUT;                                  38291500
PROCEDURE DISKCLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;% %R9038355000
BEGIN REAL RCW=+0,MSCH=-2;                                %R9038356000
    ARRAY FIB=+1[*],FPB=+2[*],HEADER=+3[*];%           %R9038357000
%% DONT ADD ANY DECLARATIONS BETWEEN "HEADER" AND "KIND" %% WCP%R38358000
INTEGER KIND=+4,NBUFS=+5,U=+6,BLEN=+7,CODE=+8,          %R9038359000
    UNLABELED=+9,COBOL=+10,I=+11,J=+12,FNUM=+13;        38360000
REAL MID=+14,FID=+15,R=+16,D=+17,C=+18,FORMS=+19,STATE=+20; 38362000
LABEL L1,L2,L3,EOF,CLEANUP;                              38363000

```

```

LABEL OBJTYPE, DUMMY;                                %R9038364000
REAL T1=+21,T2=+22,T3=+23,IOD=+24;                   %R9038365000
ARRAY SEGO=+25[*],SKEL=+26[*];                       %R9038366000
REAL T=+27,ACCESS=+28;%
BOOLEAN COMPGO=+29;
$ SET OMIT = NOT SHAREDISK
DEFINE
    LASTLOCK    = HEADER[30]#;
    IMPLOCK     = HEADER[31]#;
    PROGRAMLOCK = M[FPB INX NOT 0]#;
LABEL LSTUNLCK,IMPUNLCK,IOFINI;
SUBROUTINE SDCOOLOFF;
BEGIN
    FOR I=0 STEP 1 UNTIL NBUFS=1 DO,
    BEGIN
        IF NOT M[ALPHA+I],[19:1] THEN
        BEGIN
            IF ((T+M[M[ALPHA+I]]) EQV LASTLOCK)=NOT 0 THEN
            LSTUNLCK: T+LASTLOCK ELSE IF (T EQV IMPLOCK)=NOT 0 THEN
            IMPUNLCK: T+NABS(IMPLOCK) ELSE
            BEGIN
                SLEEP([M[ALPHA+I]],IOMASK); GO TO IOFINI;
            END; T1+T2+0;
            FOR J=0 STEP 1 UNTIL (LQAVAIL=1) DO
            IF ((T3+LQUEE[J]),[8:40] EQV T,[8:40])=NOT 0 THEN
            IF LOCATQUEE[T3,[1:7]],[3:5]=P1MIX THEN
            BEGIN
                IF J < (LQAVAIL+LQAVAIL-1) THEN
                STREAM(A+LQAVAIL-J,B+[LQUEE[J]]);
                BEGIN SI+B; SI+SI+8; DS+A WDS END;
                RETURNIOSPACE(T3,[1:7]);
                T1+1;
            END ELSE T2+1;
            IF NOT (T1 AND T2) THEN
            BEGIN
                IF NOT T1 THEN
                BEGIN
                    IF I<NBUFS THEN
                    SLEEP([M[ALPHA+I]],IOMASK);
                    T1+ABS(T) OR @2060; % UNLOCK ADDRESS
                END ELSE
                T1+(ABS(T) OR @60)&SYSNO[30:46:2]; % CLEAR ADR
                P(WAITIO([T1] INX @100000000,0,18),DEL);
            END;
            IF T GEQ 0 THEN PROGRAMLOCK+LASTLOCK+0 ELSE
            IMPLOCK+0;
        END;
    IOFINI:
    END;
    IF LASTLOCK#0 THEN GO TO LSTUNLCK;
    IF IMPLOCK#0 THEN GO TO IMPUNLCK;
END SD COOLOFF;
$ POP OMIT
SUBROUTINE COOLOFF;
BEGIN FOR I=0 STEP 1 UNTIL NBUFS=1 DO%
    BEGIN IF NOT M[ALPHA+I],[19:1] THEN%
        SLEEP([M[ALPHA+I]],IOMASK);%
        IF KIND#4 THEN
        IF M[ALPHA+I],[27:1] THEN GO TO EOF;%
    END;%

```

```

38366010
38366020
38366099
38366100
38366200
38366300
38366400
38366500
38366600
38366700
38366800
38366900
38367000
38367100
38367200
38367300
38367400
38367500
38367600
38367700
38367800
38367900
38368000
38368100
38368200
38368300
38368400
38368500
38368600
38368700
38368800
38368900
38369000
38369100
38369200
38369300
38369400
38369500
38369600
38369700
38369800
38369900
38370000
38370100
38370200
38370300
38370400
38370500
38370600
38370601
38370700
38370800
%R9038371000
%R9038372000
%R9038373000
%R9038374000
%R9038375000

```



```

EOF: END COOLOFF; % %R9038376000
% 38376500
BOOLEAN SUBROUTINE WRITTENON; % PICKS UP THE ACCESSED BITS FROM 38377000
BEGIN J:=0; % THE BUFFERS, 38377200
IF (T:=FIB[10],[3:15]) NEQ 0 THEN 38377400
BEGIN 38377600
FOR I:=NBUFS-1 STEP -1 UNTIL 0 DO 38377800
IF M[I],[11:1] THEN J:=I:=1 ELSE T:=M[T],[FF]-2; 38378000
END; 38378200
WRITTENON:=J; 38378400
END; 38378600
% 38379000
DEFINE REW=CODE,[47:1]#,% %R9038380000
KRUNCH=NOT CODE,[42:1]#,% %R9038381000
REL=CODE,[46:1]#,% %R9038382000
TIME=CODE,[45:1]#,% %R9038383000
LOCK=NOT CODE,[44:1]#,% %R9038384000
PURGE=NOT CODE,[43:1]#;% %R9038385000
DEFINE TECH=STATE,[46:2]#,% OPENIO=FIB[13],[22:1]#,% 38385400
WRITBACK=FIB[13],[23:1]#,% LASTIO=FIB[13],[46:1]#,% 38385500
WRITEAFTEROFF=FIB[13],[44:2]#,% INPUT=STATE,[43:1]#;% 38385600
%R9038386000
% START OF CODE 38386010
% 38386020
P(RCW,MSCW,STF); RCW ← RCW & P(XCH)[CTC]; 38387000
HEADER ← FIB[14]; ACCESS ← FIB[4],[27:3]; 38388000
IF COBOL THEN 38389000
BEGIN IF COBOL > 0 THEN % COBOL 61 38389100
BEGIN IF WRITBACK AND TECH=0 AND LASTIO AND 38389200
(OPENIO OR NOT(INPUT)) THEN 38389300
IF ACCESS=1 AND WRITEAFTEROFF≠0 THEN 38389400
BEGIN FIB[7] ← *P(DUP) - 1; 38389500
HEADER[7] ← *P(DUP) - 1; 38389600
END ELSE WRITEAFTEROFF ← 0; 38389700
IF TECH=0 THEN IF WRITEAFTEROFF=2 THEN 38389800
BEGIN FIB[7] ← *P(DUP) + 1; 38389900
HEADER[7] ← *P(DUP) + 1; 38390000
END ELSE IF WRITEAFTEROFF=1 THEN 38390100
BEGIN FIB[7] ← *P(DUP) - 1; 38390200
HEADER[7] ← *P(DUP) - 1; 38390300
END; 38390400
WRITEAFTEROFF ← 0; 38390500
END; 38391000
IF ACCESS=1 THEN % IF RANDOM 38391100
BEGIN IF COBOL > 0 THEN % COBOL61 38391200
BEGIN ACCESS ← 4; 38391250
IF FIB[13],[10:9] = 2 THEN % SEEK IN PROCESS 38391300
BEGIN 38391350
% SET OMIT = NOT SHAREDISK 38391399
IF FPB[FNUM+3],[43:5]=26 THEN 38391400
SDCOOLOFF ELSE 38391450
% POP OMIT 38391451
COOLOFF; FIB[13],[10:9] ← 1; 38391500
END 38391550
END ELSE IF FIB[17]<BLEN THEN ACCESS←4; % COBOL68 38391600
END; 38391700
IF FIB[13],[23:1] AND ACCESS=0 THEN 38391800
BEGIN FIB[7]←P(DUP,LOD)=1; 38391900
ACCESS←4; 38391910
END; END; 38391950

```

```

IF NOT STATE,[41:1] THEN%
BEGIN IF ACCESS=1 THEN%
BEGIN
$ SET OMIT = NOT SHAREDISK
IF FPB[FNUM+3],[43:5]=26 THEN
SDCOOLOFF ELSE
$ POP OMIT
COOLOFF;
END ELSE%
IF ACCESS=0 THEN%
BEGIN COOLOFF; IF NOT STATE,[43:1] THEN%
IF FIB[17]<BLEN AND STATE,[46:2]≠0 THEN%
BEGIN R := SPACE(((BLEN+29) DIV 30)×30 + 1);
IF (M[R]+M[FIB[16]])+%
DISKADDRESS(MID,FID,FPB[FNUM+3],FIB[7]-1,HEADER,0)) NEW 0 THEN % (SHM)
BEGIN
P(WAITIO(FIB[16]&1[24:47:1]&R[33:33:15],%
0,U),DEL);%
MOVE(FIB[17],R+BLEN-FIB[17]+1,%
FIB[16] INX BLEN-FIB[17]+1);%
P(WAITIO(FIB[16],0,U),DEL);%
IF NOT FIB[16],[24:1] THEN HEADER[4],[11:1]+1;
END;
FORGETSPACE(R);%
END;%
END ELSE%
BEGIN
$ SET OMIT = NOT SHAREDISK
IF FPB[FNUM+3],[43:5]=26 THEN
SDCOOLOFF ELSE
$ POP OMIT
COOLOFF;
IF (FIB[17] LSS BLEN AND STATE,[46:2]≠0) OR
ACCESS=4 THEN
BEGIN IF ACCESS=4 THEN
IF FIB[13],[23:1] OR NOT STATE,[43:1] THEN
ACCESS := 2;
IF (M[FIB[16]]:=DISKADDRESS(MID,FID,FPB[FNUM+3],FIB[7], % (SHM)
HEADER,0))=0 THEN ACCESS := 4;
IF ACCESS≠4 THEN
BEGIN P(WAITIO(FIB[16]&0[24:24:1],0,U),DEL);
HEADER[4],[11:1]+1; END;
END; IF ACCESS = 4 THEN ACCESS := 2;
END;%
END;%
HEADER[4],[43:1]:=FPB[FNUM+3],[15:1];
IF (NOT REW) OR LOCK OR REL OR TIME THEN
BEGIN
FORMS=HEADER[3];
STREAM(PF+[FIB[4]],D+FPB[FNUM+2],[18:30],H+[HEADER[3]],S+[T]);
BEGIN SI=PF;SI=SI+5;DS=3 OCT;SI=LOC D;DI=H;DS=8 OCT END;
HEADER[3]+(P(DUP,LOD,SSN))&(P(DUP))[12:30:18]&T[2:38:10];
END;
IF LOCK OR HEADER[4],[43:1] THEN
BEGIN IF NOT HEADER[4] THEN % FILE IS BEING CREATED
BEGIN
IF KRUNCH THEN KRUNCHER(HEADER);
HEADER[4],[9:3]=5;% MARK AS NEW FORMAT,ACCESSED
IF JAR[P1MIX,0] < 0 AND FIB[4],[29:1] THEN
% COMPILER CLOSING CODE FILE WITH LOCK *****

```

```

XR9038392000
XR9038393000
38394000
38394099
38394100
38394200
38394201
38394300
XR9038395000
XR9038396000
XR9038397000
XR9038398000
38399000
XR9038400000
% (SHM)38401000
38401500
XR9038402000
XR9038403000
XR9038404000
XR9038405000
XR9038406000
38406500
38407000
XR9038408000
XR9038409000
XR9038410000
38411000
38411009
38411010
38411020
38411021
38411030
38411030
38411500
38412000
38412500
38413000
38413500
% (SHM)38414000
38414200
38414400
38414500
38414600
38414800
XR9038415000
XR9038416000
38417000
XR9038419000
XR9038420000
XR9038421000
XR9038422000
XR9038423000
XR9038424000
XR9038425000
38426000
38427000
XR9038428000
XR9038429000
XR9038430000
XR9038431000
*****38432000

```

```

BEGIN
SEGO:=[MIGETSPACE(62,2,5)+2]&30[8:38:10];          38434000
SKEL ← 31 INX SEGO; T3 ← JAR[P1MIX,2],[FF]; %R9038435000
% READ IN SEGMENT ZERO                               %R9038436000
DISKWAIT(=SEGO,[CF],30,HEADER[10]);                 %R9038437000
% READ IN SKELETON SHEET                             %R9038438000
DISKWAIT(=SKEL,[CF],30,T3);                         %R9038439000
IF SKEL[20]<0 THEN SKEL[20] ← SEGO[7],[FF]; %R9038440000
IF JAR[P1MIX,2],[8:10]=1 THEN %R9038441000
  BEGIN % COMPILE AND GO *****38442000
  DISKWAIT(SKEL,[CF],30,T3); %R9038443000
  COMPGO ← TRUE; %R9038444000
  END %R9038445000
ELSE %R9038446000
  BEGIN % COMPILE TO LIBRARY *****38447000
  FOR T1 ← 15 STEP 1 UNTIL 22 DO %R9038448000
    SEGO[T1] ← SKEL[T1]; %R9038449000
    IF (T2 ← SKEL[13]) = 0 THEN GO TO L3; %R9038450000
    SKEL[13] ← 0; % IN CASE I CALL TERMINATE %R938451000
    DISKWAIT(SKEL,[CF],30,T3); %R9038452000
IF (T1:=DISKADDRESS(MID,FID,FPB[FNUM+3],HEADER[7])= % (SHM)38453000
  (*P(DUP))+1,HEADER,0))=0 THEN 38454000
  FILEMESS(="DISK ", "OVRFLOW",MID,FID,38455000
  R,D,C); %R9038456000
    SEGO[15] ← T1 ← HEADER[7]; %R9038457000
L1: DISKWAIT(=SKEL,[CF],30,T2); %R9038458000
    FORGETESPDISK(T2); %R9038459000
    IF (T2←SKEL[29]) = 0 THEN GO TO L2; %R9038460000
IF (T3:=DISKADDRESS(MID,FID,FPB[FNUM+3], % (SHM)38461000
  HEADER[7])=(*P(DUP))+1,
  HEADER,0))=0 THEN 38461500
  FILEMESS(="DISK ", "OVRFLOW",MID,FID,38463000
  R,D,C); %R9038464000
    SKEL[29] ← T3 ← HEADER[7]; %R9038465000
    DISKWAIT(SKEL,[CF],30, %R9038466000
  I←HEADER[T1 DIV HEADER[8]+10] +
  T1 MOD HEADER[8]); %R9038467000
    T1 ← T3; %R9038468000
    GO TO L1; %R9038469000
L2: DISKWAIT(SKEL,[CF],30, %R9038470000
  I←HEADER[T1 DIV HEADER[8]+10] +
  T1 MOD HEADER[8]); %R9038471000
L3: SEGO[6] ← P(DUP,LOD,SSN); % "NEW FORMAT" %R938474000
    HEADER[4],[10:1]+1;%MARK AS PROGRAM FILE %R938475000
    DISKWAIT(SEGO,[CF],30,HEADER[10]); %R9038476000
    END COPY OF LABEL EQUATION CARDS; %R9038477000
    FORGETSPACE(SEGO); %R9038478000
    IF HEADER[7]<HEADER[8]=1 THEN %R9038479000
    BEGIN FORGETUSERDISK(HEADER[10]+HEADER[7]+1,%R38480000
  HEADER[7]-HEADER[8]+1); 38481000
  HEADER[8] ← HEADER[7]+1; %R9038482000
    END; %R9038483000
    FOR T1:=1 STEP 1 UNTIL 4 DO %R9038484000
    IF P(,OBJTYPE,T1,+,LOD) = %R9038485000
    ABS(JAR[P1MIX,0]) THEN %R9038486000
    HEADER[4],[36:6]:=T1+2; %R9038487000
    END CODE FILE; 38488000
    HEADER[1]←FORMS&HEADER[3][6:30:18]; %R9038489000
    IF (HEADER[2]:=USERCODE[P1MIX]),[1:1] THEN %R9038490000
    HEADER[2]:=0; %R9038491000

```

DUMMY :

<pre> %       HEADER[5] := HEADER[6] := 0;       IF COMPGO THEN         BEGIN PRT[P1MIX,@26]+I0D+GETESPDISK;           DISKWAIT(HEADER,[CF],30,I0D);         END ELSE         BEGIN           ENTERUSERFILE(MID,FID,HEADER,[CF]-1);         END;       END;%     END;%     IF REW AND NOT(LOCK OR REL OR TIME) THEN     BEGIN       IF HEADER[4] THEN         IF WRITTENON THEN HEADER[4],[11:1]:=1;         STATE,[39:4]:=2;       END ELSE       BEGIN         HEADER[1]+FORMS&amp;HEADER[3][6:30:18];         IF HEADER[4] THEN % FILE IS ALREADY IN DIRECTORY         BEGIN           J:=WRITTENON OR HEADER[4],[11:1];           % SET OMIT = SHAREDISK           I:=IF FIB[5],[1:1] OR NOT J THEN FIB[5],[13:3]+10 ELSE             (HEADER INX 0)&amp;FIB[5][30:13:3];           % POP OMIT           % SET OMIT = NOT SHAREDISK           I:=((FPB[FNUM+3] AND 31)=26)&amp;FIB[5][30:13:3];           I:=IF FIB[5],[1:1] OR NOT J THEN IF I THEN 22 ELSE             I,[FF]+10 ELSE (HEADER INX 0)&amp;(IF I THEN 12               ELSE I,[FF])[CTF];           % POP OMIT           IF(I+DIRECTORYSEARCH(MID,FID&amp;J[3:47:1],I))#0 THEN             IF PURGE THEN               IF M[I+4],[12:4]=0 THEN                 IF NOT SYSTEMFILE(MID,FID) THEN                   IF SECURITYCHECK(MID,FID,USERCODE[P1MIX],I),[45:1] THEN%R938513000                     P(DIRECTORYSEARCH(=MID,FID,7),DEL);                   IF I#0 THEN FORGETSPACE(I);                 END ELSE%                 IF NOT LOCK THEN%                 IF HEADER[4],[43:1] THEN P(DIRECTORYSEARCH(=MID,FID,7),DEL) ELSE                 BEGIN                   HEADER[2]+USERCODE[P1MIX];                   DISKLOG(MID,FID,HEADER);                   FOR I=10 STEP 1 UNTIL 29 DO%                   IF HEADER[1]#0 THEN FORGETUSERDISK(HEADER[I],=HEADER[8]);%                 END;                 FORGETSPACE(HEADER);                 STATE,[39:4]-1;%               END;               IF NOT COBOL THEN FIB[4],[27:3]-3;               GO CLEANUP;%             OBJTYPE!!! "BASIC ", %1%               "ALGOL ", %2%               "COBOL ", %3%               "FORTRAN", %4%               "TSPOL ", %5%               "XALGOL ", %6% </pre>	<pre> %R9038492000 %R9038493000 %R9038494000 %R9038495000 %R9038496000 %R9038497000 %R9038498000 %R9038499000 %R9038500000 %R9038501000 %R9038502000 38503000 38503200 38503400 38503600 38503800 38504000 38504500 %R9038505000 38506000 38507000 38507500 38507799 38508000 38508500 38508501 38508599 38508600 38508700 38508800 38508900 38508901 38509000 38510000 38511000 38512000 %R938513000 %R9038514000 38515000 38516000 %R9038517000 %R9038518000 38518500 %R9038519000 %R9038520000 %R9038521000 %R9038522000 %R938523000 %R9038524000 %R9038525000 %R9038526000 38527000 %R9038528000 %R9038529000 %R9038530000 %R9038531000 %R9038532000 %R9038533000 %R9038534000 %R9038535000 </pre>
--	--

```

O; %DUMMY%                                %R9038536000
CLEANUP;                                    %R9038537000
P(P&RCW[CTC],0,RDS,0,XCH,P&P[CTF],STF);    %R9038538000
END DISK CLOSE;                             %R9038539000
PROCEDURE BACKCLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;%
BEGIN REAL RCW=+0, MSCW=-2;                 %R9038540000
ARRAY FIB=+1[*],FPB=+2[*],HEADER=+3[*];%    %R9038541000
%%% DONT ADD ANY DECLARATIONS BETWEEN "HEADER" AND "KIND" %%% WCP%R38543000
INTEGER KIND=+4,NBUFS=+5,U=+6,BLEN=+7,CODE=+8,%R9038544000
UNLABELED=+9,COBOL=+10,I=+11,J=+12,FNUM=+13; 38545000
REAL MID=+14,FID=+15,R=+16,D=+17,C=+18,FORMS=+19,STATE=+20; 38547000
LABEL AGAIN,EOF,EOT,CLOSEOUT,PBD,PUD;      38548000
REAL T1=+21,T2=+22,T3=+23,IOD=+24;         %R9038549000
ARRAY SEGO=+25[*],SKEL=+26[*];             %R9038550000
%                                             %R9038551000
SUBROUTINE COOLOFF;%                         %R9038552000
BEGIN FOR I=0 STEP 1 UNTIL NBUFS-1 DO%      %R9038553000
  BEGIN IF NOT M[ALPHA+I],[19:1] THEN%      %R9038554000
    SLEEP([M[ALPHA+I]],IOMASK);%           %R9038555000
    IF KIND#4 THEN                          %R9038556000
      IF M[ALPHA+I],[27:1] THEN GO TO EOF;% %R9038557000
    END;%                                     %R9038558000
EOF: END COOLOFF;%                           %R9038559000
%                                             %R9038560000
REAL T=+27,ACCESS=+28;%                     %R9038561000
BOOLEAN COMPGO=+29;                          %R9038562000
REAL TYPE=+30;                                38562100
DEFINE REW=CODE,[47:1]#,%                    %R9038563000
REL=CODE,[46:1]#,%                           %R9038564000
TIME=CODE,[45:1]#,%                          %R9038565000
LOCK=NOT CODE,[44:1]#,%                      %R9038566000
PURGE=NOT CODE,[43:1]#;%                     %R9038567000
$ SET OMIT = PACKETS                         38567950
DEFINE TOREELNO = 33:33:15#;                 38568000
$ SET OMIT = NOT PACKETS                     38568050
DEFINE TOREELNO = 42:42:6#;                 38568100
$ POP OMIT OMIT                              38568150
%                                             %R9038569000
SUBROUTINE CKBKUP;                           %R9038570000
BEGIN M[M[ALPHA]INX 17 ]+M[ALPHA]&(FIB[5]   ) [FTC]; 38571000
  FIB[5]+P(DUP,LOD,0,1,CFX,+);              %R9038572000
  IF NOT PRTRW[P1MIX],[7:1] THEN             38573000
    IF FIB[14],[CF]=FIB[14],[FF]           38573100
  THEN BEGIN PBIO(ALPHA,FIB[14]);SLEEP([M[ALPHA]],IOMASK)END ELSE 38574000
  BEGIN; STREAM(S+ M[ALPHA],Z+FIB[14],[FF]); %R9038575000
  BEGIN SI+S; DS+18 WDS END;                %R9038576000
  FIB[14],[FF]+P(DUP),[FF]=18;             %R9038577000
END; END;                                    %R9038578000
P(RCW,MSCW,STF);                             %R9038580000
RCW:=RCW&P(XCH)[CTC];                        %R9038581000
J=LOCK;                                       38581100
IF T1<(FIB[9],[1:1] AND KIND=7) THEN % MULTI=REEL PBT FILE 38581200
BEGIN                                         38581300
  FIB[9],[1:1]+0;                            38581400
  COOLOFF;                                    38581500
  GO TO EOT;                                  38581600
END;                                          38581700
IF FIB[17]<0 THEN                             38582000
BEGIN M[ALPHA],[FF]+@60020; IF TYPE<20 THEN CKBKUP; 38583000
M[ALPHA],[18:1]+0; CKBKUP END%              %R9038584000

```

```

ELSE IF FIB[17]<RLEN THEN%
BEGIN IF NOT COBOL THEN FIB[17]*FIB[17]=(STATE,[46:2]=3);%
STREAM(N:=FIB[17],D:=M[ALPHA],[CF]);
BEGIN N(DS*8 LIT " "); END;
M[ALPHA]*FLAG(FIB[16]&0[20:47:1]); CKBKUP;
END ELSE COOLOFF;
M[ALPHA]*(P(DUP))&(@60000)[CTF]&(TYPE<20)[32:47:1];
IF NOT UNLABELED THEN
BEGIN IF TYPE<20 THEN CKBKUP;
M[ALPHA]*(P(DUP))&2[18:45:3]&M[ALPHA=2][8:8:10];
STREAM(L*M[ALPHA=2],B*M[ALPHA]); BEGIN SI=L; DS=17 WDS END;
END; M[ALPHA],[20:1]+1; IF FIB[14],[FF]#FIB[14],[CF] THEN
BEGIN CKBKUP; FIB[14],[FF]*P(DUP); END; CKBKUP;
IF KIND=12 THEN % PBD
BEGIN T=FIB[14],[3:15];
IF(R:=M[T+7]*3)#0 THEN %PRESUMABLY 0 IMPLIES NO USER DISK
BEGIN
IF R < PBDROWSZ THEN
BEGIN FORGETUSERDISK(M[T+10]+R,PBDROWSZ=R+1);
M[T+8]+R;
END;
M[T+1]+M[T+3];
STREAM(A+FPB[FNUM+2],[18:30],T+T+3);
BEGIN SI=LOC A; DS=8 OCT; DI=DI-8; DS+2 LIT"+2";
SI+T; SI+SI+5; DS+3 CHR;
END;
M[T+1],[6:18]+M[T+3],[30:18];
IF I:=TYPE>20 THEN M[T+5],[3:1]=0;
M[T+5],[2:1]+0;
DISKWAIT(T,30,M[T-1]);
R=M[T+6];
J:=R&1[TOREELNO];
P(DIRECTORYSEARCH(=(IF I THEN P(PUD) ELSE P(PBD)),J,14),
DEL);
IF J#R THEN
BEGIN STREAM(ONE:=1, D:=J);
BEGIN SI:=LOC ONE; DS:=8 ADD END;
GO AGAIN;
END;
% SET OMIT = NOT PACKETS
IF (T1:=PSEUDOMIX[P1MIX])=0 THEN P(1) ELSE
IF PACKETPAGE[T1-32]=0 THEN P(1) ELSE P(0);
IF P THEN
BEGIN PBCOUNT:=PBCOUNT+1;
% POP OMIT
IF AUTOPRINT
THEN P(PRINTORPUNCHWAIT(R&1[32:32:16],I),DEL);
% SET OMIT = NOT PACKETS
END;
% POP OMIT
T:=CLOSEK OR (PBDREL OR CLOSEMESS);
FILEMESSAGE((IF I THEN P(PUD) ELSE P(PBD))&R[24:6:24],
("REL ")&R[30:30:18],MID,FID,
FIB[7],0,0,T);
END;
FORGETSPACE(FIB[14],[3:15]); FIB[14],[3:15]=0;
ELSE
END
EOT: BEGIN T=@1737000000000000000;
J+WAITIO([T],@40,U)#0 OR J;
I I=SPACE(8);

```

```

%R9038585000
%R9038586000
38587000
%AI38587500
%R9038588000
%R9038589000
38590000
%R9038591000
38591100
38592000
38593000
%R9038594000
%R9038595000
%R9038596000
%R9038597000
38597500
38598000
38598100
38599000
%R9038600000
%R9038601000
%R9038602000
%R9038603000
%R9038604000
%R9038605000
%R9038606000
%R9038607000
38607100
38607950
38608000
38608050
38608100
38608200
38608300
38608400
38608500
38608600
38608700
38608800
38608890
38608900
38608910
38608920
38609000
38609010
38609100
38609300
38609590
38609600
38609610
38617000
38618000
38618500
38619000
38621500
38621600
%R9038622000
38623000
%R9038624000
%R9038625000

```

```

STREAM(PN:=TYPE GEQ 20,D:=0,I); 38626000
BEGIN 38626100
    DS:=24LIT" LABEL OPBTMCP OBACK=UP"; 38626200
    PN(D:=DI; DI:=DI-14; DS:=2LIT"UT"; DI:=D); 38626300
    20(DS:=2LIT" "); 38626400
END; 38626500
IF NOT UNLABELED THEN M[I+4]+M[M[ALPHA=2] INX 4],[42:6]; %R9038628000
M[I+3]+T1; % MARK ENDING TAPE LABEL FOR MULTI-REEL COND. 38628100
J*WAITIO(I&8[8:38:10]&5[21:45:3],@40,U)#0 OR J;% %R9038629000
FORGETSPACE(I);% %R9038630000
FOR I=0 STEP 1 UNTIL 1 DO% %R9038631000
    P(WAITIO(@1000000340000005,@40,U),DEL);% %R9038632000
IF (TWO(U) AND SAVEDWORD)#0 THEN% %R9038633000
    SETNOTINUSE(U,0) ELSE %R9038634000
BEGIN% %R9038635000
RDCTABLE[U]+(*P(DUP))&0[8:8:6]&R[14:38:10]; 38637000
PRNTABLE[U],[15:15]+0; 38638000
RRRMECH:=NOT TWO(U) AND RRRMECH; 38638500
I:= IF (AUTOPRINT AND R=1) THEN NOT 38639000
    PRINTORPUNCHWAIT(-U,TYPE GEQ 20) AND 1 ELSE 1; 38639100
IF I THEN IF J THEN SETNOTINUSE(U,0) ELSE LABELTABLE[U],[1:5]:=1;%R38640000
END; END; %R9038641000
STATE,[FF]+0; %R9038642000
GO CLOSEOUT;% %R9038643000
PBD:;; "PBD "; %R9038644000
PUD:;; "PUD "; 38644500
CLOSEOUT; %R9038645000
P(P&RCW[CTC],0,RDS,0,XCH,P&P[CTF],STF); %R9038646000
END BACK CLOSE; %R9038647000
PROCEDURE OTHERCLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;% %R9038648000
BEGIN REAL RCW=+0,MSCW=-2; %R9038649000
ARRAY FIB=+1[*],FPB=+2[*],HEADER=+3[*];% %R9038650000
*** DONT ADD ANY DECLARATIONS BETWEEN "HEADER" AND "KIND" *** WCP%R38651000
INTEGER KIND=+4,NBUFS=+5,U=+6,BLEN=+7,CODE=+8, %R9038652000
UNLABELED=+9,COBOL=+10,I=+11,J=+12,FNUM=+13; 38653000
REAL MID=+14,FID=+15,R=+16,D=+17,C=+18,FORMS=+19,STATE=+20; 38655000
REAL T1=+21,T2=+22,T3=+23,IOD=+24; %R9038656000
ARRAY SEGO=+25[*],SKEL=+26[*]; %R9038657000
REAL T=+27,ACCESS=+28;% %R9038658000
BOOLEAN COMPGO=+29; %R9038659000
LABEL PX,PBD; %R9038660000
LABEL CR,LP,MT,CLOSED,DK,SP,CP,BKUP,PP,PR,DC,CD,CC; %R9038661000
SWITCH SW=CR,LP,MT,CLOSED,DK,SP,CP,BKUP,PP,PR,DC,CD,BKUP; %R9038662000
LABEL EOF,ON,DNE,CLEANUP;% %R9038663000
LABEL EOD; %R9038664000
SUBROUTINE COOLOFF;% %R9038665000
BEGIN FOR I=0 STEP 1 UNTIL NBUFS=1 DO% %R9038666000
    BEGIN IF NOT M[ALPHA+I],[19:1] THEN% %R9038667000
        SLEEP([M[ALPHA+I]],IOMASK);% %R9038668000
        IF KIND#4 THEN %R9038669000
            IF M[ALPHA+I],[27:1] THEN GO TO EOF;% %R9038670000
    END;% %R9038671000
EOF; END COOLOFF;% %R9038672000
% %R9038673000
DEFINE REW=CODE,[47:1]#,% %R9038674000
REL=CODE,[46:1]#,% %R9038675000
TIME=CODE,[45:1]#,% %R9038676000
LOCK=NOT CODE,[44:1]#,% %R9038677000
PURGE=NOT CODE,[43:1]#;% %R9038678000
% %R9038679000

```

```

SUBROUTINE EMPTY;%
IF FIB[17]<BLEN AND (STATE,[46:2]≠0 OR KIND=1) THEN
BEGIN IF NOT COBOL THEN FIB[17]*FIB[17]=(STATE,[46:2]=3);%
STREAM(KIND,N:=FIB[17],D:=M[ALPHA],[CF]);
BEGIN SI*LOC KIND; SI*SI+7;%
IF SC="2" THEN DS*LIT "+" ELSE%
IF SC="5" THEN DS*LIT "-" ELSE N(DS*8 LIT " ");%
END;%
P(WAITIO(FIB[16]&(BLEN=FIB[17])*(KIND=2))[8:38:10]%
,@40,U),DEL);%
FIB[6]*FIB[6]+1;
END ELSE COOLOFF;%
LABEL CLOSEOUT;%
LABEL EOFIT;%
%
P(RCW,MSCW,STF);
RCW:=RCW&P(XCH)[CTC];
GO TO SW(KIND);
%
CR: COOLOFF; BLASTQ(U);%
IF I≥NBUFS THEN DO UNTIL WAITIO(M[ALPHA=2],@40,U)≠0 ELSE%
BEGIN I*M[ALPHA+1],[33:15];%
T=FIB[16],[33:15]-2;%
FOR J*1 STEP 1 UNTIL NBUFS DO%
BEGIN IF (I>T) AND (I≤(T+BLEN+1)) THEN GO ON;%
T*M[T],[18:15]-2;%
END;%
ON: MOVE(10,T+2,M[ALPHA=2]);%
END;%
IF JAR[P1MIX,0]<0 THEN%
IF PRT[P1MIX,@25]≠0 THEN%
DNE: BEGIN STREAM(I; E="ENDPACK", D=M[ALPHA=2]);%
BEGIN SI*D;%
L: SI*SI+1; IF SC=" " THEN GO TO L;%
DI*LOC E; DI*DI+1;
IF 3 SC=DC THEN TALLY+1;
% SET OMIT = NOT(PACKETS)
IF TOGGLE THEN ELSE
% POP OMIT
BEGIN SI*SI-3; IF 4 SC=DC THEN TALLY+1; END;
I*TALLY;%
END;%
IF NOT P THEN%
BEGIN BLASTQ(U);%
DO UNTIL WAITIO(M[ALPHA=2],@40,U)≠0;%
GO TO DNE;%
END;%
END;%
BLASTQ(U);
CC: NT3:=GETSPACE(13,64,5)+4;
MOVE(10,M[ALPHA=2],NT3);
FORGETSPACE(M[ALPHA=2] INX NOT 1);
M[ALPHA=2]*NT3;
LABELTABLE[U]*@14;
RDCTABLE[U]*0;
M[NT3 INX 10]*UNITCODE[U=23];
FREECARD((M[ALPHA=2],[CF])&U[3:43:5]&JAR[P1MIX,6][1:1:1]);
GO CLOSEOUT;%
%

```

```

%R9038680000
%R9038681000
%R9038682000
38683000
%R9038684000
%R9038685000
%R9038686000
%R9038687000
%R9038688000
%R9038689000
%R9038690000
%R9038691000
%R9038692000
%R9038693000
%R9038694000
%R9038695000
%R9038696000
%R9038697000
%R9038698000
%R9038699000
%R9038700000
%R9038701000
%R9038702000
%R9038703000
%R9038704000
%R9038705000
%R9038706000
%R9038707000
%R9038708000
%R9038709000
%R9038710000
38711000
%R9038712000
%R9038713000
38714000
38715000
38715099
38715100
38715200
38715201
%R9038716000
%R9038717000
%R9038718000
%R9038719000
%R9038720000
%R9038721000
%R9038722000
%R9038723000
%R9038724000
%R9038725000
38726000
%R9038727000
%R9038728000
%R9038729000
%R9038730000
%R9038731000
38731100
38732000
%R9038733000
%R9038734000

```



```

CP:  EMPTY;%                               %R9038735000
      IF NOT UNLABELED THEN P(WAITIO(M[ALPHA=2],0,U),DEL);%   %R9038736000
      SETNOTINUSE(U,FORMS OR PUNCHLCK);                       38737000
      GO CLOSEOUT;%                                           %R9038738000
%                                                                 %R9038739000
LP:  EMPTY;%                               %R9038740000
      P(WAITIO(@4000100000,0,U),DEL);%                         %R9038741000
      IF NOT UNLABELED THEN P(WAITIO(M[ALPHA=2],0,U),DEL);%   %R9038742000
      SETNOTINUSE(U,FORMS);                                    %R9038743000
      GO CLOSEOUT;%                                           %R9038744000
%                                                                 %R9038745000
SP:  IF STATE,[43:1] THEN COOLOFF ELSE EMPTY;%               %R9038746000
      GO CLOSEOUT;%                                           %R9038747000
%                                                                 %R9038748000
MT:  IF NOT STATE,[41:1] THEN%                               %R9038749000
      BEGIN IF STATE,[43:1] THEN%                             %R9038750000
          BEGIN COOLOFF; BLASTQ(U);%                          %R9038751000
              IF NOT REW THEN                                  38752000
                  BEGIN T=@1000000140000005&STATE[22:44:1];% %R9038753000
                      IF I≥NBUFS THEN SKIPFILE(U,T,[22:1]);% %R9038754000
                      IF NOT UNLABELED THEN                    38754100
                          P(WAITIO(T,@377,U),DEL);%           %R9038755000
                      END;%                                     %R9038756000
          END ELSE%                                           %R9038757000
          BEGIN EMPTY;%                                       %R9038758000
              EOFIT: T=@173700000000000000;%                 %R9038759000
                  P(WAITIO([T],@40,U),DEL);%                   %R9038760000
                  IF NOT UNLABELED THEN%                       %R9038761000
                      BEGIN;STREAM(BC+FIB[6],RC+FIB[7],D=M[ALPHA=2]);% %R9038762000
                          BEGIN SI=LOC BC; DI=DI+40;%          %R9038763000
                              DS=5 DEC; DS=7 DEC;%             %R9038764000
                          END;%                                  %R9038765000
                          P(WAITIO(M[ALPHA=2],@40,U),DEL);%   %R9038766000
                          P(WAITIO([T],@40,U),DEL);%         %R9038767000
                          T=@10000003400000005;%            %R9038768000
                          P(WAITIO(T,@40,U),DEL);%            %R9038769000
                      END;%                                     %R9038770000
                  END;%                                         %R9038771000
              END ELSE%                                         %R9038772000
              IF FIB[18],[1:1] THEN BEGIN FIB[18],[1:1]+FIB[16]+0; GO EOFIT END; 38773000
              IF REW THEN%                                       %R9038774000
                  BEGIN P(WAITIO(@4200000000,@377,U),DEL);%   %R9038775000
                      STATE,[40:1]+0;%                          %R9038776000
                  END ELSE STATE,[40:1]+NOT STATE,[44:1];%    %R9038777000
              IF REL THEN%                                       %R9038778000
                  BEGIN SETNOTINUSE(U,0);%                     %R9038779000
                      STATE,[41:2]+1;%                          %R9038780000
                  END ELSE STATE,[41:2]+2;%                       %R9038781000
                  IF LOCK THEN%                                    %R9038782000
                      BEGIN SETNOTINUSE(U,1);%                 %R9038783000
                          STATE,[41:2]+1;%                     %R9038784000
                      END;%                                       %R9038785000
                  IF U LSS 16 THEN%                               %R9038786000
                      IF PURGE THEN%                             %R9038787000
                          BEGIN IF PRNTABLE[U]<0 THEN%         %R9038788000
                              FORK(P(,PURGEIT),U,-2,128,1) ELSE SETNOTINUSE(U,0);% %R9038789000
                              STATE,[41:2]+1;%                 %R9038790000
                          END;%                                     %R9038791000
                      GO TO CLEANUP;%                             %R9038792000
                  END;%                                         %R9038793000
              GO TO CLEANUP;%
%

```

```

PP: IF NOT STATE,[41:1] THEN%                                %R9038794000
    BEGIN EMPTY; P(WAITIO(@200450000000,@40,U),DEL) END;%    %R9038795000
    GO TO PX;                                                %R9038796000
%                                                            %R9038797000
PR: IF NOT STATE,[41:1] THEN BEGIN COOLOFF; BLASTQ(U) END;% %R9038798000
    IF REW THEN P(WAITIO(@10340000000,@377,U),DEL);%        %R9038799000
    GO TO PX;%                                               %R9038800000
%                                                            %R9038801000
CD: HEADER=CIDROW[U=32];                                       %R9038802000
    IF M[ALPHA],[27:1] THEN MOVE(10,FIB[16],[33:15],M[ALPHA=2]) ELSE%R38803000
EOD: DO UNTIL READEMFROMDISK(HEADER,M[ALPHA=2]);             %R9038804000
$ SET OMIT = PACKETS                                         38804999
    IF HEADER[3]<HEADER[7] THEN                                %R9038805000
$ POP OMIT                                                  38805001
    IF JAR[P1MIX,0]<0 AND PRT[P1MIX,21]≠0 OR JAR[P1MIX,1]<0 THEN %R9038806000
    BEGIN                                                    38806050
$ SET OMIT = NOT(PACKETS)                                    38806099
    PACKETERR[U=32];=TRUE;                                    38806200
    IF CIDTABLE[U=32,3] LEQ CIDTABLE[U=32,7] THEN          38806300
$ POP OMIT                                                  38806301
    BEGIN STREAM(E+"ENDWAIT": Q=@14, D=M[ALPHA=2]);         38807000
    BEGIN SI=LOC Q; SI=SI+7; IF SC≠DC THEN DI=DI+1; Q=DI; SI=Q; 38808000
    L: IF SC=" " THEN BEGIN SI=SI+1; GO TO L END;           %R9038809000
    DI=LOC E; DI=DI+1; IF 3 SC≠DC THEN TALLY+1;          38810000
$ SET OMIT = NOT(PACKETS)                                    38810099
    IF TOGGLE THEN                                          38810100
    BEGIN SI=SI-3; IF 4 SC=DC THEN TALLY+0; END;          38810200
$ POP OMIT                                                  38810201
    E=TALLY;                                                38810500
    END;                                                    %R9038811000
    IF P THEN GO TO EOD;                                    %R9038812000
    END;                                                    %R9038813000
    END;                                                    38813100
    KIND=0;                                                %R9038814000
    GO TO CC;                                              %R9038815000
%                                                            %R9038816000
CLOSEOUT: STATE,[39:4]+1; TIME+1;%                          %R9038817000
CLEANUP: CLOSED; DK; BKUP; DC;                              %R9038818000
    P(P&RCW(CTC),0,RDS,0,XCH,P&P(CTF),STF);               %R9038819000
END OTHER CLOSE;                                           %R9038820000
PROCEDURE FILEOPEN(XTRA,ALPHA);                              %R9039000000
    VALUE ALPHA,XTRA; INTEGER ALPHA,XTRA;                  %R9039000100
BEGIN REAL RCW=+0;%                                         39001000
    REAL IOM=IOMASK, IOMASK;                                39001100
    REAL XTRAR=-4,XTRAC=-6;                                  %R9039001200
    INTEGER NBUFS,FNUM,BLEN,TYPE,REEL,CDATE,CYCLE,MODE,IO,RLEN,U,KIND,39002000
    DIREC,FORMS,COBOL,UNLABELED,OPTIONAL,CNTCTL;          39003000
    REAL MFID,FID,T1,T2,MASK,STATE;                          39004000
    ARRAY FIB[*],FPB[*];%                                    39005000
    INTEGER ACCESS,FIB7;                                     39006000
    LABEL DCIN,PBS;                                         39006100
    LABEL DKRN,SPN,DKSN,DKUN,DKPN,DCN;                       39007000
    SWITCH INSW=DKRN,SPN,DKSN,DKUN,DCN;                     39008000
    LABEL LOOK,EXIT,LOOKOUT,LPS,FINALIN,FINALOUT,SPDC;%   39009000
    REAL SUBROUTINE CNTLBITS;%                                39026000
    CNTLBITS=IOMASK&MODE[21:47:1]&DIREC[22:47:1]&CNTCTL[23:47:1]39027000
    &IO[24:47:1]&(KIND=7 OR KIND>9 AND KIND≤12)[20:47:1] 39028000
    &(IF KIND=10R KIND=7OR KIND=12THEN@20ELSE 0)[27:42:6];39029000
%                                                            39030000
SUBROUTINE MAKEIODS;%                                       39031000

```

```

BEGIN FIB[16]*T1*((BLEN=1)*DIREC+M[ALPHA])&CNTRLBITS[18:18:15]% 39032000
      &(IF BLEN<1023 THEN BLEN ELSE 1023)[8:38:10]% 39033000
      &TINU[IF (KIND=7 OR KIND=12) THEN IF TYPE<20 39034000
        THEN 20 ELSE 22 ELSE 39034050
        IF KIND=11 THEN 23 ELSE U][3:3:5] OR M; 39034100
FIB[19]*(IF STATE,[46:2]=0 THEN (DIREC INX T1)% 39035000
      &(2*DIREC+(BLEN>1023)+1)[3:43:5] ELSE% 39036000
      IF STATE,[46:2]=1 THEN ((NOT RLEN INX 2)*DIREC INX T1) 39037000
      &RLEN[8:38:10]&(3*DIREC+2)[3:43:5] ELSE% 39038000
      (1=DIREC INX T1)&RLEN[8:38:10]&(DIREC+6)[3:43:5])% 39039000
      &IO[25:47:1]);% 39040000
IF NOT (IO OR COBOL)THEN% 39041000
  T1=FIB[19]&T1[3:3:5]&O[25:25:1]);% 39042000
T2=T1,[33:15]=M[ALPHA]);% 39043000
FOR MASK=0 STEP 1 UNTIL NBUFS=1 DO% 39044000
  BEGIN %P 39045000
  M[ALPHA+MASK]*FLAG((P(DUP,LOD)+T2)&P(T1,XCH)[33:33:15]);% 39046000
  END;% 39047000
END MAKEIODS;% 39048000
LABEL DKRO,SPO,DKSO,DKUO,DKPO,DCO; 39049000
SWITCH OUTSW*DKRO,SPO,DKSO,DKUO,DCO;% 39050000
LABEL FIXFIB,FIND,SPACER;% 39054000
LABEL PREFINAL,DK1;% 39055000
ARRAY HEADER(*);% 39056000
% 39082000
FIB*M[ALPHA=3]; FPB*PRT[P1MIX,3]);% 39083000
IOMASK:=IOM; 39083100
NBUFS=FIB[13],[1:9]; FNUM=FIB[4],[13:11]; BLEN=FIB[18],[3:15];% 39084000
TYPE=FPB[FNUM+3],[43:5];% 39085000
STREAM(S*[FPB[FNUM+2]],D*[CDATE]);% 39086000
BEGIN SI:=S; SI:=SI+3; DS:=5OCT; DS:=OCT; END; 39087000
IF FPB[FNUM+4]>0 THEN REEL = CDATE + CYCLE + 0; 39087500
MODE=FIB[13],[24:1]; IO=FIB[13],[27:1]; RLEN=FIB[18],[33:15];% 39088000
DIREC=FIB[13],[25:1]; FORMS=FPB[FNUM+3],[42:1];% 39089000
STATE=FIB[5]; UNLABELED=FIB[4],[2:1]; 39090000
MFID=FPB[FNUM]; FID=FPB[FNUM+1]; OPTIONAL=FIB[4],[5:1];% 39091000
COBOL=(FIB[13] AND 1)&((FIB)[8:10]=22)[1:47:1]; % COBOL 60 & 68 39091100
KIND=FIB[4],[8:4]; REEL=FIB[13],[28:10];% 39092000
IF TYPE=19 THEN TYPE = 14 ELSE 39092045
IF TYPE=26 THEN GO TO DKPN ELSE 39092050
IF TYPE>26 THEN GO TO DCN; 39092055
  IF TYPE=14 THEN IF LOGLINE,[33:7]=0 THEN TYPE:=11; %R26 39092075
IF (TYPE=0 AND NOT IO) OR TYPE GTR 20 THEN 39092080
BEGIN IF USEPBD THEN TYPE+22; GO LOOKOUT END; 39092090
IF TYPE=1 OR TYPE=4 OR (TYPE>14 AND TYPE<19) THEN 39092100
  IF IO THEN GO DCN ELSE % CANT READ PRINTERS OR PB. %106 39092110
BEGIN IF USEPBD 39092150
  THEN TYPE:=15; 39092160
% SET OMIT = NOT(PACKETS) 39092164
  IF (T1=PSEUDOMIX[P1MIX])>0 AND PACKETPAGE[T1=32]>0 THEN 39092165
  IF FORMS THEN FPB[FNUM+3],[23:1]=1 ELSE % SETS FREEF 39092170
  IF NOT FPB[FNUM+3],[23:1] THEN TYPE:=15; 39092175
% POP OMIT 39092180
  GO LOOKOUT; 39092185
END; 39092190
IF REEL=0 THEN REEL=1; 39092200
IF IO THEN 39092500
  IF TYPE#6 AND TYPE#20 THEN 39093000
  IF TYPE#10 THEN GO TO INSW[TYPE=10] ELSE GO LOOK 39093500
  ELSE GO TO DCN; 39094000

```

```

IF TYPE≥10 AND TYPE≠20 THEN GO TO OUTSW[TYPE=10] ELSE GO LOOKOUT; 39094500
LOOK: IF IO THEN OTHERFILEOPENIN(1) ELSE OTHERFILEOPENOUT(1); 39095000
IF U LSS 0 THEN GO TO EXIT ELSE GO TO PREFINAL; %R9039096000
DCN: FILEMESS(="I/O ERR",0,MFID,FID,REEL,CDATE,CYCLE);% 39143000
GETBUFFERS(BLEN,NBUFS,U,ALPHA);% 39144000
PREFINAL: MAKEIODS;% 39145000
IF KIND=11 THEN 39145100
BEGIN IF COBOL ≤ 0 THEN % ALGOL OR COBOL68 39145200
IF READEMFROMDISK(CIDROW[U=32],M[ALPHA]) THEN 39145210
M[ALPHA]+P(DUP,LOD)&O[2:2:1]&I[27:47:1]; 39145300
END ELSE 39145400
FILLBUFFERS(FIB[16],FIB[19],COBOL,NBUFS); 39146000
IF COBOL>0 THEN FIB[16]+(*P(DUP))&M[ALPHA][CTC]; 39147000
FINALIN: FIB[6] + FIB[7] + FIB[17] + 0; GO TO FIXFIB; 39148000
LOOKOUT: IF IO THEN OTHERFILEOPENIN(0) ELSE OTHERFILEOPENOUT(0); 39155000
IF U LSS 0 THEN GO EXIT ELSE GO FIND; 39156000
FINALOUT: IF NOT FIB[18],[1:1] THEN GETBUFFERS(BLEN,NBUFS,U,ALPHA);% 39230000
BIND: MAKEIODS;% 39231000
FIB[6]+FIB[7]+0;% 39232000
FIB[17]+IF COBOL THEN FIB[18],[3:15]ELSE FIB[18],[18:15];% 39233000
IF KIND = 10 THEN 39233100
M[ALPHA+1]+P(DUP,LOD)&P(DUP,LNG)[24:24:1]; 39233200
GO TO FIXFIB;% 39234000
DCIN: 39234900
DCO: U+30; KIND+10; 39235000
IF (BLEN≠RLEN) LSS 17 THEN BLEN:=17; 39235100
FIB[13],[1:9]+NBUFS+2; 39235200
FIB[18]:=(*P(DUP))&BLEN[3:33:15]&BLEN[CTF]; 39235250
GO TO SPDC; 39235300
SPO: MODE+0; U+25; KIND+5; 39236000
SPDC: CNTCTL+DIREC+0; UNLABELED+1; 39237000
STARTIMING(FNUM,U);% 39238000
GO TO FINALOUT;% 39239000
SPN: U+25; KIND+5; 39240000
MODE+CNTCTL+DIREC+0; UNLABELED+1;% 39241000
STARTIMING(FNUM,U);% 39242000
IF BLEN<10 THEN BLEN+10;% 39243000
GETBUFFERS(BLEN,NBUFS,U,ALPHA);% 39244000
MAKEIODS;% 39245000
GO TO FINALIN;% 39246000
DKRN: DKRO: ACCESS:=1; %R9039247000
GO TO DK1; %R9039248000
DKUO: IO:=1; %R9039249000
DKUN: ACCESS:=2; %R9039250000
GO TO DK1; %R9039251000
DKPN: DKPO: 39252000
$ SET OMIT = NOT SHAREDISK 39252999
ACCESS+1; 39253000
GO TO DK1; 39254000
$ POP OMIT 39254001
$ SET OMIT = SHAREDISK 39254999
GO TO DCN; 39255000
$ POP OMIT 39255001
DKSN: DKSO: ACCESS+0; 39256000
DK1: DISKFILEOPEN(0); 39257000
IF T1 THEN GO TO EXIT; 39258000
BIXFIB: FIB[4],[2:1]+UNLABELED;% 39294000
FIB[4],[8:4]+KIND;% 39295000
FIB[15],[24:6]+U; 39296000
FIB[13],[28:10]+REEL;% 39297000

```

FPB=PRT[P1MIX,3];	39297010
FPB[FNUM+3],[43:5]+TYPE;	39297020
STREAM(REEL,D+[FPB[FNUM+2]]);	39297100
BEGIN SI=LOC REEL;	39297200
IF 3 SC=DC THEN	39297300
BEGIN DI=D; SI=LOC REEL; DS=3 DEC END;	39297400
END;	39297500
RDCTABLE[U],[8:6]+P1MIX;%	39298000
IF FIB[18],[1:1] THEN%	39299000
BEGIN FIB[16]+0;%	39300000
FIB[5]+STATE&B[39:42:6];%	39301000
FIB[10],[3:15]+0;	39301100
END ELSE%	39302000
FIB[5],[CF]+STATE&DIREC[44:47:1]&IO[39:43:5]&FIB[5][45:45:1];	39303000
IF COBOL>0 OR FIB[4],[7:1] THEN M[FIB INX NOT 1],[3:6]+6	39304000
ELSE M[ALPHA=7],[3:6]+4;%	39305000
FIB[4],[27:3]+ACCESS;%	39306000
IF U<16 THEN IF KIND#7 THEN FPB[FNUM+3],[23:1]:=10;	39306010
	39306100
	39306200
	39307000
EXIT:;%	
IF XTRA THEN	%R9039307100
XTRAC:=NOT(FIB[4],[7:1] OR UNLABELED) AND XTRAC NEQ 2;	%R9039307200
IF XTRA LSS 2 THEN GO TO INITIATE;	%R9039307300
RCW:=XTRAR;	%R9039307400
END FILEOPEN;%	39308000
PROCEDURE CREATELOG(DDD); VALUE DDD; ARRAY DDD[*];	39500000
BEGIN ARRAY A=LOGARRAY[*];	39501000
DEFINE IO=A[31]#;	39502000
DELTA=A[32]#;	39503000
N=A[33]#;	39504000
S=A[34]#;	39505000
R=A[35]#;	39506000
H=A[36]#;	39507000
LABEL GETANOTHERROW,NEWLOG,AGAIN;	39507500
REAL T; INTEGER I,RC;	%LOG39508000
ARRAY T1=RC[*];	39508050
REAL B,J,K,DISK;	39508100
SUBROUTINE FIX;	39508200
BEGIN M[T]:=[M[J]]&I[8:38:10];	39508300
J:=J+1;	39508400
END;	39508500
	39508510
	39508520
SUBROUTINE BUILDHEAD;	39508540
BEGIN M[T]:=0;	39508560
MOVE(29,T,T+1);	39508580
M[T+2]:=MCP;	39508600
M[T+4],[9:1]:=1;	39508620
M[T+5]:=M[T+6]:=@14;	39508640
M[T+9]:=1;	39508660
STREAM(DATE,X:=T+3);	39508680
BEGIN SI:=LOC DATE; DS:=8 OCT;	39508700
DI:=X; DS:=2 LIT"+#";	39508720
SI:=X; SI:=SI+5; DS:=3 CHR;	39508740
END;	39508750
M[T+1],[6:18]+M[T+3],[30:18]&	39508755
(XCLOCK+P(RTR))[25:25:23];	39508760
M[T+10]:=PETUSERDISK((M[T+8]:=1)&I[2:47:1],1);	39508780
END;	39508800

```

X
DISK:= "DISK  ";
A[30]:=NOT 0;
IF (T:=DIRECTORYSEARCH("LOG  ")
$ SET OMIT = NOT(SHAREDISK)
&(SYSNO+17)[24:42:6]
$ POP OMIT
  ,DISK,5))=0 THEN
BEGIN T:=SPACE(30);      % IF YOU CANT FIND ONE, MAKE %LOG
DISKWAIT(-T,-30,DIRECTORYTOP=SYSNO);
MIT INX 20],[8:10] := N := 0;
DISKWAIT(T,-30,DIRECTORYTOP=SYSNO);
S:=(I:=300) OR MEMORY;
BUILDHEAD;
IF (R:=M[T+10])=0 THEN
BEGIN M[T INX 7]:=-1;
  RC:=0;
END ELSE
BEGIN M[T INX 7]:=899;      %EOF POINTER
  DISKWAIT(A INX 30,1,R);  %EOF MARKER
END;
M[T] := @0001200036000301; %BLOCKING %LOG
H:=EUF("LOG  ")
$ SET OMIT = NOT(SHAREDISK)
&(SYSNO+17)[24:42:6]
$ POP OMIT
  ,DISK,T=1);
END ELSE
BEGIN          %AHA, THERE REALLY IS A LOG
H+T,[FF];
  S:=M[T INX 8]&1[2:47:1];
  I:=9;
DO I:=I+1 UNTIL M[T INX I]=0 OR I=30;
RC:=I:=I-10;
IF N DIV S>=I THEN
GETANOTHERROW:
IF RC LSS 20 THEN      % DONT TRY TO GET 21-ST ROW
BEGIN N+I*S;
IF P(M[T INX I+10])=PETUSERDISK(-S,1),DUP)=0 THEN
  BEGIN P(DEL);
    RI:=0;
  END ELSE
  BEGIN RI:=P(XCH)=N;
    DISKWAIT(A INX 30,1,R+N);
    RC:=RC+1;
  END
END ELSE ELSE
BEGIN R+M[T INX I+9)=(I+(I-1)*S);
  J:=I+S;
IF N#0 OR I#0 THEN      % SET UP SEARCH FOR EOF
IF N LSS J THEN NI:=I ELSE I:=N-1 ELSE
IF M[T INX 5]#@14 THEN    % MUST BE COLD START
BEGIN M[T INX 5]:=M[T INX 6]:=@14;
  GO TO NEWLOG;
END;
KI:=1;
B := SPACE(30);
FOR I+I STEP 1 UNTIL J DO
BEGIN DISKWAIT(-B,30,R+I);
  IF (M[B]=NOT 0) OR K>(K+M[B],[25:23]) THEN

```

```

39508810
39508850
39508900
39509000
39509009
39509010
39509011
39509020
39509050
39509100
%LOG39509150
39509200
39509250
39509260
39509265
39509270
39509275
39509280
39509285
39509290
39509295
%LOG39509300
39509850
39509859
39509860
39509861
39509870
%LOG39510000
%LOG39510050
39518000
39519000
%LOG39519050
%LOG39519100
%LOG39519150
39519200
%LOG39519250
%LOG39519270
39519300
39519320
39519340
39519360
39519380
39519400
%LOG39519450
%LOG39519500
39519550
%LOG39519600
39520000
39520100
39520200
39520300
39520400
39520500
39520600
39520700
39520800
39521000
39522000
39523000
39524000

```

```

                BEGIN J←0; N←I END;
END;
FORGETSPACE(B);
IF J≠0 THEN GO GETANOTHERROW;
END;
NEWLOG: M[T INX 7] := (M[T INX 9] := RC) × 3 × S - 1;
DISKWAIT(T, [CF], 30, H);
RC := RC - 1;
END;
FORGETSPACE(T);
FORK(P, LOGWARN, RC, 0, 128, 0);
IF (T := DIRECTORYSEARCH("SYSTEM "
$ SET OMIT = NOT(SHAREDISK)
                &(SYSNO+17)[42:42:6]
$ POP OMIT
                ,DISK,5))≠0 THEN
BEGIN SYSDISKADR←M[T INX 10];
J := SPACE(10); M[J INX 4] := 0;
IF SYSDISKADR NEQ 0 THEN DISKWAIT(-J, 5, SYSDISKADR);
I := M[J INX 4];
IF I, [40:8] = 0 OR I, [32:8] = 0 OR % LMAX = 0 OR STAMAX = 0
    I, [40:8] GTR I, [32:8] OR % LMAX GTR STAMAX
    I, [32:8] NEQ M[T+7] THEN % STAMAX NEQ EOF
BEGIN SYSDISKADR := SYSDISK := 0;
STREAM(SN := ("SYSTEM "
$ SET OMIT = NOT SHAREDISK
                &(SYSNO+17)[42:42:6]
$ POP OMIT
                ), J);
BEGIN SI := LOC SN; SI := SI + 1; DS := LIT"="; DS := 7CHR;
DS := 27LIT"/DISK INCORRECT = NOT USED";
END;
SPOUT(J);
END ELSE
BEGIN STREAM(X := M[T+3], [30:18], B := [B]);
BEGIN SI := LOC X; DS := 8DEC; END;
GIMEDATE([B], [CF], "B");
STREAM(SN := ("SYSTEM "
$ SET OMIT = NOT SHAREDISK
                &(SYSNO+17)[42:42:6]
$ POP OMIT
                ), B,
                LX := I, [40:8], SX := I, [32:8], J);
BEGIN SI := LOC SN; SI := SI + 1; DS := LIT"="; DS := 7CHR;
DS := 23LIT"/DISK CHECKED (CREATED ";
SI := LOC B; SI := SI + 2; 3(DS := 2CHR; DS := LIT"/");
DI := DI - 1; DS := 7LIT", LMAX=";
DS := 2DEC; J := DI; DI := DI - 2; DS := FILL; DI := J;
$ SET OMIT = TWXONLY
                DS := 9LIT", STAMAX="; DS := 2DEC; J := DI; DI := DI - 2;
DS := FILL; DI := J;
$ POP OMIT
                DS := 2LIT")+";
END;
SPOUT(J);
END;
                SYSDISK ← M[T];
                UNLOCKTOG(SYSDISKMASK);
                T1 := IOQUE&T[CTC];
                SYSDISKIO(1, 0, T1);

```

```

39525000
39526000
39528000
%LOG39528050
39528100
39528200
%LOG39528300
39528500
%LOG39529000
%LOG39530000
39530500
39531000
39531099
39531100
39531101
39531200
39532000
39532100
39532200
39532300
39532400
39532500
39532600
39532700
39532800
39532900
39533000
39533100
39533200
39533300
39533400
39533500
39533600
39533700
39533710
39533720
39533730
39533740
39533750
39533760
39533770
39533780
39533790
39533800
39533810
39533820
39533830
39533840
39533850
39533860
39533870
39533880
39533890
39533900
39533910
39533920
39534000
39534500
39534600
39535000

```

```

                IF (LMAX:=T1[4],[40:8J]) GTR MAXLMAX THEN 39536000
                    LMAX:=MAXLMAX; 39536500
$ SET OMIT = TWXONLY 39536999
                IF (STAMAX:=T1[4],[32:8J]) GTR MAXLMAX THEN 39537000
                    STAMAX:=MAXLMAX; 39537500
                IF (K:=T1[4],[24:8J]) GTR MAXLMAX THEN K:=MAXLMAX; 39538000
$ POP OMIT 39538001
                SYSDISKIO(1,STAMAX+1,T1); 39539000
                FORGETSPACE(T); 39540000
                B:=(J:=M[RC:=(*(T:=P(.LINETABLE)))=2],[CF] 39541000
$ SET OMIT = TWXONLY 39541499
                    -LMAX=4*STAMAX-K-6)-2; 39541500
$ POP OMIT 39541501
$ SET OMIT = NOT(TWXONLY) 39541599
                    -5*LMAX=5)-2; 39541600
$ POP OMIT 39541601
                I:=LMAX+1; FIX; 39542000
$ SET OMIT = TWXONLY 39542999
                I:=STAMAX+1; 39543000
$ POP OMIT 39543001
                T:=P(.STATABLE); FIX; 39544000
                T:=P(.SEQARRAY); FIX; 39545000
                T:=P(.INPUTANK); FIX; 39546000
                T:=P(.TANKS); FIX; 39547000
$ SET OMIT = TWXONLY 39547999
                I:=K+1; 39548000
                T:=P(.TNAOG); FIX; 39549000
                IF LMAX#MAXLMAX OR K#MAXLMAX THEN 39549100
$ POP OMIT 39549101
                BEGIN 39549200
                    M[M[B]:=M[RC]&RC[CTF]], [FF]:=B; 39550000
                    M[RC],[CF]:=B; 39551000
                    FORGETSPACE(RC+2); 39552000
                END END; 39553000
% 395580900
% NOW CHECK FOR LIBMAIN, LDCNTRL AND PRNPBT AND CREATE 395580910
% THEM IF THEY ARE NOT THERE. 395580920
% 395580930
                ENTERSYSFILE(1); % "LIBMAIN" 395581000
                ENTERSYSFILE(2); % "LDCNTRL" 395582000
                ENTERSYSFILE(3); % "PRNPBT " 395583000
                END OF CHECKING LOG AND OTHER SYSTEM FILES; 39625000
PROCEDURE SUSTATUS(A,DDD,B); VALUE A,DDD,B; REAL A,B; ARRAY DDD[*]; 39900000
    BEGIN REAL RT1,I; 39901000
        ARRAY D[*],ZSF[*],VADAR[*]; 39902000
        SUBROUTINE SPOUTITNOW; 39903000
        BEGIN 39904000
            STREAM(X:=[TINU[B]], D, EUNUM:=0, SUI:=0, I, RT1); 39905000
            BEGIN SII=X; SII:=SII+5; 39906000
                DSI=LIT" "; DSI:=3 CHR; SII:=D; 39907000
                10(IF SC#"0" THEN 39908000
                    BEGIN X:=SII; EUNUM:=TALLY; 39909000
                        SII=LOC EUNUM; DSI= 3 LIT" EU"; DSI=DEC; 39910000
                        DSI=4 LIT" SU "; TALLY:=0; 39911000
                        5(SUI=TALLY; SII=X; SKIP SB; SKIP SU SB) 39912000
                            IF SB THEN 39913000
                                BEGIN SII=LOC SU; 39914000
                                    DSI=DEC; DSI=LIT", "; 39915000
                                END; 39916000
                                TALLY:=TALLY+1); 39917000
                    END
        END

```



```

                SI:=X; TALLY:=EUNUM;
                END;
                TALLY:=TALLY+1; SI:=SI+1);
                SI:=LOC I; SI:=SI+7; DI:=DI-1;
                IF SC#0 THEN
                BEGIN DS:=5 LIT WENT";
                    IF SC#2 THEN DS:=4 LIT NOT";
                END ELSE DS:=4 LIT ARE";
                DS:=8 LIT READY,+";
            END;
            SPOUT(RT1);
        END OF SPOUTING IT;

        SUBROUTINE DOIT;
        BEGIN
            IF NOT (ZSF[0] OR ZSF[1],[1:11]) # NOT 0 THEN
            BEGIN B:=18; D:=ZSF;
                RT1:=SPACE(20);
                SPOUTITNOW;
            END;
            IF NOT (ZSF[2] OR ZSF[3],[1:11]) # NOT 0 THEN
            BEGIN B:=19; D:=ZSF[2];
                RT1:=SPACE(20);
                SPOUTITNOW;
            END;
        END OF DOING IT;

%
%
%
        START OF CODE

        IF B#0 THEN
        BEGIN D:=[MULTITABLE[16]]&2[8:38:10];
            RT1:=A;
        $ SET OMIT = DFX
            IF B THEN DI:=2 INX D;
        $ POP OMIT
            IF NOT (IF B THEN P(RRR),[28:1] ELSE P(RRR),[29:1])
        $ SET OMIT = DFX
            OR NOT (D[0] OR D[1],[1:11]) = NOT 0
        $ POP OMIT
            THEN
            BEGIN STREAM(X:=[FINU[B]], RT1);
                BEGIN SI:=X; SI:=SI+5;
                    DS:=LIT " "; DS:=3 CHR;
                    DS:=11 LIT NOT READY,+";
                END;
                SPOUT(RT1);
            END ELSE SPOUTITNOW;
        END ELSE
        BEGIN ZSF:=[M[SPACE(4)]]&4[8:38:10];
            VADAR:=[MULTITABLE[16]]&4[8:38:10];
            DISKWAIT(-A,-30,DIRECTORYTOP);
            FOR I:=0 STEP 1 UNTIL 3 DO
                ZSF[I]:=VADAR[I] AND NOT DDD[23+I];
                I:=1; DOIT;
            FOR I:=0 STEP 1 UNTIL 3 DO
                BEGIN ZSF[I]:=NOT VADAR[I] AND DDD[23+I];
                    DDD[23+I]:=VADAR[I];
                END;
            DISKWAIT(A,-30,DIRECTORYTOP);
            I:=2; DOIT;

```

```

39918000
39919000
39920000
39921000
39922000
39923000
39924000
39925000
39926000
39927000
39928000
39929000
39929100
39930000
39931000
39932000
39933000
39934000
39935000
39936000
39937000
39938000
39939000
39940000
39941000
39942000
39942900
39942910
39942920
39943000
39944000
39945000
39945999
39946000
39946001
39947000
39947999
39948000
39948001
39949000
39950000
39951000
39952000
39953000
39954000
39955000
39956000
39957000
39958000
39959000
39960000
39961000
39962000
39963000
39964000
39965000
39966000
39967000
39968000
39969000

```

```

FORGETSPACE(ZSF);
END;
END;
PROCEDURE DIRECTORYBUILDER(A,DDD);
VALUE A,DDD;
REAL A;
ARRAY DDD[*];
BEGIN REAL Y,Z,B,C,I,J,T,RA,RL,RT1,R; INTEGER RADD,RLEN;
REAL NEXTLINK,AD,X,K,SEVEN7,FORTY,L,EUSU;
ARRAY SU[*];
ARRAY HEAD[*],KK[*],PL[*];
REAL W,ESPADD,DISKTOP,SUPER,EUM,NT1,NT2,NT3,NT4;
BOOLEAN UCHANG,ERROR; INTEGER LO,REM,TN,TM,MN; REAL X1,X2,EUMASK;
ARRAY ZSF[*],SOCK[*];
REAL D,Y1,Y2;
REAL AA,AAA;
LABEL FORGET;
ARRAY V[*,*];
INTEGER S;
ARRAY VR=V[*];
REAL H,FI,FJ;
$ SET OMIT = NOT SHAREDISK
REAL HOLDER,NEXTSLOT,BYPASS;
$ POP OMIT
LABEL LOOKATDKB,BACK,EXIT,M1,SKBLK,LTR;
DEFINE ROW=SU[X],[3:4]#,
LASTAVAIL=HEAD[0],[3:15]#,
AVAILABLE=HEAD[0],[FF]#,
FIRSTLINK=HEAD[0],[CF]#,
DA=9:24#,DAC=9:24:24#,
SIZE=PL[1],[DA]#,
ADDRESS=PL[0],[DA]#,
HIGHLINK=PL[0],[CF]#,
LOWLINK=PL[1],[CF]#,
DISKRUNNING=[18:1]#,
FORTYMILLDISK=[19:1]#,
OCCUPIED=[20:1]#,
AV1=480#,AVBLOCK=16#;
SUBROUTINE SAVIT;
BEGIN
IF (W+W+2)≥28 THEN
BEGIN ZSF[29]←ESPADD;DISKWAIT(ZSF INX 0,30,ESPADD+GETESPDISK);
W←0 END;ZSF[W]←T;ZSF[W+1]←DDD[479-2×I];
END SAVIT;
SUBROUTINE CLEAR;
BEGIN V[S,0]←0;
V[S,1] ← BYPASS,[CF];
V[S,2]←@14;
V[S,3]←V[S,4]←0;
MOVE(57,[V[S,2]],[V[S,5]]);
END;
SUBROUTINE SETUP;
BEGIN
LO:=(X+1) MOD 5;LO:=LO+(LO=0)×5;
IF RADD NEQ (LO:=LO×FORTY) OR (LO=RADD AND RLEN LSS FORTY) THEN
BEGIN
IF Y:=(SU[X],[CF]=0) THEN
BEGIN
NT1:=SU[X];:=SPACE(16)&SU[X][18:18:9];
MOVE(16,NT1-1,NT1);

```

```

39970000
39971000
39972000
40000000
40001000
40002000
40003000
40004000
40004500
40005000
40005050
40005100
40005110
40005200
40005210
40005220
40005230
40006000
40006100
40007000
40007500
40007990
40008000
40008010
40008050
%MC40008100
40008110
40008120
40008130
%MC40008140
40008150
40008160
40008170
40008180
%MC40008190
%MG40008200
%MC40008210
40008220
%MQ40008300
%MQ40008310
%MQ40008320
%ME40008330
%MC40008340
%MC40008350
40009000
40010000
40011000
40012000
40013000
40014000
40015000
%MC40016000
40016020
40016025
40016026
%MC40016100
40016200
%MC40016220
40016240
%027=40016260

```

```

END;
M[SU[X] INX K]:=RT1:=SPACE(64+Y);
KK*[M[RT1]]*(64+Y)[8:38:10];JUNK*61+Y;
MOVE(64+Y,RT1=1,RT1);
FOR R:=3*X STEP 2 UNTIL JUNK DO KK[R]:=RT1+R+2;
HEAD*[M[M[SU[X]]]]&1[8:38:10];
IF Y THEN
BEGIN
  KK[1]:=KK[2]:=SEVEN7;
  KK[1],[DA]:=L0;
  KK[2],[DA]:=IF X EQ 0 THEN FORTY-(DISKBOTTOM+5) ELSE FORTY;
  M[SU[X]],[DA]:=L0;
  HEAD[0]:=RT1+1;
END;
HEAD[0],[FF]*RT1+3*Y;
HEAD[0],[3:15]*62+RT1+Y;
END
ELSE
DO
BEGIN SU[X],OCCUPIED:=1;
  RADD:=RADD-FORTY;
  X:=X+1;
END UNTIL (RLEN:=RLEN-FORTY) LSS FORTY;
END OF SETUP;
SUBROUTINE BUILDVAIL;
BEGIN
BACK:=ERROR+1;REM*0;
IF (Z:=SU[X])#0 AND Z,[CF]=0 THEN
  BEGIN K:=0; SETUP; GO BACK END;
IF (Z:=SU[X]),DISKRUNNING AND NOT Z,OCCUPIED AND RLEN>0 THEN
  BEGIN
    IF M[SU[X]],[DA] GEQ RADD THEN
      BEGIN
        P(M[M[SU[X]]],O&RADD[9:24:24],LLL*0,INX*,AD*,DEL);
        HEAD*[M[M[SU[X]]]]&1[8:38:10];PL*[M[AD]]&2[8:38:10];
        IF ((RA:=ADDRESS)-(RL:=SIZE) LSS RADD-RLEN OR
          (REM:=IF(NT1:=RADD MOD FORTY)=0 THEN 0 ELSE NT1-RLEN) LSS
          0)AND RADD NEQ RA THEN
          BEGIN
            IE REM LSS 0 THEN RLEN:=RADD MOD FORTY;
            IF AVAILABLE=0 THEN%NEED ANOTHER ROW
              BEGIN
                K*ROW;K*+1;ROW*+K;
                IF K GTR 15 THEN
                  BYBY("TOO MANY ROWS NEEDED BY DIRECTORYBUILDER*",41);
                SETUP;
              END;
            NEXTLINK*[M[R]-AVAILABLE];
            M[R]*AD&(RADD-RLEN)[DAC];
            IF AD,[CF]=SEVEN7 THEN M[SU[X]],[DA]*RADD-RLEN;
            IF LOWLINK=SEVEN7 THEN
              FIRSTLINK:=R
            ELSE
              M[LOWLINK],[CF]*R;
            M[R+1]*PL[1]&(RADD-RLEN-(RA-RL))[DAC];
            PL[1]*R&(RA-RADD)[DAC];
            RLEN*0;
            AVAILABLE*NEXTLINK;ERROR*FALSE;
          END
          ELSE%REDUCE EXISTING AREA(BEWARE OF ADDRESS CONFLICT OR%

```

```

%MC40016300
40016400
%027-40016410
%027-40016420
40016500
%MC40016510
%MC40016600
40016700
40016800
40016900
40016910
40016920
40017050
40017100
%MC40017200
%MC40017250
40017260
40017270
40017275
40017280
40017285
40017290
40017295
%MC40017300
%MC40027100
%MC40027200
%MC40027230
40027240
40027245
40027250
%MC40027260
40027270
40027280
%MC40027290
%MC40027295
40039000
40039100
40039200
%MC40040000
40040500
%MC40041000
%MC40042000
%MC40042100
40043000
40043500
%MC40044000
%MC40045000
%MC40046000
%MC40047000
%MC40047100
40048000
40049000
40050000
40051000
40055000
40056000
%MC40056100
%MC40057000
%MC40058000
%MC40059000

```

```

        %EU UNDERFLOW),
BEGIN
  IF RADD=RA AND RL GE0 RLEN THEN
  BEGIN
    ADDRESS←RA-RLEN;
    IF HIGHLINK=SEVEN7 THEN
    MESU[X]], [DA]←ADDRESS;
    SIZE←RL-RLEN; ERROR←RLEN←0;
  END
  ELSE
  IF RLEN>RL THEN
  IF LOWLINK=SEVEN7 AND(X-1)MOD 5≠4 THEN
  BEGIN
    RADD←RADD-RL-1; RLEN←RLEN-RL-1; SIZE←0; ERROR←0;
  END
  ELSE
  IF RADD=RLEN LSS (NT1:=M[LOWLINK], [DA]) THEN
  BEGIN
    RLEN:=RLEN-(RADD-(RADD:=NT1));
    SUPER:=1; GO BACK;
  END
  ELSE
  IF RADD GTR RA-RL THEN
  BEGIN
    RLEN:=RADD-(RA-RL); SUPER:=1;
    GO BACK;
  END
  ELSE RLEN← 0
  ELSE
  BEGIN SIZE←RL-RLEN; ERROR←RLEN←0; END;
END;
  IF SIZE=0 THEN
  BEGIN
    IF HIGHLINK=SEVEN7 AND LOWLINK=SEVEN7 THEN
    BEGIN
      SU[X].OCCUPIED←TRUE;
      K←1;
      WHILE(Y←M[SU[X]INX (K+K+1)])≠0 AND K≤15 DO
        FORGETSPACE(Y);
        FORGETSPACE(SU[X]);
      END
    ELSE
    BEGIN
      IF HIGHLINK=SEVEN7 THEN
      BEGIN
        M[PL[1]], [CF]←SEVEN7;
        M[SU[X]], [DA]←M[PL[1]], [DA];
      END
      ELSE
      BEGIN
        M[PL[0]+1], [CF] :=LOWLINK;
        IF LOWLINK=SEVEN7 THEN
          FIRSTLINK←HIGHLINK
        ELSE
          M[PL[1]], [CF] :=HIGHLINK;
        END;
        IF M[LASTAVAIL]=0 THEN
          M[LASTAVAIL]←AD; LASTAVAIL←AD;
        IF AVAILABLE=0 THEN AVAILABLE←AD;
        PL[0] :=0;

```

```

%MC40060000
%MG40060050
  40060100
%MC40060200
%MC40060300
%MC40060302
%MG40060305
%MC40060400
%MG40060500
%MC40060600
%MC40061000
%MC40062000
%MC40063000
%MC40064000
%MC40065000
  40065010
  40065020
  40065030
  40065040
  40065050
  40065060
  40065070
  40065080
  40065090
  40065100
  40065110
  40065120
%MC40066000
%MC40067000
%MC40068000
%MC40068050
%MG40068100
%MC40069000
%MC40070000
%MG40071000
%MC40072000
%ME40073000
%MC40074000
%MC40075000
%MC40076000
%MG40077000
%MC40078015
%MG40078020
%MC40078030
%MC40078031
%MC40078032
%MG40078033
%MC40078034
%MG40078035
%MC40078036
  40078038
%MC40078040
%MG40078042
%MC40078046
  40078048
%MC40078050
%MG40078052
%MG40078054
%MC40078058
  40078060

```

```

                END;
            END;
    IF REM LSS 0 THEN BEGIN RADD=X MOD 5;RADD=(RADD+(RADD=0))*FORTY;
        RLEN=ABS(REM); END;
        X=X-(RLEN#0);
    END ELSE
        IF(NT1:=M[SU[X]],[DA]) GTR RADD=RLEN THEN
            BEGIN RLEN:=RLEN-(RADD-(RADD#NT1));
                SUPER:=1; GO BACK;
            END
        ELSE
            RLEN#0;
    END;
    IF RLEN#0 AND NOT ERROR THEN GO BACK;
    SUPER:=SUPER OR (ERROR AND SU[X],DISKRUNNING);
    END OF COMPLEMENTING DISK DIRECTORY;
    SUBROUTINE LOCKED;
    BEGIN
    IF (X1:=(RADD-RLEN) DIV TN)=(X2:=RADD DIV TN) THEN
    IF(TWO(X1) AND EUM)=0 THEN BUILDAVAIL ELSE GO FORGET ELSE
    BEGIN
        Y1:=RADD;Y2:=RLEN;
        IF(RLEN:=(X1+1)*TN-(RADD-Y2 ) )GTR 0 AND (TWO(X1) AND EUM)=0 THEN
            BEGIN RADD:=(X1+1)*TN;X1:=5*D+((Y1-Y2)DIV FORTY);BUILDAVAIL END;
        IF (RLEN:= Y1-(X2*TN)) GTR 0 AND (TWO(X2) AND EUM) EQL 0 THEN
            BEGIN RADD:=Y1;X1:=5*D+RADD DIV FORTY;BUILDAVAIL; END;
        WHILE (X2:=X2-1) GTR X1 DO
            BEGIN
                RLEN:=TN;X1:=5*D+((RADD:=(X2+1)*TN)-1)DIV FORTY;
                IF (TWO(X2) AND EUM)=0 THEN BUILDAVAIL;
            END;
    END;
    FORGET;
    END OF LOCKED;
    *
    $ SET OMIT = NOT SHAREDISK
    BYPASS:=DISKBOTTOM+2; % SET AT 44240580
    HOLDER:=DIRECTORYTOP-7-(HOLDMAX+29) DIV 30;
    $ POP OMIT
    SU:=[M[RT1:=SPACE(100)]]&100[8:38:10];
    SEVEN7:=@77777;FORTY:=40000;TN:=10000;MN:=1000000;TM:=10000000;
    MOVE(100,RT1-1,RT1);
    SOCK:=[M[RT1:=SPACE(40)]]&40[8:38:10];
    MOVE(40,RT1-1,RT1);
    X1:=NEUP,[3:15]=1;% CHECK ONLY UNITS THAT EXIST
    VR:=[MULTITABLE[16]]&4[8:38:10];
    LOOKATDKB;
    FOR J:=0 STEP 1 UNTIL X1 DO
    BEGIN
        X2:=19;
        FOR I:=0STEP 1 UNTIL X2 DO
        BEGIN
            RADD:=MN*XJ+I*TN;
            STREAM(Q:=RADD,B:=40+A);
            BEGIN SI:=LOC Q;DS:=8 DEC END;
            IF I EQL 0 THEN
                BEGIN X2:=20*WAITIO(40+A INX@140000000,@64,18+C),[43:1]+X2;
                    IF X2#39 THEN VR[NT1:=1+C*X2]:=P(DUP,LOD) OR TWO(11-J);
                END;
            IF NOT(R*WAITIO(40+A INX @100000000,@64,18+C)),[42:1] THEN

```

```

%MC40078065
%MC40078067
40078068
40078069
%MG40078070
40078072
40078074
40078076
40078078
%MC40078080
40078085
%MG40078087
%MQ40078090
%MC40078091
40078092
%024-40078093
40100000
40100100
40100200
40100300
40100400
40100500
40100600
40100700
40100800
40100900
40101100
40101200
40101250
40101300
40101400
40101500
40101510
40101600
40199900
40199990
40200000
40200100
40200110
40249100
40249105
%027-40249110
40249120
40249130
40249200
40249250
%MC40249300
40250000
40251000
40252000
40253000
40254000
40254100
40255000
40256000
40257000
40257030
40257060
40257100
%024-40258000

```

```

BEGIN 40261000
NT2:=(NT1:=5×J+50×C)+(I DIV(SU[NT1],FORTYMILLDISK+1)DIV 4); 40261010
SU[NT2]:=P(DUP,LOD)&1[18:47:1]&(X2>19)[19:47:1]; %031=40261040
IF R,[43:1] THEN 40261042
BEGIN FORTY:=FORTY×((X2 GTR 19)+1); 40261043
SOCK[C×10+J]:=(P(DUP)) OR TWO(I); 40261044
X:=NT2;RADD:=(RADD MOD MN)+(RLEN:=TN);BUILDAVAIL; 40261046
FORTY:=40000; 40261047
END ELSE SOCK[C×10+J+20]:=(P(DUP)) OR TWO(IF X2=19 THEN I ELSE 40261048
(I DIV 8)×4 + (I AND 3)); 40261049
END ELSE %NOT READY CHECK NEXT SU 40261050
BEGIN EUSU:=EUSU OR TWO(4-(IF X2=19 THEN I ELSE (I DIV 8)×4+(I AND 40261100
3)DIV 4); 40261150
I:=I+(((SU[NT1:=(5×J+50×C)],FORTYMILLDISK+1)×4)-1); 40261200
END END; 40261250
STREAM(A:=(NOT EUSU),[43:5], J, D:=VR INX C INX C); 40261300
BEGIN SI:=LOC A; SI:=SI+7; 40261350
DII:=DI+J; DS:=CHR; 40261400
END; 40261450
EUSU:=0; 40261500
END; 40262000
$ SET OMIT = NOT(DKBNODFX AND NOT DFX) 40262299
IF NOT C AND (X1+NEUP,[FF]=NEUP,[CF]=1)≥0 THEN 40262300
BEGIN CI=1; 40262310
IF P(RRR),[28:1] THEN GO TO LOOKATDKB ELSE 40262320
BEGIN STREAM(J:=J:=SPACE(60)); 40262330
DS:=44 LIT"DKB NOT READY - IGNORED BY DIRECTORYBUILDER*"; 40262340
SPOUT(J); 40262350
END; END; 40262360
$ POP OMIT 40262361
$ SET OMIT = NOT(DFX) 40262369
NEUP:=NEUP&NEUP[CF]; % REMOVE ANY EUS DECLARED ON DKB 40262370
$ POP OMIT 40262371
J+DIRMOD; 40262500
V := [M[SPACE(J)]]&J[8:38:10]; 40263000
J + J=1; 40264000
H+I+I←-1; 40264100
FOR S ← 0 STEP 1 UNTIL J DO 40264500
BEGIN IF T=I THEN 40265000
LTR: BEGIN IF (H+H+1)≤CHUNKMAX THEN 40265100
BEGIN IF TOTAL[H]≠0 THEN GO TO LTR; 40265200
I←CHUNKZIZE×H+FENCE; 40265300
END ELSE I := SPACE(CHUNKZIZE); 40265400
T←I+CHUNKZIZE; 40265500
END; 40265600
VR[S]←[M[I]]&62[8:38:10]; 40265700
I←I+64; 40265800
BYPASS←BYPASS+2; 40266000
CLEAR; 40267000
END; 40268000
AAA:=AAI:=SPACE(480); 40275200
DISKWAIT(=A,480,J:=DIRECTORYTOP+4); 40275300
ZSF←IOQUE&SPACE(31)[CTC]; 40275500
ZSF[0]←@14; 40275600
W←0; 40275700
FOR J:=J STEP 16 WHILE J≠16 DO 40276000
BEGIN 40277000
DISKIO(NT3,=(AAA=1),480,J+16); 40278000
IF J+15≥BYPASS,[CF] THEN DIRECTORYFULL(BYPASS); 40278100
BYPASS,[FF]←J+15; 40278200

```

```

FOR I ← 0 STEP 1 UNTIL 14 DO%           40279000
  BEGIN I ← DDD[478-2×I];%             40280000
  H:=J+14-I;                            40280100
  IF T=@114 THEN                        40281000
    BEGIN DDD[479-2×I]=0;               40281100
    UCHANG:=0;                          %R6140281110
    I:=15;                               40281200
  END ELSE                               40281300
  IF T=@14 OR                            40282000
  DDD[424-I×30],[1:1] THEN              40283100
  BEGIN                                  40283200
    UCHANG:=0;                          %R6140283210
    DDD[478-2×I]+@14;                   40283230
    DDD[479-2×I]=NEXTSLOT;              40283300
    IF NEXTSLOT=0 THEN                  40283400
    BEGIN FI:=I;FJ:=J+15 END;           40283500
    NEXTSLOT:=H;                        40283600
  END ELSE                               40284000
  BEGIN DDD[429-I×30],[1:42]=0;         40285000
  B:=DDD[429-I×30];                     40285005
  IF (C+DDD[423-I×30])≥0 THEN           40285010
  BEGIN DDD[423-I×30]←                  40285020
    =C&C[2:8:10];                      40285030
    UCHANG:=0;                          %R6140285035
    DDD[424-I×30]+0;                    40285135
  END                                    40285140
  ELSE                                   40285150
  DDD[424-I×30]+P(DUP,LOD)              40285160
  AND @0037000000007774;               40285170
  IF C,[2:10]=0 OR                      40285500
  DDD[424-I×30],[44:1] THEN            40285600
  SAVIT;                                40285700
  FOR C:=1 STEP 1 UNTIL B DO            40286000
  BEGIN RADD:=DDD[429-I×30+C];          40287000
  IF RADD GEQ DISKBOTTOM+5 THEN         40290000
  BEGIN                                  40290100
    IF (RADD:=RADD+(RLEN:=DDD[428-I×30])) GTR TM THEN 40290200
    BEGIN RADD:=RADD MOD TM;X:=50 END ELSE X:=0;      40290300
    IF SU[X:=X+5×(D:=RADD DIV MN)],FORTYMILLDISK THEN 40290400
    FORTY:=P(FORTY,DUP,+);              40290500
    X:=((RADD:=RADD MOD MN)-1) DIV FORTY + X;         40290600
    IF (EUM:=SOCK[D]) NEQ 0 THEN LOCKED ELSE BUILDVAIL; 40292050
    FORTY:=40000;                       40292060
  END                                     40292200
  %F SUPER THEN                          40292210
  BEGIN                                  %MC40292210
  STREAM(A:=T,B:=DDD[479-2×I],T:=SUPER:=SPACE(10)); 40292212
  BEGIN DS:=2LIT", "; SI:=LOC A; SI:=SI+1; DS:=7CHR; DS:=LIT"/"; 40292214
  SI:=SI+1; DS:=7CHR; DS:=19LIT" DISK ADDRESS ERROR"; 40292216
  DS:=LIT"+";                            40292218
  END;                                    40292220
  SPOUT(SUPER);                          40292222
  ERROR:=SUPER:=0;                       40292230
  END;                                    %MC40292240
  END;                                    40292250
  END;                                    %MC40292300
  B:=DDD[479-2×I];                       40293010
  S:=(S:=DISKBOTTOM                       %10440293020
  =SCRAMBLE(T,B)),                       %10440293030
  [36:11];                               %10440293040
  C:=V[S,0];                             40293050

```

```

V[S,C+2]:=T; V[S,C+3]:=B; 40293060
V[S,C+4]:=H; 40293070
IF (V[S,0]:=C+3)=60 THEN 40293080
BEGIN V[S,4],[FF]=BYPASS+ 40293090
    BYPASS=2; 40293100
    IF J+15>BYPASS,[CF] THEN 40293101
    DIRECTORYFULL(BYPASS); 40293102
    DISKWAIT([V[S,2]], [CF], 40293110
        60,V[S,1]); 40293120
    CLEAR; 40293140
    END; 40293150
PBCOUNT:= (((("PBD " EQV T) = NOT 0) OR
    ("PUD " EQV T) = NOT 0)) AND
    (B,[CF] = 1)) + PBCOUNT; 40309100
    END; END;% 40309150
SLEEP([NT3],NOT 0); 40309200
DDD:=DDD&P(DUP,AAA)[CTC]; 40310000
AAA:=P INX 0; %SWAP DDD BUFFERS 40311000
DISKWAIT(AAA,480,J); 40311100
IF I = 16 THEN% 40311200
    BEGIN% 40311300
        J = 0;% 40312000
    END;% 40313000
    END;% 40314000
FOR I:= 0 STEP 1 UNTIL DIRMOD=1 DO 40315000
    DISKWAIT([V[I,1]] INX 1, 60, V[I,1]); 40317000
FOR I=0 STEP CHUNKSIZE DIV 64 UNTIL DIRMOD=1 DO 40317200
    BEGIN V[I,0]=0; 40317210
        T=[V[I,0]], [CF]; 40317220
        MOVE(CHUNKSIZE=1, T, T+1); 40317230
        IF T<FENCE THEN FORGETSPACE(T); 40317240
    END; 40317250
    B:=V, [CF]; 40317260
    IF NEXTSLOT#0 THEN 40317270
    BEGIN 40317300
        DISKWAIT(-B, 30, FJ); 40317310
        VR[-2*FI+29]:=H; 40317320
        DISKWAIT(B, 30, FJ); 40317400
        END ELSE NEXTSLOT:=H; 40317500
        FORGETSPACE(B); 40317600
        DDD:=DDD&A[CTC]; FORGETSPACE(AA); 40317610
        IF PBCOUNT=0 OR AUTOPRINT THEN ELSE 40317700
        BEGIN;STREAM(PBCOUNT,X:=X:=SPACE(10)); 40317800
            BEGIN DS=11 LIT" THERE ARE"; X=DI; SI=LOC PBCOUNT; 40320100
                DS=4 DEC; DS=18 LIT" PB FILES ON DISK="; 40320200
                DI=X; DS=3 FILL; 40320300
            END;LI=X; 40320400
        END; 40320500
        END; 40320600
Z=USERDISKBOTTOM; 40320700
XI:=5; DDD[1]:=0; %MC 40321000
$ SET OMIT = NOT(SHAREDISK) 40321100
RI=(NEUP,NEUF+1) DIV 2 + NEUP,NEUF + 2; 40321104
VR:= [M[SPACE(R)]]&R[8:38:10]; 40321105
MOVE(R,VR,[CF]=1,VR); 40321110
VR[0]:=0&R[TONUMENT]; 40321115
$ POP OMIT 40321125
$ SET OMIT = SHAREDISK 40321126
R:=0; VR:=AVTABLE; 40321129
$ POP OMIT 40321130
RADD:=R; RI:=R-1; 40321131
40321135

```



```

NT3:=NEUP,NEUF=1; % DONT USE NT3 BETWEEN HERE AND 40334065
FOR NT2:=0 STEP 1 UNTIL NT3 DO
  BEGIN I+RA+1;RLEN+RL+0;RADD+RADD+(Z-USERDISKBOTTOM)*30;
    FORTY:=(SU[X:=X+5],FORTYMILLDISK+1)*FORTY;
  WHILE (C:=SU[X+(I:=I+1)]),DISKRUNNING AND I LEQ 4 DO
  IF NOT C,OCUPIED THEN
  BEGIN
    IF C,[CF]=0 THEN
    BEGIN
      RA+RA+1;
      C:=0;
      S:=(I+1)*FORTY;
      J+IF X+I=0 THEN
        FORTY=(DISKBOTTOM+5) ELSE FORTY;
      END
      ELSE
      BEGIN AD+M[M[SU[X+I]]],[CF];RA+1; END;
    DO
    BEGIN
      IF C#0 THEN BEGIN S+M[AD],[DA];J+M[1+AD],[DA] END;
      S:=S+(X MOD 50)DIV 5*MN;
      IF J>RLEN THEN RLEN:=J;
      IF X GEQ 50 THEN S:=S+TM;
      IF J GTR 0 AND (NT1:=S-J) GEQ DISKBOTTOM+3 THEN
      IF (Y:=DDD[ABS(R)]),DEND EQL NT1 THEN
      BEGIN DDD[R]:=S&(LO:=Y,DSIZE+J)[TODSIZE];
      IF LO GTR RLEN THEN RLEN:=LO END
      ELSE
      BEGIN
        IF R=AV1 THEN
        BEGIN
          DISKWAIT(A,AV1,Z);Z+Z + AVBLOCK;R+1;
        END;
        DDD[R+1]:= S & J[TODSIZE];RL+RL+1;
        END;
      IF C#0 THEN
      IF M[AD],[CF]#SEVEN7 THEN
      AD+M[AD],[CF] ELSE
      BEGIN
        K:=1;
        WHILE (B*(M[SU[X+I]]INX(K+K+1)))#0 AND K#15 DO
          FORGETSPACE(B);FORGETSPACE(SU[X+I]);
        C+0;
        END;
      END UNTIL C=0;
    END;
  IF (DDD[R],DEND MOD MN)=[(NT1:=5*FORTY)-1] THEN DDD[R],DEND:=NT1+
  NT2*MN; % NT2 = X DIV 5
  RL+RL+1;
  VR[NT2+1]:=0&(SU[X],FORTYMILLDISK+1)[TOSPEED]&RL[TONUMENT]&
  RADD[TOSTARTWRD]&RLEN[TOMAXSIZ]&(NT2#NEUP,[3:15] AND NT2<10)[TOEUNP];
  IF R=AV1 THEN
  BEGIN
    DISKWAIT(A,AV1,Z);
    Z+Z+AVBLOCK;R+1;
  END;
  DDD[R+1]:=400000 DIV(2-SU[X],FORTYMILLDISK)+(X MOD 100)DIV 5*MN+1;
  IF (LO:=RL DIV 4) LSS AVDIFFMIN THEN LO:=AVDIFFMIN ELSE
  IF LO>AVDIFFMAX THEN LO:=AVDIFFMAX;
  IF (RI:=R+LO) GTR AV1 THEN

```

```

40321140
40321200
%024-40321300
40321310
40321400
40321500
%MC40321600
%MC40321700
%MG40321800
%MC40321810
40321910
40322000
%MG40322100
40322150
%MC40322200
%MC40322210
%MC40322220
%MC40322250
%MC40322300
%MC40322400
40322410
40322420
40322425
40322430
40322440
40322442
40322444
%MC40322450
%MC40322460
%MC40322470
%MC40322480
%MC40322600
%MC40322700
40322800
%MC40323000
%MC40323100
%MC40323200
%MG40323300
%MG40323400
%MG40323500
%MC40323600
%MC40323700
%MC40323710
%MC40323800
%MC40323900
40324000
40324102
40324104
40324120
40324200
40324210
40324300
40324400
40324500
40325000
40326000
40327000
40328000
40329000
40330000

```

```

BEGIN
    DISKWAIT(A,AV1,Z);Z+Z+AVBLOCK;
    R1=R-AV1 ;
END;
FORTY:=40000 ;
RADDI=R+1 ;
END;
DISKWAIT(A,AV1,Z);
NT2:=NT3 + 3; % NT2:=NEUP,NEUF+2
FOR NT1:=NT3 STEP -1 UNTIL 0 DO
    IF (NT4:=(NOT SOCK[NT1+20]),[28:20]) # 0 THEN % LOCK OUT THIS EU
    BEGIN EUMASK:=TWO(NT1) OR EUMASK; % TURN ON EU LOCK OUT MASK
        IF NT1 THEN VR[NT1 DIV 2 + NT2],[8:20]:=NT4
        ELSE VR[NT1 DIV 2 + NT2],[28:20]:=NT4;
    END;
VR[0]:=P(DUP,LOD)&EUMASK[TOMAXSIZ];
% SET OMIT = NOT(SHAREDISK)
DISKWAIT(-A,60,USERDISKBOTTOM);
DISKWAIT(EUIO,[CF],EUIO,[8:10],EUIOHOLDER) ;
MOVE(VR[0] AND NUMENTM,[VR[0]],A) ;
DISKWAIT(A,60,USERDISKBOTTOM);
AVS:=(AVS:=IF(AVS:=(Z-USERDISKBOTTOM)*30+R+AVDIFFMIN)>AVSMAX THEN AVSMAX
    ELSE IF AVS LSS AVSMIN THEN AVSMIN ELSE AVS)+30-(IF (AVS:=AVS MOD 30)
    #0 THEN AVS ELSE 30) ;
FORGETSPACE(VR);
    DISKWAIT([HOLDER],[CF],-3,DIRECTORYSEG); % CLOBBERS FJ
% SET OMIT = NOT STATISTICS OR OMIT
    BYPASSBOTTOM:=BYPASS.[CF];
% POP OMIT OMIT
    FORGETSPACE(SU);
% SET OMIT = SHAREDISK
    UNLOCKDIRECTORY;
% POP OMIT
    UNLOCKTOG(USERDISKMASK);
    IF L>1 THEN SPOUT(L); % THERE ARE X PB FILES ON DISK
    MESSAGEBUILDER;
    FOR W+W STEP -2 WHILE ZSF[W]#@14 DO
    BEGIN
        IF W<0 THEN
        BEGIN
            DISKWAIT(-ZSF,[CF],30,ESPADD);
            FORGETESPDISK(ESPADD);
            ESPADD+ZSF[29];
            W+26;
        END;
        FORGETSPACE(DIRECTORYSEARCH(ZSF[W],ZSF[W+1],6));
    END;
    FORGETSPACE(ZSF); FORGETSPACE(SOCK);
    SUSTATUS(A,DDD,0);
END;
PROCEDURE REALFILECLOSE(ALPHA); VALUE ALPHA; INTEGER ALPHA;%
BEGIN ARRAY FIB[*],FPB[*],HEADER[*];%
%% DONT ADD ANY DECLARATIONS BETWEEN "HEADER" AND "KIND" %% WCP
    INTEGER KIND,NBUFS,U,BLEN,CODE,UNLABELED,COBOL,I,J,FNUM;
    REAL MID,FID,R,D,C,FORMS,STATE;
    REAL RCW:=+0,XTRA=-3;
    LABEL PX,PBD;
    LABEL CR,LP,MT,CLOSED,DK,SP,CP,BKUP,PP,PR,DC,CD,CC;
    SWITCH SW+CR,LP,MT,CLOSED,DK,SP,CP,BKUP,PP,PR,DC,CD,BKUP;

```

```

40331000
40332000
40333000
40334000
40334054
40334055
40334056
40334057
40334060
40334065
40334070
40334075
40334077
40334079
40334081
40334085
40334308
40334310
40334315
40334320
40334330
40334335
40334337
40334338
40334500
40334600
40334690
40334700
40334710
40335000
40335990
40336000
40336010
40336100
40338000
40339000
40353100
40353110
40353120
40353130
40353140
40353160
40353170
40353180
40353190
40353200
40353210
%MQ40353300
40356550
40356800
40400000
%R9041000000
41001000
41001500
41002000
41003000
%R9041003100
%P 41004000
41005000
%PB41006000

```

LABEL EOF,ON,DNE,CLEANUP;%	41007000
LABEL EOD;	41007100
LABEL OBJTYPE, DUMMY;	%R6241007150
REAL T1,T2,T3,IOD; ARRAY SEG0[*],SKEL[*]; LABEL L1,L2,L3;	41007200
REAL T,ACCESS;%	41017000
BOOLEAN COMPGO;	41017200
REAL TYPE;	41017300
DEFINE REW=CODE,[47:1]#,%	41018000
KRUNCH=NOT CODE,[42:1]#,%	%R1741018100
REL=CODE,[46:1]#,%	41019000
TIME=CODE,[45:1]#,%	41020000
LOCK=NOT CODE,[44:1]#,%	41021000
PURGE=NOT CODE,[43:1]#;%	41022000
%	41023000
LABEL CLOSEOUT;%	41035000
LABEL EOFIT;%	41036000
CODE=(NOT *P(,ALPHA)),[18:15];%	41038000
ALPHA=P(,ALPHA,LOD),[33:15];%	41039000
FIB=M[ALPHA-3]; FPB=PRT[P1MIX,3];%	41040000
IF (STATE+FIB[5]),[42:1] THEN GO TO CLOSED;%	41041000
NBUFS=FIB[13],[1:9]; FNUM=FIB[4],[13:11];%	41042000
U=FIB[15],[24:6];	41043000
UNLABELED=FIB[4],[2:1];%	41044000
BLN=FIB[18],[3:15];%	41045000
STREAM(S=[FPB[FNUM]],D=[MID]);%	41046000
BEGIN SI:=S; DS:=2WDS; DS:=3OCT; DS:=5OCT; DS:=OCT; END;	41047000
FORMS=FPB[FNUM+3],[42:1];%	41048000
I=FIB[13],[28:10];%	41049000
IF I#0 OR R#0 THEN R=I;	41050000
COBOL=(FIB[13] AND 1)&([FIB],[8:10]=22)[1:47:1]; % COBOL 60 & 68	41051000
IF FIB,[7:1] THEN CHECKJOBORFILEMESS(P1MIX,ALPHA-3,U);	41051620
GO TO SW[KIND=FIB[4],[8:4]];%	41052000
%	41053000
CR:CC:CP:LP:SP:MT:PP:PR:CD;	%R9041054000
OTHERCLOSE(0);	%R9041055000
GO TO CLEANUP;%	41142000
%	41143000
BKUP: TYPE:=FPB[FNUM+3],[43:5]; BACKCLOSE(0);	41144000
DCI;	41186000
CLOSEOUT: STATE,[39:4]+1; TIME+1;%	41187000
CLEANUP:;%	41188000
IF TIME THEN STOPTIMING(FNUM,1023);	%R6241188100
IF NOT STATE,[41:1] THEN%	41189000
IF KIND#2 OR KIND=11 OR KIND#6 AND KIND#9	41190000
% SET OMIT = NOT(PACKETS)	41190099
OR KIND=4	41190100
% POP OMIT	41190101
THEN BEGIN	41190200
% SET OMIT = NOT(PACKETS)	41190249
IF KIND=4 THEN T:=64 ELSE	41190250
% POP OMIT	41190251
T:= (CLOSEMESS AND ((T:=JAR[P1MIX,0])>0 OR T<0 AND COPNMESS))	41190300
OR CLOSEK;	41190500
FILEMESSAGE((	41190600
% SET OMIT = NOT(PACKETS)	41190699
IF PURGE THEN " PRG" ELSE IF LOCK THEN " LOK" ELSE	41190700
% POP OMIT	41190701
" REL")&TINU[U][6:30:18],0,MID,FID,	41190800
IF KIND=2 OR KIND=9 THEN R ELSE 0,	41190900
IF KIND=2 OR KIND=9 THEN D ELSE 0,	41191000

```

C,T);
$ SET OMIT = NOT(PACKETS)
  IF KIND#4 THEN
    BEGIN
$ POP OMIT
  T := SPACE(10)-1; MOVE(10,T,T+1);
  MOVE(ETRLNG,[FPBFNUM],T+1);
  MAKELOG(T,FILESTATS);
  FORGETSPACE(T+1);
$ SET OMIT = NOT(PACKETS)
  END;
$ POP OMIT
  END;
  IF (FIB[5]+STATE).[42:1] THEN FIB[4].[8:4]+3;%
  IF (T+FIB[16].[33:15])#0 THEN%
  BEGIN T+T-1=(IF STATE.[44:1] AND (KIND=2) THEN BLEN ELSE 1);%
    FOR I+0 STEP 1 UNTIL NBUFS=1 DO%
      BEGIN J+M[T].[18:15]-2;%
        FORGETSPACE(T);%
        T+J;%
        M[ALPHA+1]+P(DUP,LOD)&0[2:2:1]&1[25:47:1];%
          &(ALPHA+1)[33:33:15];%
        END;%
        FIB[16]+0;%
      END;%
      IF NOT UNLABELED THEN%
      IF KIND#0 THEN%
      IF (T+M[ALPHA-2].[33:15])#0 THEN%
        FORGETSPACE(T-2);%
        M[ALPHA-2]+P(DUP,LOD)&P(0,XCH)[8:8:10];%
        FIB[6]+FIB[7]+0;%
        GO TO CLOSED;%
      %
DK: DISKCLOSE(0);
GO CLEANUP;%
CLOSED:
  RCW:=XTRA;
END FILE CLOSE;
PROCEDURE LINKUP(TYPE,KEY); VALUE TYPE,KEY; REAL TYPE,KEY;
  BEGIN
  KEY := P(,KEY,LOD) INX 0 -1;
  M[KEY+1]:= (*P(DUP))&TYPE[3:42:6]&(LOGENTRY:=LOGENTRY+1)[25:34:14];
  M[KEY+2] := (*P(DUP)) & (XCLOCK + P(RTR))[3:24:24];
  IF (LOGHOLDER INX 0) = 0 THEN
    BEGIN LOGHOLDER,[CF] := KEY;
      FORK(P(,MAINTLOGGER),0,0,128,1);
    END ELSE M[LOGHOLDER,[FF]],[CF] := KEY;
  M[KEY],[CF] := 0; LOGHOLDER,[FF] := KEY;
  IF (NUMAINTMESS:=NUMAINTMESS+1) > 0 THEN SLEEP([NUMAINTMESS],-0);
  END LINKUP;
PROCEDURE CHECKJOBORFILEMESS(MIX,FIB,U);
  VALUE MIX,FIB,U; REAL MIX,FIB,U;
  BEGIN
  REAL KEY,FNUM;
  IF NOT JAR[MIX,2],[3:1] THEN
    BEGIN
      JAR[MIX,2],[3:1] := 1;
      KEY := GETSPACE(5,73,5)+2;
      M[KEY ] := 0 & MIX[20:43:5];
      M[KEY+1] := JAR[MIX,5],[6:18];

```

```

41191100
41193049
41193050
41193060
41193061
41193100
41193200
41193300
41193400
41193499
41193500
41193501
41193900
41194000
41195000
41196000
41197000
41198000
41199000
41200000
41201000
41202000
41203000
41204000
41205000
41206000
41207000
41208000
41209000
41210000
41211000
41213000
41214000
XR9041215000
41269000
41308000
XR9041309000
XR9041310000
41310100
41310200
41310300
41310400
41310500
41310600
41310700
41310800
41310900
41311000
41311100
41311200
41312000
41312100
41312200
41312300
41312400
41312500
41312600
41312700
41312900
41313000

```

```

M[KEY+2] := JAR[MIX,5];
M[KEY+3] := JAR[MIX,0];
M[KEY+4] := JAR[MIX,1];
LINKUP(12,KEY);
END;
IF FIB#0 THEN IF NOT M[FIB],[6:1] THEN
BEGIN
M[FIB],[6:1] := 1;
FNUM := MEM[FIB] INX 4],[13:11];
KEY := GETSPACE(5,73,5)+2;
M[KEY ] := 0 & MIX[20:43:5]
& ((FNUM DIV ETRLNG)+1)[9:39:9];
M[KEY+1] := JAR[MIX,5],[6:18];
M[KEY+2] := M[(FNUM:= PRT[MIX,3] INX FNUM)+3];
M[KEY+3] := M[FNUM];
M[KEY+4] := M[FNUM+1];
LINKUP(13,KEY);
END;END CHECKJOBORFILEMESS;
PROCEDURE LOGOUTMAINT(B); VALUE B; REAL B;
BEGIN
REAL RCW = +0;
REAL FH = +1, T1 = +2, T2 = +3, T3 = +4, SAVENTRY = +5;
REAL MFID = +6, FID = +7; BOOLEAN FORKED = +8;
INTEGER LASTL = +9, SEGNO = +10, SEGSIZ = +11, LDATE = +12;
LABEL CS,SCAN,NEWLOG,BUILDMESS,EXIT,FINISHUP;
SUBROUTINE FIXCOLDHDR;
BEGIN
M[FH INX 0] := @0000500036000601;
M[FH INX 1] := (XCLOCK+P(RTR)) & LDATE[6:30:18];
STREAM( DATE,X:=FH INX 3);
BEGIN SI:=LOC DATE; DS:=8 OCT; DI:=X; DS:=2 LIT"+#";
SI:=X; SI:=SI+5; DS:=3 CHR;
END;
$ SET OMIT = NOT(SHAREDISK)
STREAM(SYS:=SYSNO,DI:=M[FH INX 4]);
BEGIN
SKIP 9DB; DS:=SET; SKIP 2DB; SYS(SKIP DB); DS:=SET;
END;
M[FH INX 4],[45:1] := 0; % TURN OFF COLD-START BIT
$ POP OMIT
$ SET OMIT = SHAREDISK
M[FH INX 4] := 0 & 72[9:41:7]; % SYSTEM DATA FILE
$ POP OMIT
M[FH INX 7] := (LOGSIZE*6)-1;
END FIXCOLDHDR;
P(0,0,0,0,0,0,0,0,0,0,0,0);
IF FORKED:= B#0 THEN % INDEPENDENT RUNNER
BEGIN IF MROW > 0 THEN SLEEP([MROW],#0);
MROW := ABS(MROW);
LASTL := LOGENTRY;
LOGENTRY := 0;
END ELSE LASTL:=ABS(B)-2;
FID:= "MNTLOG "
$ SET OMIT = NOT(SHAREDISK)
& (SYSNO+17)[42:42:6]
$ POP OMIT
;
STREAM( DATE,C:=[LDATE]); BEGIN SI:=LOC DATE; DS:=8 OCT; END;
T1:=SPACE(335);
IF (FHI=DIRECTORYSEARCH(MFID:="MAINT ",T3:="LOG "

```

```

41313100
41313200
41313300
41313400
41313500
41313600
41313700
41313800
41313900
41314000
41314200
41314300
41314400
41314500
41314600
41314700
41314800
41314900
41316000
41316100
41316200
41316300
41316400
41316410
41316500
41316505
41316510
41316515
41316520
41316525
41316530
41316535
41316540
41316545
41316550
41316555
41316560
41316565
41316567
41316570
41316575
41316580
41316585
41316590
41316595
41317100
41317200
41317300
41317400
41317500
41317600
41317700
41317710
41317719
41317720
41317721
41317730
41317780
41317790
41317800

```

```

$ SET OMIT = NOT(SHAREDISK)                                41317900
      & (SYSNO+17)[24:42:6]                                41318000
$ POP OMIT                                                  41318001
      ,5))=0 THEN                                          41318100
      BEGIN                                               41318200
      FH:=SPACE(30);                                       41318210
      MOVE(30,FH=1,FH);                                    41318220
      M[FH+ 9]:= 1;                                        41318230
      M[FH+10]:= GETUSERDISK(=(M[FH+8]:=LOGSIZE:=1000)); 41318240
CS:  FIXCOLDHDR;                                          41318250
      IF FH,[FF]=0 THEN ENTERUSERFILE(=MFID,T3,FH=1)     41318360
      ELSE DISKWAIT(FH INX 0,30,FH,[FF]);                41318370
      FID:= T3;                                           41318380
      MROW:= M[FH INX 10];                                 41318400
      GO BUILDMESS;                                       41318500
      END;                                                 41318600
      LOGSIZE:= M[FH INX 8];                               41318610
      IF M[FH INX 4],[45:1] THEN FORKED:=FORKED OR 2; % JUST COLD STARTED 41318620
      IF B>0 THEN                                         41318630
      BEGIN                                               41318640
      $ SET OMIT = NOT(SHAREDISK)                          41318649
      STREAM(SYS:=SYSNO,DI=[M[FH INX 4]]);                41318650
      BEGIN SKIP 9DB; DS:=SET; SKIP 2DB; SYS(SKIP DB); DS:=SET; 41318669
      END;                                                 41318670
      M[FH INX 4],[45:1]:=0; % TURN OFF COLD-START BIT   41318671
$ POP OMIT                                                41318672
$ SET OMIT = SHAREDISK                                    41318679
      M[FH INX 4]:= 0 & 72[9:41:7]; % SYSTEM DATA FILE 41318680
$ POP OMIT                                                41318681
      DISKWAIT(=T1,5,MROW:=M[FH INX 10]);                41318740
      MLOG:= SEGNO:= M[T1],[24:15];                       41318760
SCAN: IF MLOG<LOGSIZE-1 THEN                             41318780
      BEGIN                                               41318800
      IF (FORKED AND 2)≠0 THEN GO CS;                     41318810
      IF MLOG≠SEGNO THEN DISKWAIT(=T1,5,MROW);           41318820
      M[T1]:= P(DUP,LOD) & 1[2:47:1];                    41318840
      DISKWAIT(T1,5,MROW);                                41318860
      MLOG:= IF SEGNO<LOGSIZE-1 THEN SEGNO ELSE LOGSIZE-2; 41318880
      GO NEWLOG;                                          41318900
      END;                                                 41318920
      DISKWAIT(=T1,30,MROW+(MLOG:=MLOG+1));              41318940
      IF M[T1]≠ NOT 0 THEN GO SCAN;                       41318960
      MLOG:= MLOG-1;                                       41318980
      LOGENTRY:= M[T1+1],[CF]; LASTL:= M[T1+1],[FF];     41319000
      IF (T3:=LOGHOLDER INX 0) ≠ 0 THEN                  41319020
      WHILE T3≠0 DO                                       41319040
      BEGIN                                               41319060
      IF M[T3]<0 THEN M[T3],[FF]:= LOGENTRY:=LOGENTRY+1  41319080
      ELSE M[T3+1],[25:14]:= LOGENTRY:=LOGENTRY+1;      41319100
      T3:= M[T3] INX 0;                                    41319120
      END;                                                 41319140
      IF LASTL≠0 THEN                                     41319160
      BEGIN                                               41319180
      DISKWAIT(=T1,30,MROW+(SEGNO:=LASTL DIV 30));       41319200
      T3:= (M[T1+(SEGSIZ:=LASTL MOD 30)],[39:9]+1)×5;    41319220
      IF T3>5 THEN IF LASTL+T3 > (T2:=(MLOG+1)×30) THEN 41319240
      BEGIN                                               41319260
      M[T1+SEGSIZ]:= P(DUP,LOD) & 1[2:47:1] &           41319280
      ((T2=LASTL) DIV 5 -1)[39:39:9];                    41319300
      DISKWAIT(T1,30,MROW+SEGNO);                        41319320

```

END;END;	41319340
END;	41319360
M[T1 ] := 5 & 62[3:42:6] &	41319600
(MLOG +(MDELTA#0))[24:33:15] & LASTL[9:33:15];	41319700
M[T1+1] := LDATE & (XCLOCK+P(RTR))[3:24:24];	41319900
M[T1+2] := PATCHLEVEL;	41320000
M[T1+3] := LOGVERSION;	41320100
M[T1+4] := DATE;	41320200
DISKWAIT(T1,5,MROW);	41320220
IF B>0 THEN % CALLED FROM INITIALIZE	41320240
BEGIN	41320250
IF (FORKED AND 2)#0 THEN FIXCOLDHDR;	41320255
DISKWAIT(FH INX 0,30,FH,[FF]);	41320260
GO FINISHUP;	41320270
END;	41320280
NEWLOG;	41320300
IF HOLDFREE=0 THEN SLEEP([TOGGLE],HOLDMASK);	41320310
LOCKTOG(HOLDMASK);	41320320
DISKWAIT(-T1,-30,DIRECTORYTOP=SYSNO);	41320330
SEGNO:= (M[T1+22],[38:10] +1) MOD 1000;	41320340
M[T1+22]:= P(DUP,LOD) & SEGNO[38:38:10];	41320345
DISKWAIT(T1,-30,DIRECTORYTOP=SYSNO);	41320350
UNLOCKTOG(HOLDMASK);	41320355
STREAM(A:=[ACTDATE],B:=SEGNO,C:=[MFID]);	41320360
BEGIN	41320370
SI:=A; SI:=SI+2; DI:=DI+1; DS:=4 CHR; SI:=LOC B; DS:=3 DEC;	41320380
END;	41320390
IF DIRECTORYSEARCH(-MFID,FID,5) # 0 THEN GO NEWLOG;	41320400
M[FH INX 3]:= P(DUP,LOD) & LDATE[12:30:18]; % ACCESSED	41320410
MOVE(10,FH INX 0,T1);	41320420
M[FH INX 1]:= (XCLOCK+P(RTR)) & LDATE[6:30:18];	41320430
M[FH INX 3]:= P(DUP,LOD) & LDATE[30:30:18]; % CREATION	41320440
M[T1+ 4]:= 0 & 1[9:47:1]; % TYPE DATA	41320450
M[T1+ 7]:= (T2:= (MLOG+(MDELTA#0)+2))*6 -1;	41320460
M[T1+ 8]:= T2+10; % TO SIMPLIFY DUMPING	41320470
M[T1+ 9]:= 2;	41320480
M[T1+10]:= 0; MOVE(20,[M[T1+10]],,[M[T1+11]]);	41320490
IF (M[T1+10]:= GETUSERDISK(-T2-10 OR M)) = 0 THEN	41320500
BEGIN	41320550
STREAM(A:=[MFID],C:=T3:=SPACE(5));	41320600
BEGIN	41320700
DS:=18 LIT"-NO USER DISK FOR "; SI:=A; SI:=SI+1;	41320800
DS:=7 CHR; DS:=LIT"/"; SI:=SI+1; DS:=7 CHR; DS:=LIT"+";	41320900
END;	41321000
SPOUT(T3);	41321100
M[T1+10]:= GETUSERDISK(-T2-10);	41321200
END;	41321300
T3:=0; SEGNO:=M[T1+10];	41321350
DO BEGIN	41321400
DISKWAIT(-T1-31,300,MROW+T3);	41321450
DISKWAIT(T1+31,300,SEGNO+T3);	41321500
END UNTIL (T3:=T3+10) GEQ T2;	41321550
ENTERUSERFILE(-MFID,FID,T1-1);	41321600
DISKWAIT(FH INX 0,30,FH,[FF]);	41321650
BUILDMESS;	41321700
MLOG:= MDELTA:= 0;	41321750
M[T1 ] := 5 & 62[3:42:6];	41321800
M[T1+ 1] := LDATE & (XCLOCK+P(RTR))[3:24:24];	41321900
M[T1+ 2] := PATCHLEVEL;	41322000
M[T1+ 3] := LOGVERSION;	41322100

M[T1+ 4]:= DATE;	41322200
M[T1+30]:= NOT 0;	41322300
M[T1+31]:= NUMAINTMESS+100;	41322400
DISKWAIT(T1,32,MROW);	41322450
STREAM(A:=[MFID],TOG:=MFID="MAINT ",M:=MROW,BI=T1);	41322500
BEGIN	41322600
DS:=29 LIT"#NEW MAINTENANCE LOG FILE IS "; SI:=A; SI:=SI+1;	41322700
DS:=7 CHR; DS:=LIT"/"; SI:=SI+1; DS:=7 CHR; DS:=LIT"+";	41322800
TOG(DI:=DI-1; DS:=4 LIT" AT "; SI:=LOC M; DS:=8 DEC; DS:=LIT"+";	41322820
DI:=DI-9; DS:=7 FILL);	41322840
END;	41322900
EXIT;	41323000
SPOUT(T1);	41323100
IF B>0 THEN	41323110
BEGIN	41323120
T1:= GETSPACE(15,9,5)+2;	41323130
FINISHUP:	41323140
MOVE(13,B,T1+2);	41323150
M[T1 ]:= 2;	41323160
M[T1+1]:= LDATE;	41323170
LINKUP(15,T1);	41323180
END ELSE	41323190
IF (T1:=P(.MAINTLOGARRAY,LOD) INX 0)≠0 THEN MOVE(31,T1-2,T1-1);	41323300
T1 := FH INX 0;	41323400
SPOUTMCP(-(T1+4));	41323500
STREAM(KTR:=T1+4);	41323600
BEGIN SI:=KTR;	41323700
4(52(IF SC#"+" THEN SI:=SI+1 ELSE JUMP OUT 2 TO LL));	41323800
LL: KTR:=SI;	41323900
END;	41324000
NT1:= P INX 0;	41324100
M[T1]:= (NT1-T1) DIV 5;	41324200
M[T1+1]:= LDATE;	41324300
\$ SET OMIT = NOT(SHAREDISK)	41324400
DISKWAIT(-(T2:=SPACE(AVS)),AVS,USERDISKBOTTOM);	41324500
STREAM(A:=T2+1,N:=NEUP,NEUF,D:=T1+3);	41324600
\$ POP OMIT	41324601
\$ SET OMIT = SHAREDISK	41324700
STREAM(A:=[AVTABLE[1]],N:=NEUP,NEUF,D:=T1+3);	41324800
\$ POP OMIT	41324801
BEGIN SI:=LOC N; DS:=WDS; DI:=DI-6; SI:=A; SI:=SI+4;	41324900
N(IF SB THEN DS:=SET ELSE DS:=RESET; SI:=SI+8);	41325000
END;	41325100
\$ SET OMIT = NOT(SHAREDISK)	41325200
FORGETSPACE(T2);	41325300
\$ POP OMIT	41325301
M[T1+2] := MCPBASE;	41325400
LINKUP(16,T1);	41325500
IF FORKED THEN BEGIN MROW:=NABS(MROW); KILL([B] INX NOT 1); END;	41325600
END LOGOUTMAINT;	41325700
PROCEDURE MAINTLOGGER(B); VALUE B; REAL B;	41327000
BEGIN	41327100
REAL RCW = +0;	41327200
ARRAY MLA = MAINTLOGARRAY[*];	41327300
REAL KLUDGE = +1, KEY = +2, TRANS = +3, RECS = +4, WT = +5;	41327400
REAL WMCP = +6, WLOG = +7, WD = +8, A = +9, LASTENTRY = +10;	41327500
REAL T1 = +11, T2 = +12, U = +13;	41327600
REAL LOCN= WLOG, NUM= WD;	41327700
LABEL LOGANOTHER,RECYCLE,KILL;	41327800
P(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);	41328000



```

IF MROW > 0 THEN SLEEP([MROW],-0);
MROW := ABS(MROW);
IF (A:=P(,MLA,LOD) INX 0) = 0 THEN
BEGIN
MLA := [M[(A:=GETSPACE(33,9,5)+3))] & 32[8:38:10];
MOVE(31,A-2,A-1);
MLA[30] := NOT 0;
IF MDELTA#0 THEN DISKWAIT(-A,30,MROW+MLOG+1);
END;
LOGANOTHER:
IF M[LOCN:=LOGHOLDER INX 0] < 0 THEN
BEGIN
MOVE(4,LOCN,[TRANS]); KLUDGE := TRANS INX 0;
KEY := -0 & TRANS[26:20:13] & (TRANS,[2:1]+4)[3:42:6] &
TRANS[9:9:9] & TRANS[18:18:2] & TRANS[20:4:5];
TRANS:= TRANSACTION[U:=TRANS,[2:2]+16]&(XCLOCK+P(RTR))[3:24:24];
LOGHOLDER,[CF] := LOCN := [KLUDGE] INX 0;
IF KLUDGE=0 THEN LOGHOLDER,[FF] := LOCN;
END;
NUM := (M[LOCN+1],[39:9]+1) * 5;
IF (LASTENTRY:=(MLOG+1)*30+MDELTA) + NUM > (LOGSIZE-1)*30 THEN
BEGIN
IF MDELTA#0 THEN
BEGIN MLA[31]:=LOGENTRY; DISKWAIT(A,32,MROW+MLOG+1); END;
LOGOUTMAINT(-(M[LOCN+1],[25:14]+1));
LOGENTRY := 0; T1 := LOCN;
WHILE T1 # 0 DO
BEGIN
IF M[T1]<0 THEN M[T1],[FF] := LOGENTRY:=LOGENTRY+1;
ELSE M[T1+1],[25:14] := LOGENTRY:=LOGENTRY+1;
T1 := M[T1] INX 0;
END;
LASTENTRY := 30;
END;
RECYCLE:
IF (T1:=30-MDELTA) > NUM THEN
BEGIN
MOVE(NUM,LOCN+1,[MLA[MDELTA]]);
MDELTA := MDELTA + NUM;
END ELSE
BEGIN
MOVE(T1,LOCN+1,[MLA[MDELTA]]); MLA[31]:=LOGENTRY & LASTENTRY[CTF];
DISKWAIT(A, 32,MROW+(MLOG:=MLOG+1));
LOCN := LOCN + T1;
NUM := NUM - T1;
MDELTA := 0; MOVE(31,A -2, A -1);
IF NUM # 0 THEN GO RECYCLE;
END;
NUMAINTMESS:=NUMAINTMESS - 1;
IF (T1:=M[T2:=LOGHOLDER INX 1]) < 0 THEN % SPOUT MESSAGE FOR RE=
IF (T1,[3:6] AND @76) = 4 THEN % COVERED DISK/DRUM ERR
BEGIN STREAM(A:=TINU[U], R:=RECS,[1:4], X:=KEY,[20:5], S:=WT,[27:6],
B:=[RECS], DSK:=T1,[8:1], DI:=T1:=SPACE(10));
BEGIN SII:=LOC A; SII:=SII+5; DSI:=LIT" "; DSI:=3 CHR;
DSI:=3 DEC; A:=DI; DI:=DI-3; DSI:=2 FILL; DI:=A;
DSI:=14 LIT" RETRIES, MIX=";
DSI:=2 DEC; A:=DI; DI:=DI-2; DSI:=FILL; DI:=A;
SII:=B; DSI:=5 LIT", DA="; CI:=CI+DSK; GO TO DRM;
SII:=SII+1; DSI:=7 CHR; DSI:=7 LIT", SEGS=";
SII:=LOC S; DSI:=2 DEC; SII:=B; SII:=SII+16; GO TO L;

```

```

41328200
41328300
41328400
41328500
41328600
41328700
41328800
41328900
41329000
41329100
41329200
41329300
41329400
41329500
41329600
41329700
41329800
41329900
41330000
41330100
41330200
41330300
41330400
41330500
41330600
41330700
41330800
41330900
41331000
41331100
41331200
41331300
41331400
41331500
41331600
41331700
41331800
41331900
41332000
41332100
41332200
41332300
41332400
41332500
41332600
41332700
41332800
41332900
41332950
41333000
41333002
41333005
41333010
41333015
41333020
41333025
41333030
41333033
41333036
41333037

```

```

DRM: SI:=SI+1; 5(DS:=3 RESET; 41333039
      3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB)); SI:=SI+2; 41333042
L: DS:=4 LIT", R="; 41333045
      16(DS:=3 RESET; 41333048
          3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB)); 41333050
      SI:=SI-5; DS:=5 LIT", IO="; 41333055
      IF SB THEN DS:=2 LIT"4,"; SKIP SB; 41333060
      IF SB THEN DS:=2 LIT"3,"; SKIP SB; 41333065
      IF SB THEN DS:=2 LIT"2,"; SKIP SB; 41333070
      IF SB THEN DS:=2 LIT"1,"; 41333075
      DI:=DI-1; DS:=LIT"+"; 41333080
      END; 41333085
      SPOUTER(T1,PSEUDOMIX[KEY,[20;5]],DISKMSG OR 34); 41333090
END; 41333095
IF (T1:=M[LOGHOLDER] INX 0) = 0 THEN 41333100
BEGIN 41333200
  IF MDELTA # 0 THEN 41333300
  BEGIN 41333400
    MLA[31] := LOGENTRY & LASTENTRY[CTF]; 41333500
    DISKWAIT(A,32,MROW+MLOG+1); 41333600
  END; 41333700
  RECS := 5 & 62[3;42;6] & MLOG[24;33;15] & LASTENTRY[9;33;15]; 41333800
  WT := 0 & (XCLOCK+P(RTR))[3;24;24]; 41333900
  WMCP := PATCHLEVEL; 41334000
  WLOG := LOGVERSION; 41334100
  WD := DATE; 41334200
  DISKWAIT([RECS ] INX 0,5,MROW); 41334300
  T1 := M[LOGHOLDER] INX 0; 41334400
  END; 41334500
  IF M[T2] LSS 0 THEN M[T2],[2;1] := 1 ELSE FORGETSPACE(T2); 41334600
  IF T1 # 0 THEN BEGIN LOGHOLDER,[CF] := T1; GO LOGANOTHER; END; 41334700
KILL; 41334800
  LOGHOLDER,[CF] := 0; MROW := NABS(MROW); 41334900
  IF LOGHOLDER,[9;9]=0 THEN BEGIN FORGETSPACE(A -1); MLA:=0; END; 41335000
  KILL([B] INX NOT 1); 41335100
END MAINTLOGGER; 41335200
PROCEDURE MESSAGETABLEBUILDER; 41430000
BEGIN 41430100
  INTEGER I,I1,I2,TBL,TBLCNT; 41430300
  DEFINE MARKER = "++++++"; 41430400
  LABEL L, START; 41430500
  GO TO START; P(,L); 41430600
L:; 41430700
*** BEGINNING OF OPTION RESERVED WORD TABLE ***** 41430800
"DRAO", "0000", %47% 41430900
"DRBO", "0000", %46% 41431000
"BOJO", "0000", %45% 41431100
"EOJO", "0000", %44% 41431200
"OPEN", "0000", %43% 41431300
"TERM", "NATE", %42% 41431400
"DATE", "0000", %41% 41431500
"TIME", "0000", %40% 41431600
"NOT ", "USED", %39% 41431700
"AUTO", "PRNT", %38% 41431800
"NOT ", "USED", %37% 41431900
"NOT ", "USED", %36% 41432100
"CMPL", "FILE", %35% 41432500
"CLOS", "E000", %34% 41432600
"ERRO", "RMSG", %33% 41432700
"RETO", "0000", %32% 41432800

```

"LIBM",	"SG00",	%31%	41432900
"SCHE",	"DMSG",	%30%	41433000
"SECM",	"SG00",	%29%	41433100
"DSKT",	"OG00",	%28%	41433200
"RELT",	"OG00",	%27%	41433300
"PBDR",	"EL00",	%26%	41433400
\$ SET OMIT = NOT(DEBUGGING OR CHECKLINK)			41433500
"CHEC",	"K000",	%25%	41433600
\$ SET OMIT = DEBUGGING OR CHECKLINK			41433800
0,	0,	%25%	41433900
\$ RESET OMIT			41434000
"DISK",	"MSG0",	%24%	41434100
"DISK",	"LOG0",	%23%	41434200
"LIBE",	"RRO0",	%22%	41434300
"PBDO",	"NLY0",	%21%	41434400
"SAVE",	"PBTO",	%20%	41434500
"RSMS",	"G000",	%19%	41434600
"AUTO",	"UNLD",	%18%	41434700
"RNAL",	"L000",	%17%	41434800
"CODE",	"OLAY",	%16%	41434900
"NOT ",	"USED",	%15%	41435000
"DATA",	"OLAY",	%14%	41435100
"HALT",	"0000",	%13%	41435200
"REMO",	"TE00",	%12%	41435300
"CEME",	"SS00",	%11%	41435400
"BATC",	"HZIP",	%10%	41435500
"NOBA",	"TCHO",	% 9%	41435600
"STOP",	"TEST",	% 8%	41435700
"PNCH",	"LOCK",	% 7%	41435800
"CDON",	"LY00",	% 6%	41435900
"PKTO",	"NLY0",	% 5%	41436000
"SEPA",	"RATE",	% 4%	41436100
"AUTO",	"CE00",	% 3%	41436200
"ARDV",	"ARK ",	% 2%	41436300
"AUTO",	"MESS",	% 1%	41436400
"OPTN",	"0000",	% 0%	41436500
"-000",	"0000",	%STP	41436600
MARKER,			41436700
**** END OF OPTION RESERVED WORD TABLE ****			41436800
**** BEGINNING OF TERMINAL MESSAGE TABLE ****			41440000
0,		% 0%	41440100
"8STACK ",	"60VRFLW",	% 1%	41440200
"8OPRTR ",	"5DS=EDO",	% 3%	41440300
"8FLAG B",	"2IT0000",	% 5%	41440400
"8INVALD",	"6 INDEX",	% 7%	41440500
"8EXPON ",	"60VRFLW",	% 9%	41440600
"8INTGR ",	"60VRFLW",	%11%	41440700
"8DIV BY",	"5 ZERO0",	%13%	41440800
"8EXCESS",	"5 TIME0",	%15%	41440900
"8INVALD",	"6 ADRSS",	%17%	41441000
"8UNEXP ",	"610 ERR",	%19%	41441100
"8MISSIN",	"8G DISK",	%21%	41441200
"5 FILE0",	0,		41441300
"8FILE U",	"8NOPENE",	%25%	41441400
"1D00000",			41441500
"8INVALI",	"5D EQJO",	%28%	41441600
"8INVALI",	"5D PRLO",	%30%	41441700
"8MEMORY",	"8 PARIT",	%32%	41441800
"1Y00000",			41441900
"8OPRTR ",	"5ES=ED.",	%35%	41442000

"8INVALID", "8 ARRAY",	%37%	41442100
"8 SIZE ", "3IDN...",		41442200
"8INVALID", "8 INPUT",	%41%	41442300
"6 DATUM",		41442400
"8TYPE M", "8ISMATC",	%44%	41442500
"8H READ", "4STMT...",		41442600
"8OUT OF", "5 DATA",	%48%	41442700
"8NON-CO", "8NFORMA",	%50%	41442800
"8L ARRA", "2YS...",		41442900
"8NON-SQ", "8UARE M",	%54%	41443000
"5ATRIX",		41443100
"8NEARLY", "8 SINGU",	%57%	41443200
"8LAR MA", "4TRIX...",		41443300
"8USER D", "4S-ED...",	%61%	41443400
"8INVALID", "8 DYNAM",	%63%	41443500
"8IC DIA", "1L...",		41443600
"8TANK O", "6VRFLOW",	%67%	41443700
0,		41443800
"8PARITY", "6 ERROR",	%70%	41443900
"8DIMENS", "8ION SI",	%72%	41444000
"6ZE ERR",		41444100
"8INVALID", "8 FILE ",	%75%	41444200
"4NAME...",		41444300
"8INVALID", "8 BLOCK",	%78%	41444400
"5 EXIT",		41444500
"8OUT OF", "4 MEM...",	%81%	41444600
"8EXCESS", "8 IO TI",	%83%	41444700
"2ME...",		41444800
"8INVALID", "8 LINKE",	%86%	41444900
"6D TAPE",		41445000
"8TIMELI", "8MIT EX",	% 89 %	41445100
"8CEEDED", "1 .....",		41445200
0,	%STP	41449700
MARKER,		41449800
**** END OF TERMINAL MESSAGE TABLE ****		41449900
**** BEGINNING OF KEYIN MESSAGE TABLE ****		41450000
COMMENT		41450100
KEYIN MESSAGE TABLE ENTRIES -		41450200
EACH TABLE WORD IS CONFIGURED AS FOLLOWS:		41450300
[ 616 ] = - MIX OR INFO CODE -		41450400
0 = INFO MESSAGE ONLY		41450500
1 = MIX OR INFO MESSAGE		41450600
2 = MIX MESSAGE ONLY		41450700
[12:12] = TWO LETTER KEYBOARD MESSAGE		41450800
[24:6 ] = - KEYIN PROCEDURE TO BE CALLED -		41450900
0 = PROCEDURE KEYINO ( DIRECT CALL )		41451000
1 = PROCEDURE KEYIN1 ( DIRECT CALL )		41451100
2 = PROCEDURE KEYIN2 ( INDEPENDENT RUNNER )		41451200
[33:1 ] = 1 FOR ALLOWABLE STANDARD RJE REQUESTS		41451300
[34:2 ] = - MIXCODE ( FOR MIX MESSAGES ) -		41451400
1 = JOB SHOULD BE WAITING FOR THIS INPUT		41451500
2 = JOB SHOULD BE RUNNING, BUT NOT NECESSARILY		41451600
WAITING		41451700
3 = JOB NEED NOT BE RUNNING		41451800
[36:12] = LABEL NUMBER ( SWITCH LOCATION IN PROCEDURE )		41451900
END OF COMMENT:		41452000
"2AX0101", %AX% SPO INPUT TO JOB		41452100
"2IL0102", %IL% INPUT LABEL		41452200
"2UL0103", %UL% UNKNOWN LABEL		41452300
"1QT0204", %QT% QUIT PROCESSING		41452400

"20U0105",	%OU% OUTPUT UNIT	41452500
"1WY0106",	%WY% LIST REASON FOR WAIT	41452600
"0RY0007",	%RY% READY UNIT OR LINE	41452700
"2DS0208",	%DS% TERMINATE JOB	41452800
"2SD0209",	%SD% TERMINATE WITHOUT REMOVING DECK	41452900
"ORS0010",	%RS% SEND SPO MSG TO ASSIGNED STATIONS	41453000
"OSS0011",	%SS% STATION TO STATION MESSAGE	41453100
\$ SET OMIT = NOT(DUMP OR DEBUGGING)		41453200
"1DP0212",	%DP% MEMORY DUMP	41453300
\$ SET OMIT = DUMP OR DEBUGGING		41453400
"1DP7212",	%DP% MEMORY DUMP	41453500
\$ SET OMIT = NOT DEBUGGING		41453600
"ODD0013",	%DD% DISK DUMP	41453700
"ODB0014",	%DB% DISKBUG	41453800
\$ SET OMIT = DEBUGGING		41453900
"ODD7013",	%DD% DISK DUMP	41454000
"ODB7014",	%DB% DISKBUG	41454100
\$ RESET OMIT		41454200
"2ST0215",	%ST% STOP EXECUTION	41454300
"OCM0016",	%CM% CHANGE MCP	41454400
"OMF0017",	%MF% SET UP TO MOVE FENCE AT NEXT H/L	41454500
"OSV0018",	%SV% SAVE UNIT OR SCHEDULE LINE	41454600
"OCL0019",	%CL% CLEAR UNIT OR LINE	41454700
"1BK0320",	%BK% BREAK FOR SPO	41454800
"2OK0121",	%OK% PERMIT PROCESSING TO CONTINUE	41454900
"2FM0122",	%FM% FORMS OK	41455000
"2FR0123",	%FR% FINAL REEL (COBOL)	41455100
"2OF0124",	%OF% OPTIONAL FILE=COBOL, OK FILE=LIBMAIN	41455200
"2IF0525",	%IF% IGNORE IN=USE FILE	41455300
"1**77**",	***% END OF FIRST KEYIN PROCEDURE CALLS	41459900
"ODT1001",	%DT% ENTER CURRENT DATE	41460000
"OWD1002",	%WD% PRINT CURRENT DATE	41460100
"OTR1003",	%TR% TIME RESET	41460200
"OWT1004",	%WT% PRINT CURRENT TIME	41460300
"OTF1005",	%TF% PRINT CORE FACTOR	41460400
"OSF1006",	%SF% SET CORE FACTOR	41460500
"OWM1007",	%WM% PRINT CURRENT MCP NAME	41460600
"OCX1008",	%CX% SPO INPUT TO CANDE	41460700
"OCE1009",	%CE% STARTS CANDE	41460800
"OCC1010",	%CC% CONTROL CARD (SEE 16037780 FOR QMARK)	41460900
"OOL1011",	%OL% PRINT OUTPUT LABEL OF UNIT	41461000
"OPB1012",	%PB% START PRINTER BACK UP	41461100
"OBS1013",	%BS% SET BACK UP SPO	41461200
"OUS1014",	%US% RESET BACK UP SPO	41461300
"OSC1015",	%SC% TYPE SPO CONSOLES	41461400
"ORN1016",	%RN% SET PSEUDO READERS	41461500
"OLD1017",	%LD% START LOAD CONTROL	41461600
"ORD1018",	%RD% REMOVE PSEUDO DECK	41461700
\$ SET OMIT = NOT PACKETS		41461800
"ORP1018",	%RP% REMOVE PACKET	41461900
\$ SET OMIT = PACKETS		41462000
"ORP7018",	%RP% REMOVE PACKET	41462100
\$ RESET OMIT		41462200
"OED1019",	%ED% ELIMINATE PSEUDO DECK	41462300
\$ SET OMIT = NOT STATISTICS		41462400
"OSI1020",	%SI% SET STATISTICS INTERVAL TIMER	41462500
\$ SET OMIT = STATISTICS		41462600
"OSI7020",	%SI% SET STATISTICS INTERVAL TIMER	41462700
\$ SET OMIT = NOT AUXMEM		41462800
"OLA1021",	%LA% LIST AUXMEM FILES	41462900

"OCA1022",	%CA% CHANGE AUXMEM FILES	41463000
\$ SET OMIT = AUXMEM		41463100
"OLA7021",	%LA% LIST AUXMEM FILES	41463200
"OCA7022",	%CA% CHANGE AUXMEM FILES	41463300
\$ RESET OMIT		41463400
"OSQ1023",	%SQ% DISK SQUASH	41463500
\$ SET OMIT = NOT SEPTICTANK		41463600
"OCS1024",	%CS% CREATE SEPTIC TANK	41463700
"OHS1025",	%HS% HALT SEPTIC TANK	41463800
\$ SET OMIT = SEPTICTANK		41463900
"OCS7024",	%CS% CREATE SEPTIC TANK	41464000
"OHS7025",	%HS% HALT SEPTIC TANK	41464100
\$ RESET OMIT		41464200
"1**77**",	***% END OF SECOND KEYIN PROCEDURES	41469900
"OMX2001",	%MX% LIST JOBS CURRENTLY RUNNING	41470000
"OR02002",	%RO% RESET OPTION BIT	41470100
"OS02003",	%SO% SET OPTION BIT	41470200
"1TS2304",	%TS% TYPE OUT SCHEDULE [NAMES IN SHEET]	41470300
"2PS2305",	%PS% CHANGE PRIORITY IN SCHEDULE	41470400
"1ES2306",	%ES% ELIMINATE JOB FROM SCHEDULE	41470500
"1XS2307",	%XS% EXECUTE JOB FROM SCHEDULE	41470600
"2TI2208",	%TI% PRINT TIME USED BY JOB	41470700
"2PR2209",	%PR% CHANGE PRIORITY OF JOB	41470800
"OLF2010",	%LF% LIST FILES FOR USER	41470900
"OLC2011",	%LC% LIST FILES FOR CREATOR	41471000
"OLS2012",	%LS% LIST FILES SECURITY	41471100
"OEX2013",	%EX% LIST FILES EXPIRED	41471200
"OPD2014",	%PD% DIRECTORY SEARCH FOR FILES	41471300
"2OT2215",	%OT% PRINT VALUE OF PRT CELL	41471400
"2IN2216",	%IN% ENTER VALUE IN PRT CELL	41471500
"2IT2217",	%IT% KEYBRD INTERRUPT FOR ONLINE MAINT.	41471600
"OT02018",	%TO% TYPE OPTION	41471700
"OP02019",	%PO% PRINT SPECIFIC OPTION	41471800
"OPG2020",	%PG% PURGE A TAPE	41471900
\$ SET OMIT = NOT AUXMEM		41472000
"1AU2621",	%AU% PRINT AUXMEM IN USE	41472100
\$ SET OMIT = NOT(AUXMEM OR MONITOR)		41472110
"OMS2022",	%MS% SET OR RESET SYSTEM MONITOR	41472200
\$ SET OMIT = AUXMEM		41472300
"1AU7321",	%AU% PRINT AUXMEM IN USE	41472400
\$ SET OMIT = AUXMEM OR MONITOR		41472410
"OMS7022",	%MS% SET OR RESET SYSTEM MONITOR	41472500
\$ RESET OMIT		41472600
"OLN2023",	%LN% INITIATE LOGGING ROUTINE	41472700
"OCD2024",	%CD% PRINT PSEUDO DECKS ON DISK	41472800
\$ SET OMIT = NOT PACKETS		41472900
"OPP2024",	%PP% PRINT PACKETS ON DISK	41473000
\$ SET OMIT = PACKETS		41473100
"OPP7024",	%PP% PRINT PACKETS ON DISK	41473200
\$ RESET OMIT		41473300
"OFE2025",	%FE% ENTER COMMENTS INTO MAINT.LOG	41473400
"1CU2226",	%CU% PRINT SYSTEM CORE USAGE	41473500
\$ SET OMIT = NOT STATISTICS		41473600
"OSY2027",	%SY% CREATE NEW STATISTICS FILE	41473700
\$ SET OMIT = STATISTICS		41473800
"OSY7027",	%SY% CREATE NEW STATISTICS FILE	41473900
\$ RESET OMIT		41474000
"1OC2228",	%OC% ENTER OPERATOR COMMENT IN LOG	41474100
"ORW2029",	%RW% REWIND TAPE	41474200
"OCI2030",	%CI% CHANGE INTRINSICS	41474300

"1SM2231",		%SM% START MIX MESSAGES	41474400
"2CT2232",		%CT% CHANGE TIME LIMITS FOR JOB	41474500
"2XT2233",		%XT% EXTEND TIME LIMITS FOR JOB	41474600
"2TL2234",		%TL% PRINT IO AND PROCESSOR TIME LIMITS	41474700
"1WU2235",		%WU% TYPE USERS ID'S OF LINES IN MIX	41474800
"0XD2036",		%XD% CREATE BADISK AREA	41474900
"OMR2037",		%MR% RESERVE DISK FOR NO USER DISK	41475000
"OWI2038",		%WI% PRINT CURRENT INTRINSIC NAME	41475100
"OMC2039",		%MC% MAKE COMPILER FILE	41475200
\$ SET OMIT = NOT PACKETS			41475300
"OPC2040",		%PC% PACKET COUNT	41475400
\$ SET OMIT = PACKETS			41475500
"OPC7040",		%PC% PACKET COUNT	41475600
\$ RESET OMIT			41475700
"OHD2041",		%HD% HOW MUCH (AVAILABLE) DISK	41475800
"2SA2242",		%SA% SEG & REL ADDR OF RUNNING PROG	41475900
"1←←0000",		%←←% END OF TABLE	41479700
MARKER,			41479800
**** END OF KEYIN MESSAGE TABLE ****			41479900
**** BEGINNING OF CC RESERVED WORD TABLE ****			41480000
"UNLOCK ",	22 ,		41480100
"USE ",	23 ,		41480200
"LOCK ",	24 ,		41480300
"FREE ",	25 ,		41480400
"PUBLIC ",	26 ,		41480500
"PACKET ",	27 ,		41480700
"USER ",	28 ,		41480900
"RUN ",	29 ,		41481000
"R ",	29 ,		41481100
"COMPILE",	30 ,	% SWITCH TYPE(CONTROLCARD)%	41481200
"C ",	30 ,		41481300
"EXECUTE",	31 ,	% "RUN" = "LABEL"	41481400
"EX ",	31 ,		41481500
"DUMP ",	32 ,		41481600
"UNLOAD ",	33 ,		41481700
"ADD ",	34 ,		41481800
"LOAD ",	35 ,		41481900
"REMOVE ",	36 ,		41482000
"CHANGE ",	37 ,		41482100
"UNIT ",	38 ,		41482200
"PACKEND",	39 ,		41482400
"END ",	39 ,		41482600
\$ SET OMIT = NOT PACKETS			41482700
"WAIT ",	40 ,		41482800
\$ POP OMIT			41482900
"DATA ",	41 ,		41483000
"LABEL ",	42 ,		41483100
"SET ",	43 ,		41483200
"RESET ",	44 ,		41483300
"FILE ",	47 ,		41483400
"EXPIRED",	48 ,		41483500
"ACCESSD",	49 ,		41483600
"PROCESS",	50 ,	% A STORE NEAR THE END OF PCC	41483700
"IO ",	51 ,	% MAKES USE OF THE ORDER AND VALUES	41483800
"PRIORIT",	52 ,	% OF "PROCESS" THRU "SAVE",	41483900
"COMMON ",	53 ,		41484000
"CORE ",	54 ,		41484100
"STACK ",	55 ,		41484200
"SAVE ",	56 ,		41484300
"ALGOL ",	60 ,		41484400

"XALGOL "	61 ,		41484500
"FORTRAN"	62 ,		41484600
"TSPOL "	63 ,		41484700
"BASIC "	64 ,		41484800
"COBOL68"	65 ,		41484900
"WITH "	66 ,		41485000
"COBOL "	67 ,		41485100
"LIBRARY"	68 ,		41485200
"SYNTAX "	69 ,		41485300
"FROM "	70 ,		41485400
"TO "	71 ,		41485500
"FORM "	78 ,	% SWITCH D(PCC)	41485600
"NO "	79 ,	% "FORM"="SPECIAL"	41485700
"DISK "	80 ,		41485800
"TAPE "	81 ,		41485900
"PUNCH "	82 ,		41486000
"PRINT "	83 ,		41486100
"BACK "	85 ,		41486200
"SPECIAL"	89 ,		41486300
			41486400
"SERIAL "	86 ,		41486500
"UPDATE "	87 ,		41486600
"SPO "	88 ,		41486700
"PAPER "	84 ,		41486800
"EU "	91 ,		41486900
"SLOW "	92 ,		41487000
"B6500 "	93 ,		41487100
"FAST "	94 ,		41487200
"COPY "	95 ,		41487300
"MAXIMUM"	96 ,		41487400
"FREEF "	97 ,		41487500
"FIXED "	98 ,		41487600
"SENSITI"	100 ,		41487650
"PROTECT"	99 ,		41487700
"LATEST "	101 ,		41487900
"CC "	14 ,	% CC MUST EQUAL QUEST %	41488900
0 ,	0 ,		41489000
MARKER,			41489100
**** END OF CC RESERVED WORD TABLE ****			41489200
**** BEGINNING OF LBMESS MESSAGE TABLE ****			41490000
" , % 0			41490100
"LOADED " , % 1			41490200
"DUMPED " , % 3			41490300
"CHANGED" , % 5			41490400
"REMOVED" , % 7			41490500
"MC=ED " , % 9			41490600
"FIXED " , % 10			41490700
"RESET " , % 11			41490800
"SET " , % 12			41490900
"ACCESSE" , "D" , % 13			41491000
"NOT ON " , "DISK" , % 15			41491100
"NOT ON " , "TAPE" , % 17			41491200
"NOT EXE" , "CUTABLE" , " CODE" , % 19			41491300
"NOT A C" , "OMPILER" , " " , % 22			41491400
"SYSTEM " , "FILE" , % 25			41491500
"TAPE PA" , "RITY" , % 27			41491600
"DUP FIL" , "E" , % 29			41491700
"NO USER" , " DISK" , % 31			41491800
"UNEXPED" , " EOF" , % 33			41491900
"DISK PA" , "RITY" , % 35			41492000



```

"BAD NAM", "E"      ", % 37          41492100
"INV REC", " SIZE" ", % 39          41492200
"INVALID", " USER" ", % 41          41492300
"BAD HEA", "DER"   ", % 43          41492400
"IN USE", "        ", % 45          41492500
"INEXECU", "TABLE I", "PC CODE", " FILE" ", % 46  41492600
"AUTO=ZI", "PPED"  ", % 50          41492700
"CHANGED", " TO "  ", % 52          41493010
"MC=ED T", "O "    ", % 54          41493020
"EXTRA R", "ECORDS", " % 56          41493030
"        ", "        ", % 58 TAKE UP SOME SPACE 41493040
"SENSITI", "VE"   ", % 60          41493050
"BEING B", "LANKED", " % 62          41493060
"NOT LAT", "EST VER", "SION" ", % 64      41493070
MARKER;                                41493100
**** END OF LBMESS MESSAGE TABLE ***** 41493200
**** END OF RESERVED WORD AND MESSAGE TABLES ***** 41500000
START;                                  41500100
TBL:=I2:=M(PC, MESSAGETABLEBUILDER), [CF]+2; 41500200
WHILE M[TBL:=TBL+1]#MARKER DO; % SEARCH FOR END OF OPTION TBLE 41500300
I1:=TBL; TBL:=TBL+1;                    41500400
FOR I:=2 STEP 1 UNTIL MESSAGETABLESIZE DO 41500500
WHILE M[TBL:=TBL+1] # MARKER DO;        41500600
I:=I1-I2; I1:=(TBL+2)-I1;                41500700
STREAM(A:=I DIV 60, B:=(I:=(I MOD 60)), C:=I1 DIV 60, 41500800
      D:=(I1:=(I1 MOD 60)), E:=I2);      41500900
BEGIN                                    41501000
  SI:=E; DI:=E;                           41501100
  A(60(SI:=SI+4; DS:=4 CHR));              41501200
  B(SI:=SI+4; DS:=4 CHR);                  41501300
  C(DS:=60 WDS);                           41501400
  D(DS:=WDS);                               41501500
END;                                       41501600
TBL:=I2;                                  41501700
FOR TBLCNT:=0 STEP 1 UNTIL (MESSAGETABLESIZE-1) DO 41501800
BEGIN                                     41501900
  WHILE M[TBL:=TBL+1]#MARKER DO; I:=TBL-I2; 41502000
  MESSAGETABLE[TBLCNT]:=GETUSERDISK((I+29) DIV 30)&I[8:38:10]; 41502100
  DISKWAIT(I2, I, MESSAGETABLE[TBLCNT], [22:26]); 41502300
  I2:=TBL:=TBL+1;                          41502400
END;                                       41502500
END BUILDING TABLES;                    41502600
PROCEDURE ENTERSYSFILE(N); VALUE N; REAL N; 41600000
%                                          41600100
BEGIN                                     41600200
  REAL A, J, WC, MFID, DISK;               41600300
  ARRAY ODD[*];                             41600400
  LABEL RETURN, EXIT;                       41600500
%                                          41600600
  IF N=1 THEN                               41600700
  BEGIN                                     41600800
    MFID := "LIBMAIN"; J := 1;              41600900
  END ELSE                                  41601000
  IF N=2 THEN                               41601100
  BEGIN                                     41601200
    MFID := "LDCNTRL";                     41601300
  END ELSE                                  41601400
  IF N=3 THEN                               41601500
  BEGIN                                     41601600
    MFID := "PRNPBT ";                     41601700

```

```

END ELSE
GO EXIT;
%
DISK I= "DISK ";
IF (A:=DIRECTORYSEARCH(MFID,DISK,5)) # 0 THEN
BEGIN
M[A INX 2] I= MCP;
M[A INX 5] I= M[A INX 6] I= @14;
DISKWAIT(A,[CF],30,A,[FF]);
GO RETURN;
END;
DDD I= [M[A I= SPACE(WC I= 181+30×J)]]&WC[8:38:10];
MOVE(WC,A=1,A);
STREAM(DATE,D:=A+3);
BEGIN
SI:=LOC DATE; DS:=8 OCT;
DI:=D; DS:=2 LIT"+#";
SI:=D; SI:=SI+5; DS:=3 CHR;
END;
DDD[ 0] I= @3600036000101;
DDD[ 1] I= (XCLOCK+P(RTR))&DDD[3][6:30:18];
DDD[ 2] I= MCP;
DDD[ 4],[9:2] I= 3;
DDD[ 5] I= DDD[6] I= @14;
DDD[ 7] I= 4+J;
DDD[ 9] I= 1;
DDD[10] I= PETUSERDISK((DDD[8] I= 5+J)&1[2:47:1],1);
DDD[31] I= 3=J;
DDD[32] I= DDD[38] I= 2;
DDD[33] I= 4=J;
DDD[34] I= 22;
DDD[35] I= 2+J+J;
DDD[36] I= 6;
DDD[37] I= IF J THEN =1 ELSE 1;
DDD[47] I= DDD[48] I= @3777777777777777; % TIME LMT
DDD[49] I= IF J THEN (SHEETMAX) DIV 2 ELSE 0; % PRIORITY
DDD[51] I= IF J THEN 64 ELSE 4; % CORE EST
DDD[52] I= IF J THEN 200 ELSE 150; % STACKSIZ
DDD[61] I= @0000012600001011
&(IF J THEN 35 ELSE IF N=2 THEN 23 ELSE 19)[24:38:10];
DDD[62] I= @0024101100000000;
DDD[122=30×J] I= @0000220000200001;
DDD[169=30×J] I= FLAG(@2740010000100000);
IF NOT J THEN
STREAM(C:=N=2, D:=[DDD[91]]);
BEGIN
CI:=CI+C; GO L1;
DS:=40 LIT
"012CONTROLDECK 1A022BACK=UPOF DECK1B00";
GO L2;
L1: DS:=40 LIT
"012PRINTERBACK=UP1A0220000000PRINTER1B00";
L2:
END;
ENTERUSERFILE(MFID,DISK,A=1);
DISKWAIT(A+31,WC=31,DDD[10]);
RETURN;
FORGETSPACE(A);
EXIT;
END ENTERSYSFILE;

```

```

41601800
41601900
41602000
41602100
41602200
41602300
41602400
41602500
41602600
41602700
41602800
41602900
41603000
41603100
41603200
41603300
41603400
41603500
41603600
41603700
41603800
41603900
41604000
41604100
41604200
41604300
41604400
41604500
41604600
41604700
41604800
41604900
41605000
41605100
41605200
41605300
41605400
41605500
41605600
41605700
41605800
41605900
41606000
41606100
41606200
41606300
41606400
41606500
41606600
41606700
41606800
41606900
41607000
41607100
41607200
41607300
41607400
41607500
41607600
41607700

```

```

COMMENT ARTN RETURNS ALL STORAGE FOR AN N-DIMENSIONAL ARRAY A;% 42473000
PROCEDURE ARTN(A,N); VALUE A,N; ARRAY A[*]; INTEGER N;% 42474000
BEGIN INTEGER I;% 42475000
  ARRAY LOC = +2[*]; 42475100
  INTEGER INDEX = NT1, 42475200
  WORD = NT2, 42475300
  COUNT = NT3, 42475400
  DRUM = NT4, 42475500
  X = NT5; 42475600
  LABEL L; 42475700
  WAITSTORE(P1MIX); 42476000
  IF A,[18:15]#0 THEN A+M[A,[18:15]]; 42476100
  IF N>1 THEN DO ARTN(A[I],N-1) UNTIL (I+I+1)≥A,SIZE; 42477000
  X := N#(-1); N := A INX 0; 42478000
  IF A,PBIT THEN 42478200
    BEGIN I+M[N-1],[FF]; 42478400
    IF X THEN FORGETSPACE(N); 42478600
    END ELSE I+N; 42478800
  IF I GTR 511 THEN 42479000
    BEGIN COMMENT "DISKRTN" ROUTINE CALLED HERE...; 42479100
$ SET OMIT = NOT(AUXMEM) 42479109
  IF I,[33:3] = 7 THEN 42479110
    BEGIN COMMENT AUXILIARY MEMORY RETURN AND COUNTER DECREMENT; 42479120
    AUXDATA[P1MIX] := *P(DUP)-A,[8:6]-1; 42479130
    FORGETAUXILIARYSPACE(A,SIZE,I); P(XIT); 42479140
  END; 42479150
$ POP OMIT 42479151
  P(DALOC[P1MIX,*]); 42479200
  COUNT := TWO(24-6*(I,[39:9] DIV 100)); 42479300
  X := (INDEX := 0&I[41:33:6])=1; 42479400
  IF (WORD := LOC[INDEX],[18:30]-COUNT)=0 THEN 42479500
    BEGIN LOC[INDEX] := 0; 42479600
  L1: IF P(LOC[0],[FF],DUP)#0 THEN 42480300
    IF LOC[POLISH-1]<0 THEN P(XIT); 42480400
    LOC[0] := (*P(DUP))&INDEX[CTF]; 42480500
  END ELSE BEGIN 42480600
    LOC[INDEX] := (*P(DUP))&WORD[18:18:30]; 42480700
    IF (WORD DIV COUNT),[42:6]=0 THEN 42480800
      IF LOC[X]<0 THEN GO TO L1; 42480850
  END END END ARTN; 42480900
COMMENT ASR IS THE ALGOL STORAGE RETURN COMMUNICATE;% 42481000
PROCEDURE ASR; BEGIN INTEGER I,BCNTR; ARRAY AIT[*]; REAL TEMP;% 42482000
  LABEL L1;% 42483000
  REAL MOTHER; ARRAY DESC[*];% 42484000
  WHILE (AIT+PRT[P1MIX,AITNDX]),PBIT=0 DO% 42485000
    MAKEPRESENT([PRT[P1MIX,AITNDX]] INX 0);% 42486000
  MEMORY[AIT INX NOT 1],[2:1]-1;% 42487000
  I:=AIT[0]+1; 42488000
  IF (BCNTR:=PRT[P1MIX,CURBLKCNTR]) LEQ 0 THEN 42488100
    BEGIN TERMINATE(P1MIX); 42488200
    TERMINALMESSAGE(78); 42488300
  END; 42488400
  WHILE (TEMP+AIT[I+I-1]),BLKCNTR≥BCNTR% 42489000
    DO BEGIN DESC+MEMORY[MOTHER+TEMP,MOM];% 42490000
    IF TEMP,[1:2]#1 THEN % CHECK FOR FAULT ENTRY 42490100
      IF TEMP,FILEBIT THEN% 42491000
        IF TEMP,[33:15]=2 THEN BEGIN% 42492000
          FILECLOSE((MOTHER+3)&((DESC[4],[25:2]*2)*@12) 42493000
            [18:33:15]); 42494000
        END; 42495000

```

```

GO TO L END ELSE% 42496000
BEGIN FILECLOSE((DESC INX 5)& 42497000
((MEDESC[2] INX 4],[25:2]=2)*@12) 42498000
[18:33:15]); 42499000
FORGETSPACE(DESC INX 0);% 42500000
END ELSE ARTN(DESC,TEMP,DIMENSIONS);% 42501000
MEMORY[MOTHER]=0;% 42502000
L: 42503000
END;% 42504000
AIT[0]=1;% 42505000
PRTIP1MIX,CURBLKCNTJ+BCNTR=1;% 42506000
IF I>0 THEN DO%%WIPE OUT BAD LABELS IN FAULT CELLS 42506100
IF AIT[I],[1:2]=1 THEN 42506200
IF M[AIT[I],MOM],BLKCNTJ>BCNTR THEN 42506300
M[AIT[I],MOM]=0 UNTIL (I+I=1)S0; 42506400
MEMORY[AIT INX NOT 1],[2:1]=0;% 42507000
END ASR;% 42508000
SAVE REAL PROCEDURE COREND; FORWARD; 42509000
PROCEDURE INTERRUPT(TYPE); VALUE TYPE; REAL TYPE; 42510000
BEGIN LABEL FLAGBIT,INVALIDINDEX,EXPUNDERFLOW,DIVIDEBYZERO; 42511000
LABEL XYT; 42511500
SWITCH SW=FLAGBIT,INVALIDINDEX,EXPUNDERFLOW,DIVIDEBYZERO; 42512000
ARRAY TOP=-5[*]; 42513000
REAL FLAGTESTER=-3; 42513500
REAL MOM,SIZE,ALOC,I; 42514000
REAL RCW=+1,RCWL=+2,SAVIT=+4; NAME A=+3; 42515000
REAL R=+1,S=+2,Y=+3; 42516000
% 42517000
BOOLEAN SUBROUTINE DOUBLEPRECISION; 42517010
BEGIN R=MIS+PRT[P1MIX,8] INX 0; %IRCW 42517020
STREAM(R+(R INX 0)&R[30:10:2],Y+[Y]); %GET OP CODE 42517030
BEGIN SI=R; SI=SI-2; DI=DI+6; DS=2 CHR END; 42517040
DOUBLEPRECISION=Y,[45:3]=5; 42517050
END; 42517060
CHECKSTACKSPACE;% %WF 42517100
GO TO SW[TYPE]; 42518000
% 42519000
FLAGBIT: 42520000
SAVIT=TOP; 42521000
NT1=ANALYSIS; 42522000
IF SYLLABLE,[41:7]#@35 THEN 42523000
IF SYLLABLE,[45:3]#0 THEN 42524000
BEGIN ERRORFIXER(16); TERMINATE(P1MIX); TERMINALMESSAGE(5) END; 42524100
A=PRT[P1MIX,4]; 42524200
RCW = M[RCWL + PRT[P1MIX,8] INX NOT ((SYLLABLE=@235)+2)];% 42525000
IF RCW,[33:1] THEN % TYPE 13 INTRNSC 42525100
BEGIN 42525110
I:=0; 42525115
Y+[I],[CF]; 42525120
I +FLAG((@2520000000000000)&(RCW,[34:14])[CTC]); 42525130
MAKEPRESENT(Y); 42525140
M[RCWL]=FLAG(RCW&(M[RCW,[FF]]INX (NFLAG( I )),[CF])[CTC]); 42525150
GO TO INITIATE; 42525160
END ELSE 42525170
IF NOT PRT[P1MIX,A[RCW],[8:10]],[2:1] THEN% 42525500
MAKEPRESENT(PRTROW[P1MIX] INX A[RCW],[8:10]);% 42526000
M[RCWL]=FLAG(RCW&(M[RCW,[18:15]] INX A[RCW],[18:15])[33:33:15]); 42527000
GO TO INITIATE; 42528000
% 42529000
INVALIDINDEX; 42530000

```

FOR I←6 STEP 5 UNTIL 11 DO	42531000
IF TOP.[18:15]=(MOM+[PRT[P1MIX,I]],.[33:15]) THEN	42532000
IF (SIZE+M[MOM],[8:10])<1023 THEN	42533000
BEGIN IF M[MOM],[2:1]=0 THEN MAKEPRESENT(MOM);	42534000
M[(ALOC+M[MOM],[33:15])=2],[2:1]+1;	42535000
IF M[ALOC-1],[FF]≠0 THEN ARTN(M[MOM], -1);	42535500
M[MOM]+FLAG(0&MOM[18:33:15]	42536000
&(IF SIZE<512 THEN 2×SIZE ELSE 1023)[8:38:10];	42537000
IF TYPE ← P(FLAGTESTER, TOP, XCH, DEL) THEN MAKEPRESENT(MOM)	42537050
ELSE	42537060
MAKEPRESENT(ANALYSIS);	42538000
MOVE(SIZE, ALOC, M[MOM]);	42539000
FORGETSPACE(ALOC);	42539050
IF TYPE THEN GO XYT;	42539060
GO TO INITIATE;	42539100
END;	42539200
ERRORFIXER(4); TERMINATE(P1MIX); TERMINALMESSAGE(7);	42540000
%	42541000
EXPUNDERFLOW;	42542000
IF DOUBLEPRECISION THEN M[S=3]+0;	42546000
M[S=2]+0;	42547000
IF JAR[P1MIX,2],[3:1] AND(PRT[P1MIX,@51] AND @20)≠0 THEN	42547100
PRT[P1MIX,@51]+P(DUP,L0D) OR 6;	42547200
GO TO INITIATE;	42548000
%	42549000
DIVIDEBYZERO;	42550000
IF (P(JAR[P1MIX,2],DUP)≥0 AND NOT(P(XCH).[3:1] AND	42550500
PRT[P1MIX,@51],[44:1])) THEN	42550600
BEGIN ERRORFIXER(8); TERMINATE(P1MIX); TERMINALMESSAGE(13) END	42551000
ELSE IF JAR[P1MIX,2] < 0 THEN IF PRT[P1MIX,11],[FF] = 0 THEN	42551090
PRT[P1MIX,11]+1 ELSE PRT[P1MIX,PRT[P1MIX,11],[FF]]+1	42551100
ELSE	42551110
BEGIN PRT[P1MIX,@51]+P(DUP,L0D) OR 1;	42551200
IF DOUBLEPRECISION THEN M[S=3]+0;	42551300
M[S=2]+0;	42551400
END;	42551500
GO TO INITIATE;	42552000
XYT;	42553000
END INTERRUPT;	42554000
\$ SET OMIT = NOT(STATISTICS)	42599999
PROCEDURE FILLSYSTAT;	42600000
BEGIN	42600100
REAL RCW=+0,X1,X2,X3,X4;	42600200
X2:=DIRECTORYSEARCH("SYSTEM "	42600300
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42600309
&(SYSNO+17)[42:42:6]	42600310
\$ POP OMIT	42600311
,"STATS ",4);	42600320
IF COUNTARRAY[28] NEQ DATE THEN	42600700
BEGIN	42600800
DISKWAIT(=(X1:=SPACE(30)),-30,0);	42600850
COUNTARRAY[28]:=DATE;	42600900
M[X1+5+SYSNO]:=0;	42600910
DISKWAIT(X1,-30,0);	42600920
FORGETSPACE(X1);	42600930
END ELSE	42601000
BEGIN	42601100
DISKWAIT(=COUNTARRAY,[CF],60,SYSTATBASE);	42601140
COUNTARRAY[29]:=XCLOCK;	42601200
X4:=(M[X2 INX 10]+(M[X2 INX 7]×2));	42601300

DISKWAIT(COUNTARRAY,[CF],61,X4);	42601600
M[X2 INX 7]:=P(DUP)+1;	42601700
COUNTARRAY[29]:=XCLOCK+INTERVAL;	42601750
END;	42601800
DISKWAIT(X2,[CF],30,X2,[FF]);	42601810
IF M[X2 INX 7]=99 THEN SAVESTATISTICS;	42601860
FOR X3:=0 STEP 1 UNTIL 27 DO COUNTARRAY[X3]:=0;	42601900
FOR X3:=30 STEP 1 UNTIL 59 DO COUNTARRAY[X3]:=0;	42602000
COUNTARRAY[47]:=XCLOCK;	42602050
FORGETSPACE(X2);	42602100
FORGETSPACE(DIRECTORYSEARCH("SYSTEM "	42602110
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42602119
&(SYSNO+17)[42:42:6]	42602120
\$ POP OMIT	42602121
,"STATS ",14));	42602130
KILL([RCW] INX NOT 2);	42602200
END OF FILLSYSTAT;	42602300
PROCEDURE SAVESTATISTICS;	42700000
BEGIN	42701000
REAL RCW:=0,X,X1,X2,X3;	42702000
REAL NAMEIT;	42703000
LABEL XOUT,GOTNAME;	42704000
XI=DIRECTORYSEARCH("SYSTEM "	42710000
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42710099
&(SYSNO+17)[42:42:6]	42710100
\$ POP OMIT	42710101
,"STATS ",4);	42710200
MOVE(30,X,[CF],X3:=SPACE(30));	42710300
DISKWAIT(-(X2:=SPACE(30)),-30,0);	42711000
IF COUNTARRAY[28]#DATE THEN M[X2+5+SYSNO],[CF]:=X1:=0 ELSE	42711500
X1:=M[X2+5+SYSNO],[CF];	42712000
WHILE (X1:=X1+1) < 100 DO	42712500
BEGIN	42713000
STREAM(A:=[NAMEIT],B:=[DATE],C:=X1);	42713100
BEGIN	42713200
SII=LOC C; DII=A; DII:=DII+1; DS:=2 DEC;	42713300
DS:=2LIT"ON"; SII:=B; SII:=SII+5; DS:=3 CHR;	42713400
END;	42713450
IF DIRECTORYSEARCH(-NAMEIT,"SYSTEM "	42713500
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42713549
&(SYSNO+17)[42:42:6]	42713550
\$ POP OMIT	42713551
,5)=0 THEN GO GOTNAME;	42713600
END;	42713650
STREAM(X1:=X1:=SPACE(10));	42713700
BEGIN	42713750
DS:=37LIT"# STATISTICS SYSTEM FILE NOT CREATED ";	42713800
DS:=24LIT"- MAX FILE NO. EXCEEDED*";	42713850
END;	42713900
M[X2+5+SYSNO],[CF]:=0; DISKWAIT(X2,-30,0);	42713950
SPOUT(X1); GO XOUT;	42714000
GOTNAME:	42714050
IF (M[X3+10]=GETUSERDISK(-200 OR M))=0 THEN	42714100
BEGIN	42716000
STREAM(A:=[NAMEIT],B:=(SYSNO+17),C:=X1:=SPACE(5));	42717000
BEGIN	42718000
SII=A; DS:=17LIT"NO USER DISK FOR ";	42719000
SII:=SII+1; DS:=7 CHR; DS:=7 LIT"/SYSTEM";	42720000
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42720099
SII=LOC B; SII:=SII+7; DS:= CHR;	42720100

\$ POP OMIT	42720101
DS:=LIT" ";	42720200
END;	42721000
SPOUT(X1);	42721500
GO TO XOUT;	42722000
END;	42723000
SYSTATBASE:=M[X3+10];	42723100
M[X2+5+SYSNO],[CF]:=X1;	42723500
STREAM(A:=[DATE],B:=X INX 3,C:=0);	42724000
BEGIN	42725000
SI:=A; DI:=LOC C; DS:=8 OCT; SI:=LOC C; SI:=SI+5;	42726000
DI:=B; DI:=DI+5; DS:=3 CHR;	42727000
END;	42728000
COUNTARRAY[29]:=XCLOCK;	42728050
DISKWAIT(COUNTARRAY,[CF],61,(M[X INX 10]+(M[X INX 7]x2)));	42728100
FOR X1:=0 STEP 1 UNTIL 59 DO COUNTARRAY[X1]:=0;	42728200
COUNTARRAY[28]:=DATE; COUNTARRAY[29]:=XCLOCK+INTERVAL;	42728300
COUNTARRAY[47]:=XCLOCK;	42728400
M[X INX 10]:=*P(DUP)+2;	42728600
M[X INX 8]:=*P(DUP)-2;	42728700
M[X3+7]:=1;	42730000
DISKWAIT(X2,-30,0);	42731000
DISKWAIT(X3,30,X,[FF]);	42731100
ENTERUSERFILE(-NAMEIT,"SYSTEM "	42732000
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42732099
&(SYSNO+17)[42:42:6]	42732100
\$ POP OMIT	42732101
,X INX 0=1);	42732200
STREAM(A:=[NAMEIT],B:=(SYSNO+17),C:=X1:=SPACE(6));	42733100
BEGIN	42733150
DS:=21LIT"NEW STATISTICS FILE: ";	42733200
SI:=A; SI:=SI+1; DS:=7 CHR; DS:=7 LIT"/SYSTEM";	42733300
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42733349
SI:=LOC B; SI:=SI+7; DS:= CHR;	42733350
\$ POP OMIT	42733351
DS:= 9 LIT " CREATED+";	42733360
END;	42733400
SPOUT(X1);	42733500
FORGETSPACE(DIRECTORYSEARCH(NAMEIT,"SYSTEM "	42733510
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42733519
&(SYSNO+17)[42:42:6]	42733520
\$ POP OMIT	42733521
,14));	42733530
XOUT;	42733550
FORGETSPACE(X);	42733600
FORGETSPACE(X2);	42736000
FORGETSPACE(X3);	42737000
FORGETSPACE(DIRECTORYSEARCH("SYSTEM "	42739000
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	42739099
&(SYSNO+17)[42:42:6]	42739100
\$ POP OMIT	42739101
,"STATS ",14));	42739200
END;	42740000
\$ POP OMIT	42740001
% THE FORMAT OF DIRECTORY TOP%	44000000
%    D[0]=OPTION WORD%	44001000
%    D[1]=DATE%	44002000
%    D[2]=NUMBER OF ELECTRONIC UNITS%	44003000
%    D[3]=HIGHEST ADDRESS OF BACKUP STORAGE%	44004000
%    D[4]=HIGHEST ADDRESS OF DIRECTORY%	44005000

%	D[5]=LAST NUMBER USED FOR CONTROL DECK%	44006000
%	D[6]=FIRST CONTROL DECK QUEUED (LOCATION IN DIRECTORY)%	44007000
%	D[7]=LAST CONTROL DECK QUEUED (LOCATION IN DIRECTORY)%	44008000
%	D[8]=NEXT NUMBER AVAILABLE FOR PRINTER BACKUP DISK	%P 44008100
%	D[9]=CORE, CONTAINS MULTIPROCESSING FACTOR	44008200
%	D[10] THRU D[15] SPECIFY WHICH DC-STATIONS ARE SPO-LIKE.	44008300
%	D[16]=QUEUE VALUES FOR SPO STATIONS(BATCH MCP)	44008310
%	D[17] SPECIFIES SPO UNITS FOR BATCH MCP	44008320
%	D[18]=TIME OF DAY	44008330
%	D[19]=LOCATION OF FENCE	44008340
%	D[20],[8:10]=NUMBER OF LAST LOG FILE	44008350
%	.[18:30]=NUMBER OF ENTRIES IN LOG (UPDATED BY NSECOND)	44008360
%	D[21]=SCHEDWRD	44008370
%	D[22],[38:10]=NUMBER OF CURRENT MAINTENANCE LOG,	44008380
%	.[28:10]=NUMBER OF CURRENT REMOTE LOG (DC MCP).	44008390
%	D[23] THRU D[26] SPECIFY WHICH SU-S WERE READY AT THE LAST H/L.	44008400
%	D[27] IS RESERVED FOR USE BY THE LOCAL SITE.	44008410
%	D[28]=DISK ADDRESS OF DIRECTORYTOP	44008499
	\$ SET OMIT = NOT STATISTICS	44008997
	PROCEDURE INTFINISH; FORWARD;	44008998
	\$ POP OMIT	44008999
	SAVE PROCEDURE INITIALIZE;%	44009000
	BEGIN REAL	44010000
	I = +1,	44010010
	A = I+1,	44010020
	T = A+1,	44010030
	B = T+1,	44010040
	C = B+1,	44010050
	J = C+1,	44010060
	W = J+1,	44010070
	LASTL = W+1,	44010080
	LDATE = LASTL+1,	44010090
	MEND = LDATE+1,	44010100
	INTS = MEND+1,	44010110
	INTSS = INTS+1,	44010120
	MEMASK = INTSS+1;	44010120
REAL	T1 = LASTL,	44010200
	SHLM = MEND,	44010210
	MSTART = INTS,	44010220
	Y = C,	44010230
	Z = T1;	44010240
INTEGER	XCLOCK = XCLOCK;	44010400
ARRAY	DDD = MEMASK+1[*],	44010500
	X = W[*];	44010510
DEFINE	NUMSTACK = 2#;	44011000
\$ SET OMIT = NOT SHAREDISK		44024990
REAL	HOLDER = I,	44025000
	BYPASS = J;	44025100
\$ POP OMIT		44025110
\$ SET OMIT = NOT(AUXMEM)		44025499
LABEL	AUXAGN,AUXMESS;	44025500
\$ POP OMIT		44025501
	LABEL TRYNEXTMOD,RESTARTCYCLE,NULLINT;	44026000
SUBROUTINE XXXXXX; BEGIN A+X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X; END;%		44027000
	DEFINE SETUPINITIALBUFFERS =	44040000
	J:=220 + 20*(BIGUNMIN + LMAX DIV 2);	44040500
	I:=(T:=GETSPACE(J+2,5,0)+1)+3 AND NOT 3;	%R0744041000
	MOVE(J+2,T,T+1); T:=I+J-1;	%R0744042000
	FOR I+1 STEP 20 UNTIL T DO FORGETAREA(2,I);	44043000
	AREARDY + TRUE#;	44044000



```

SUBROUTINE FIXEX;                                44075000
  BEGIN DISKWAIT(=(C:=C-(J:=M[I],[8:10])), J, M[I],[FF]+MCPBASE); 44076000
    M[I]:=(P(DUP))&C&CTC];                        44077000
  END;                                           44078000
SUBROUTINE FIX;%                                44079000
  BEGIN M[T]:=[M[J]]&I[8:38:10];                44080000
    J = J+I];%                                   44081000
  END;%                                           44082000
SUBROUTINE FIXFENCE;                            44082100
  BEGIN I:=T DIV 4096;                          % POINT IT TO AN ON-LINE MOD 44082200
    WHILE (TWO(7-I) AND MEMASK)≠0 AND I LSS 8 DO 44082300
TRYNEXTMOD:  T:=(I:=I+1)×4096;                  44082400
    IF T=4096×I ≠ 0 THEN                          44082500
    IF (TWO(8-I) AND MEMASK)=0 THEN ELSE GO TRYNEXTMOD; 44082600
% THIS INSURES THERE IS ROOM FOR TABLES IMMEDIATELY BELOW 44082700
% THE FENCE, (KILL ASSUMES IT HAS A STACKQ STACK IF S>PRT.) 44082800
  END;                                           44082900
MCPBASE:=M[0],[18:30];                          44083000
DIRECTORYTOP:=M[1];                              44083100
$ SET OMIT = NOT(SHAREDISK)                      44083200
SYSMAX:=M[0],[14:2];                             44083300
SYSNO:=M[0],[16:2];                              44083400
$ POP OMIT                                       44083401
$ SET OMIT = NOT(DEBUGGING)                     44083500
PAUSEVALUE:=0;                                  44083600
$ POP OMIT                                       44083601
PEUIO:=EUIO:=M[@133]&20[8:38:10] ;              44085100
  IOQUESLOTS:=32;                                44085500
  IOQUEAVAIL:=31;                                44085600
RESTARTCYCLE:= COMMENT RETURN TO HERE TO CHANGE FENCE POSITION; 44087000
  P(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);          44087010
  RRRMECH = @1400000000;%                         44088000
  LMAX:=STAMAX:=MAXLMAX;                          44090000
  WITCHINGHOUR:=5184000;                          44090500
  WORDOFEASE:=@2525252525252525;                 44091000
  NOPROCESSTOG = -1;%                             44092000
$ SET OMIT = NOT STATISTICS                     44092490
  LEFTHALF:=16777216;                             44092500
$ POP OMIT                                       44092510
  STREAM(S+18,D+100);%                             44093000
  BEGIN%                                           44094000
    SI = S; DS = 11 WDS; D = DI;                  44095000
    DI = S; 11(DS + 8 LIT "102(0000" );           44096000
    19(SI + SI+8); S = SI;                        44097000
    DI=D; DS=2 WDS;%                               44098000
    DI=S; DS=16 LIT"042(0000"%                   44099000
  END;%                                           44100000
MSTART:=P(.,INITIALIZE,LOD),[CF];                44101000
HOLDER:=DIRECTORYTOP-7-(HOLDMAX+29) DIV 30; % SEE ALSO 40200100 44101100
USERDISKBOTTOM:=HOLDER-DISKAVAILTABLEMAX;        44101200
IF (I:=(USERDISKBOTTOM=50) DIV SYSMAX) > 247 THEN I:=247; 44101300
ESPDISKBOTTOM:=50+(SYSNO×I);                     44101400
ESPDISKTOP:=ESPDISKBOTTOM+I-7;                  44101500
FOR I:=0 STEP 1 UNTIL 7 DO                        44102000
  BEGIN M[4096×I]:=1; DO UNTIL (T:=COREND)≠16;   44102500
    MEMASK:=2×MEMASK+T;                           44103000
  END;                                           44103500
IF FENCE,[CF]≠FENCE OR FENCE LSS 8192 THEN FENCE:=16384; 44104000
T:=FENCE;                                         44105000
FIXFENCE;                                        44105500

```

```

FENCE:=T;
CHUNKMAX:=(@100000-FENCE) DIV CHUNKZIZE;
J:=(A:=FENCE:=@100000-CHUNKMAX*CHUNKZIZE) DIV 4096 -1;
MEMROW[0]:=[M[0]]&4[8:38:10];
MOVE(MIXMAX, MEMROW, [MEMROW[1]]);
WHILE (TWO(7-J) AND MEMASK)≠0 DO
    BEGIN J=J-1; A+A=4096 END;
M[M[AVAIL]+(MEND+A-3)+1]+MEND INX @77777000Q1;
M[MEND+2]+MEND+1;
M[MEND] + 0&1[2:47:1]
                &MSTART[CTF]]%;
M[0]+MSTART&MEND[CTF]&@100001[2:32:16];
M[MSTART]+LASTL+MEND];%
M[LEFTLIT]=0;
J=J+1;
FOR I+1 STEP 1 UNTIL J DO
IF (TWO(7-I) AND MEMASK)≠0 THEN
    BEGIN C+4096*I;
        M[MSTART],[CF]+B+C-1;
        DO BEGIN I+I+1; C+C+4096 END
        UNTIL (TWO(7-I) AND MEMASK)=0;
        M[B]+C&MSTART[CTF]&1[2:47:1];
        M[C]+MEND&B[CTF];
        M[MEND],[FF]+C;
        FORGETSPACE(MSTART+2);
        MSTART+C;
    END;
    FORGETSPACE(MSTART+2);

$ SET OMIT = NOT(DFX)
DISKOUNT=P(RRR),[29:1]+P(RRR),[28:1];
EUW=@7777777777;
$ POP OMIT
    STREAM(S+100,D+18)];%
    BEGIN%
    SI + S; DS + 11 WDS;
    19(DI + DI+8); DS + 2 WDS;
    END;%
    M[16]:=@0010413100000000; %DO UNTIL FALSE
    TAR:=[M[MEND-130]]&2[8:38:10];
    SPACESTACK+MEND-128;
    INTS:=GETSPACE(P(.,COREND,LOD),[CF] -
                P(.,INITIALIZE,LOD),[CF],1,1)+2;
    INTSS:=GETSPACE(200,12,1)+2;
    WHILE FALSE DO;% FIX C RELATIVE CONSTANT ERRORS
$ SET OMIT = NOT(AUXMEM)
    A := P(RRR),[31:1]; B := P(RRR),[30:1];
$ POP OMIT
    I + (MIXMAX+1)*6          % PRT, JAR, DAT, UV, REPLY, TAR
    + CHUNKMAX                % CT
    + UVSIZE                   % UVROW[0]
    + PUNTSIZE
    + SHEETMAX+1
    + MESSAGETABLESIZE
    + SPACESTACKSIZE
    + NUMSTACK*STANDARDSTACK
    + (W:=(ESPDISKTOP=ESPDISKBOTTOM+47) DIV 48)
    + 580                      % 3 SPACER
                                % 4 CIDROW
                                % 5 CHANNEL, CHANIO

```

```

44106000
44107000
44108000
44109000
44109500
44110000
44110100
44110200
44111000
44112000
44113000
44114000
44115000
44116000
44117000
44118000
44119000
44120000
44121000
44122000
44123000
44124000
44125000
44125500
44126000
44127000
44128000
44141000
44142000
44142999
%DFX44143000
%DFX44143100
44143101
44144000
44145000
44146000
44147000
44148000
44148100
44148200
44148500
44149000
44149100
44149200
44150000
44150009
44150010
44150011
44150500
44150600
44150700
44150750
44150800
44150900
44151000
44151100
44151200
44151300
44151310
44151320

```

	% 13	UNITCODE	44151340
	% 16	PRNTABLE	44151350
	% 20	MAINTBUFFER	44151360
	% 32	SO, IOQUE, LOCATQUE,	44151370
	%	FINALQUE, WAITQUE,	44151380
	%	TRANSACTION	44151390
	% 36	LABELTABLE, MULTITABLE,	44151400
	%	RDCTABLE	44151410
	% 38	LOGARRAY	44151420
	% 48	ISTACK	44151430
	% 128	WORKERSTACK	44151440
\$ SET OMIT = NOT AUXMEM			44151495
+ 10 + 74*(A+B)	% CTABLE		44151500
+ P(MIXMAX+1,DUP,+)	% AUXDATA, AUXCODE		44151520
\$ POP OMIT			44151585
\$ SET OMIT = NOT PACKETS			44151595
+ 4	% PSEUDO		44151600
+ P(MIXMAX+1,DUP,+)	% PSEUDOMIX, NYLONZIPPER		44151620
\$ POP OMIT			44151685
\$ SET OMIT = NOT SHAREDISK			44151695
+ LQMAX			44151700
\$ POP OMIT			44151785
\$ SET OMIT = NOT STATISTICS			44151795
+ 94	% COUNTARRAY, DISK WAIT TIME		44151800
+ MIXMAX+1	% SWAP DELAY		44151820
\$ POP OMIT			44151885
\$ SET OMIT = NOT (SAVERESULTS OR DEBUGGING)			44151895
+ RESULTMAX	% RESULTHOLDER		44151900
\$ POP OMIT			44151985
\$ SET OMIT = NOT DEBUGGING			44151995
+ 122	% DBARRAY, WB, RB, TBL, STOPS		44152000
\$ POP OMIT			44152085
+ (T +	% FIXEX PROCEDURES		44152500
P(,OLAY,LOD),[8:10]			44152520
+ P(,SHORTCOMMUNICATES,LOD),[8:10]			44152540
+ P(,DCIOFINISH,LOD),[8:10]			44152560
+ P(,NEXTDCIO,LOD),[8:10]			44152580
+ P(,DCWRITE,LOD),[8:10]			44152600
+ P(,ENTERLINEQ,LOD),[8:10]			44152620
\$ SET OMIT = NOT AUXMEM			44152790
+ P(,AUXILIARYSPACE,LOD),[8:10]			44152800
+ P(,FORGETAUXILIARYSPACE,LOD),[8:10]			44152820
+ P(,FILLORKILL,LOD),[8:10]			44152840
\$ POP OMIT			44152850
\$ SET OMIT = NOT DEBUGGING			44152890
+ P(,DT,LOD),[8:10]			44152900
+ P(,EXP,LOD),[8:10]			44152920
\$ POP OMIT			44152930
)			44153000
M[C:=J:=GETSPACE(1,0,0)+T+2]I=0;			44153500
MOVE(I-T-1,J,J+1);			44154000
I + MIXMAX+1;%			44155000
T + P(,PRT); FIX;%			44156000
\$ SET OMIT = NOT(AUXMEM)			44156199
T + P(,AUXDATA); FIX;			44156200
T + P(,AUXCODE); FIX;			44156300
\$ POP OMIT			44156301
T + P(,JAR); FIX;%			44157000
T + P(,DAT); FIX;			44157100
T + P(,UV); FIX;			44160000

```

T:=P(,TAR)); FIX;
$ SET OMIT = NOT(PACKETS)
T:=P(,PSEUDOMIX)); FIX;
T:=P(,NYLONZIPPER)); FIX;
$ POP OMIT
T + P(,REPLY)); FIX;%
$ SET OMIT = NOT(STATISTICS)
T:=P(,SWAPDELAY)); FIX;
$ POP OMIT
I + CHUNKMAX;
T + P(,CT)); FIX;
$ SET OMIT = NOT(PACKETS)
I:=4; T:=P(,PSEUDO)); FIX;
$ POP OMIT
I + 5; T + P(,CHANNEL)); FIX;
T:=P(,CHANIO)); FIX;
$ SET OMIT = NOT(AUXMEM OR MONITOR)
I:=10; T:=P(,CTABLE)); FIX;
$ SET OMIT = NOT(AUXMEM)
IF A THEN BEGIN CTABLE[0],[FF] + J; J + J+74 END;
IF B THEN BEGIN CTABLE[1],[FF] + J; J + J+74 END;
$ RESET OMIT
I + 4; T + P(,CIDROW)); FIX;
$ SET OMIT = NOT(SAVERESULTS OR DEBUGGING)
I + RESULTMAX;
T := P(,RESULTHOLDER)); FIX;
$ POP OMIT
I + 3; T+P(,SPACER)); FIX;
$ SET OMIT = NOT(DEBUGGING)
I + 31; T + P(,DBARRAY)); FIX;
I + 10; T + P(,WB)); FIX;
I + 11; T + P(,RBX)); FIX;
I + 20; T + P(,STOPS)); FIX;
I + 50; T + P(,TBL)); FIX;
$ POP OMIT
I + SHEETMAX+1; T+P(,SHEET)); FIX;
I + MESSAGETABLESIZE; T + P(,MESSAGETABLE)); FIX;
J + J+1; I + 37; T + P(,LOGARRAY)); FIX;
$ SET OMIT = NOT(STATISTICS)
J:=J+1; I:=61; T:=P(,COUNTARRAY)); FIX;
$ POP OMIT
I + 16; T + P(,PRNTABLE)); FIX;
I + 32;%
T + P(,SQ)); FIX;
T + P(,FINALQUE)); FIX;%
T + P(,LOCATQUE)); FIX;%
T + P(,IOQUE)); FIX;%
$ SET OMIT = NOT(STATISTICS)
T:=P(,DISKWAITIME)); FIX;
$ POP OMIT
T + P(,TRANSACTION)); FIX;%
T + P(,WAITQUE)); FIX;%
SPACESTACK:=J; J:=J+SPACESTACKSIZE;
$ SET OMIT = NOT(SHAREDISK)
I:=LQMAX;
T:= P(,LQUE)); FIX;
$ POP OMIT
ESPTAB+J; J+J+W;
ESPCount:=ESPDISKTOP-ESPDISKBOTTOM;
I + 36;

```

```

44160050
44160399
44160400
44160500
44160501
44162000
44162899
44162900
44162901
44163100
44163200
44163299
44163300
44163301
44164000
XR59 44164040
44164059
44164060
44164065
44164070
44164080
44164081
44164100
44164118
44164119
44164120
44164121
44164200
44164999
44165000
44165100
44165200
44165300
44165400
44165401
44165500
44165700
44166000
44166099
44166100
44166101
44167000
44168000
44168100
44169000
44170000
44171000
44171899
44171900
44171901
44172000
44173000
44173100
44173139
44173140
44173150
44173151
44173200
44173300
44174000

```

```

T ← P(,LABELTABLE); FIX;% 44175000
T ← P(,MULTITABLE); FIX;% 44176000
T ← P(,RDCTABLE); FIX;% 44177000
I ← UVSIZE; T←UV.[CF]; FIX; 44178000
I:=13; T:=P(,UNITCODE); FIX; 44178100
I:=48; T:=P(,ISTACK); FIX; 44179000
I:=20; T:=P(,MAINTBUFFER); FIX; 44179050
WORKERSTACK:=J; J:=J+128; 44179100
I←J; 44179200
FOR T:=2 STEP 1 UNTIL NUMSTACK DO 44179300
BEGIN M[J←J+STANDARDSTACK]←I; I←J END; 44179400
J←(STACKQ+J)+STANDARDSTACK; 44179500
I:=PUNTSIZE; T:=P(,PUNTER); FIX; 44179900
STACKUSE ← TRUE;% 44180000
$ SET OMIT = NOT(DEBUGGING) 44180099
NSYMBS ← 34; 44180100
TBL[32] ← "SP0"; TBL[33] ← [TYPETOG],[33:15]; 44180120
TYPETOG ← "0"; 44180140
TBL[34] ← "F"; 44180200
$ POP OMIT 44180201
STREAM(PUNTER); 44180300
BEGIN DI:=DI+24; DS:=8 LIT "DATACOM "; 44180400
DS:=16 LIT "INVALID LINK←"; 44180500
DS:=16 LIT "INVALID ADDRESS←"; 44180600
$ SET OMIT = NOT(SHAREDISK) 44180699
DS:=16 LIT"LOCK QUE OVFLOW←"; 44180700
$ SET OMIT = NOT AUTODUMP 44180750
DS:=32 LIT"10100)0)4A0DKI002900SI000×+A144A"; 44180800
DS:=32 LIT"1DM908/11×007Y0(1×00P×1≤0SK)0QKI"; 44180900
DS:=25 LIT"0WK)0HKI0,K)08KI0JK)1C←R1"; DS:=LIT""; 44181000
DS:=30 LIT"KI00002900SI0)000000512900SI00"; 44181100
DS:=28 LIT"806#8A#04A1,1D4A#31#4A0)0Y/I"; 44181200
$ POP OMIT OMIT 44181250
END; 44181300
HALTSET:=1; 44181400
FOR I:=0 STEP 1 UNTIL 35 DO 44181500
BEGIN LABELTABLE[I]:=0; 44181750
IF I<32 THEN 44182000
BEGIN IOQUE[I]:=I-1; 44182500
TINU[I],[18:12]:=0; 44183000
IF I LEQ 12 THEN UNITCODE[I]:=0; 44183500
END END; 44184000
UNITCODE[7]:=0; 44184500
LABELTABLE[25]:=0; 44185000
FORKQUE ← M OR P(,FORKQUE)&P(,FORKQUE)[CTFJ&@777[9:39:9]; 44186500
BED1 ← FLAG(BED+[BED] INX @10077777770000); 44187000
CLICK ← @777777777777; 44187100
PRIORITY←PRYOR[0]←-1; 44187200
NUMESS := NUMAINTMESS := -100; 44188000
CLOCK:=SPOWORD:=0; 44188100
LOGHOLDER:=LOGENTRY:=MDELTA:=MLOG:=MAINTLOGARRAY:=NXDISK:=0; 44188200
MROW := 100; 44188300
KEYBOARDCOUNTER:=1; % KEEPS KEYIN FROM RUNNING. 44188500
$ SET OMIT = NOT(DEBUGGING) 44188999
P(,DT,LOD,0,DIB 5,TRB 1,DT,←);% 44189000
$ POP OMIT 44189001
M[WORKERSTACK]:=WORDOFFEASE; %INITIALIZE MCP STACKS 44189500
MOVE(126,I:=WORKERSTACK,[CF],I+1); %TO SPOT POSSIBLE OVER 44189600
MOVE(48,I,ISTACK); %FLOWS 44189700
MOVE(SPACESTACKSIZE,I,SPACESTACK); 44189750

```

```

MOVE(@60,I,@100); 44189800
NT1:=0; MOVE(14,@160,@161); % NT1 = @160 44189900
$ SET OMIT = NOT SHAREDISK 44189995
MOVE(LQMAX=1,I,LQUE,[CF]); 44190000
$ POP OMIT 44190005
FOR I:=0 STEP 1 UNTIL 2 DO 44190100
    SPACER[I]:=P(DUP),[CF]&P(DUP)[CTF]; 44190200
SETUPINITIALBUFFERS; 44190300
EVENT[0],[FF]:=[EVENT[0]]; 44190400
LOGARRAY[31]+IOMASK+@2000000000; 44191000
DISKWAIT(-A:=SPACE(30)),-30,DIRECTORYTOP=SYSNO); 44191005
T:=M[A+19],[CF]; 44191010
FIXFENCE; 44191015
IF T#FENCE AND T#0 THEN 44191020
BEGIN IF (T := @100000-(((P(DUP)-T)DIV CHUNKZIZE)
    XCHUNKZIZE)) 44191025
    Z@20000 AND T#@70000 THEN FENCE := T; 44191030
    M[A+19]:=FENCE; 44191040
    DISKWAIT(A,-30,DIRECTORYTOP=SYSNO); 44191050
    TOGGLE:=0; 44191060
    P(0,0,RDF,FCX,STS); GO TO RESTARTCYCLE; 44191090
END FENCE MOVING JAZZ; 44191100
FORGETSPACE(A); 44191110
PROCTIME[0]+IOTIME[0]+@2003777777777777; 44192000
I:=P(,OLAY); FIXEX; 44192005
$ SET OMIT = NOT(DEBUGGING) 44192009
I+P(.DT);FIXEX;I+P(.EXP);FIXEX; 44192010
$ POP OMIT 44192011
$ SET OMIT = NOT(AUXMEM) 44192059
I:=P(,AUXILIARYSPACE); FIXEX; 44192060
I:=P(,FORGETAUXILIARYSPACE); FIXEX; 44192062
I:=P(,FILLORKILL); FIXEX; 44192064
$ POP OMIT 44192065
I+P(,SHORTCOMMUNICATES,NT1,DEL);FIXEX; 44193020
I:=P(,DCIOFINISH); FIXEX; 44193030
I+P(,NEXTDCIO);FIXEX; 44193032
INTERROGATEMASK:=@0400004000000000; 44193033
I+P(,DCWRITE);FIXEX; 44193034
I+P(,ENTERLINEQ); FIXEX; 44193035
LINETABLE:=GETSPACE(C:=6*LMAX+6,0,0)+2; % RIGHT IN FRONT 44193500
MOVE(C,LINETABLE=1,LINETABLE); % OF FIXEX PRCDRS, 44193600
$ SET OMIT = NOT SHAREDISK 44193890
UNLOCK(DIRECTORYTOP=SYSNO); 44193900
$ POP OMIT 44193910
$ SET OMIT = NOT(DEBUGGING) 44196999
FOR I + 0 STEP 2 UNTIL 30 DO BEGIN% 44197000
    TBL[I] + TINU[I DIV 2] ,[30:18];% 44198000
    TBL[I+1] + [LABELTABLE[I DIV 2]], [33:15] END;% 44199000
$ POP OMIT 44199001
% FIND INITIAL VALUE FOR CORE 44201200
CORE:=P(,COREEND,LOD),[CF]=P(,INITIALIZE,LOD),[CF]; 44201300
I+M[M[M[AVAIL]]]; 44201400
WHILE I,[FF] # @77777 DO 44201500
    BEGIN CORE + CORE + I.[FF]; I + M[I] END; 44201600
CORE + CORE DIV 64; 44202000
INDIAN,[FF]+[INDIAN]; 44202095
DDD:=M[A:=SPACE(483)]&483[8:38:10]; 44202500
DISKWAIT(-A,-30,0); 44202600
DISKWAIT(-31=A,-30,MCPNAMESEG); 44202700
MOVE(2,A+10+5*SYSNO,A+51+2*SYSNO); 44202800

```

	DISKWAIT(A+31,-30,MCPNAMESEG);	44202900
	STREAM(ML:=MARKLEVEL,PL:=M[3]:=PATCHLEVEL,LL:=LOCALEVEL	44203000
	,FENCE,MEMASK,N:=A+10+5*SYSNO	44203150
\$ SET OMIT =	NOT(SHAREDISK)	44203179
	,SYS=SYSNO+17	44203180
\$ POP OMIT		44203181
	,T:=B:=SPACE(15));	44203200
	BEGIN DS=5 LIT "=H/L ";	44204000
\$ SET OMIT =	NOT(SHAREDISK)	44204099
	SI=LOC SYS;SI=SI+7;DS=7 LIT"SYSTEM ";DS=CHR;	44204100
\$ POP OMIT		44204101
	DS=6 LIT" WITH ";SI=N;SI=SI+1;DS=7 CHR;	44204200
	DS:=LIT"/";SI:=SI+1;DS:=7 CHR;DS:=6 LIT" MARK ";	44204500
	SI:=LOC ML; IF SC GEQ " " THEN;	44204600
	8(IF TOGGLE THEN IF SC="0" THEN SI:=SI+1 ELSE DS:=CHR	44204650
	ELSE DS:=CHR); DS:=LIT",";	44204700
	SI:=LOC PL; IF SC GEQ " " THEN;	44204750
	6(IF TOGGLE THEN IF SC="0" THEN SI:=SI+1 ELSE DS:=CHR	44204800
	ELSE DS:=CHR); DS:=2CHR;	44204850
	SI:=LOC LL; IF SC GEQ " " THEN;	44205000
	8(IF TOGGLE THEN IF SC="0" THEN SI:=SI+1 ELSE DS:=CHR	44205050
	ELSE DS:=CHR);	44205100
	DS= 3 LIT ",F="; SI= LOC FENCE; DS= 5 DEC;	44205140
	DS= 7 LIT " [MODS=";	44205160
	SI=LOC MEMASK; SKIP 40 SB;	44206000
	8(IF SB THEN DS=LIT"@ " ELSE DS=LIT"R"; SKIP SB);	44206100
	DS= 4 LIT "]" =+";	44206200
	END;	44206300
	SHLM:=GETSPACE(15,9,5)+2;	44206400
	MOVE(15,B,SHLM);	44206500
\$ SET OMIT =	NOT (DUMP OR DEBUGGING)	44206899
	MEMOD=MEMASK;	%A  44206900
\$ POP OMIT		44206901
\$ SET OMIT =	NOT(SHAREDISK)	44207899
	FOR I:=13 STEP 5 UNTIL 28 DO	44207900
	LDATE:=LDATE OR (DDD[I]#0 AND ((I-13) DIV 5)#SYSNO);	44207910
	UNLOCK(0);	44207920
	IF NOT LDATE THEN	44207930
	FOR I:=0 STEP 1 UNTIL 3 DO	44207940
	BEGIN	44207950
	J:=@4060&I[30:46:2]; % CLEAR CONTENTION BITS	44207960
	P(WAITIO([J] INX @100000000,0,18),DEL);	44207970
	J,[37:1]:=1; % UNLOCK ADDRESSES	44207980
	P(WAITIO([J] INX @100000000,0,18),DEL);	44207990
	END;	44208000
\$ POP OMIT		44208001
	FOR I:=0 STEP 1 UNTIL 15 DO	44208900
	J = WAITIO(@4200000000,@377,1);%	44209000
	DISKWAIT(-A,30,DIRECTORYTOP=SYSNO);	44213000
	CORE,[4:14]:=IF DDD[9]=0 THEN 100 ELSE DDD[9],[4:14];	44213500
	OPTION = DDD[0];%	44214000
	IF (SCHEDWRD:=NABS(DDD[21]))=0 THEN	44214300
	SCHEDWRD:=(-1)&1[CTF];	44214310
	REMOTE=0;	44215000
	LOGARRAY[33]:=DDD[20],[30:18];	44216100
	LOGARRAY[32]=0;	44216110
\$ SET OMIT =	NOT SHAREDISK	44216690
	DISKWAIT(-A,-30,DIRECTORYTOP);	44216700
\$ POP OMIT		44216710
	XCLICK:=(@7777777777700 AND DDD[18]) MOD 5184000;	44216900

	NEUP:=DDD[2] MOD 100 ;	44216910
	NEUP:=NEUP&(NEUP+DDD[2] DIV 100)[CTF]&NEUP[3:33:15];	44216940
	DATE:=DDD[1];	44216950
	SPOUTIT(B,HALTK);	44216955
\$ SET OMIT =	NOT(SHAREDISK)	44216959
	UNLOCK(DIRECTORYTOP);	44216960
\$ POP OMIT		44216961
	STREAM(B:=0;A:=BYPASSI=DDD[4],DI=P(,DIRDSK));	44240500
	BEGIN SI:=LOC A; DS:=8 DEC; DI:=LOC B;	44240520
	SI:=LOC D; SI:=SI+8; DS:=WDS;	44240540
	END; I:=P INX 0; % GET LOCATION IN INITIALIZE	44240560
	DISKBOTTOM:=BYPASS-2;	44240580
	M[INTS-2]:=J:=*P(DUP))&I[CTC];	44240600
	M[J],[FF]:=I;	44240620
	M[I]:=J&(INTS-2)[CTF];	44240640
	FORGETSPACE(INTS); % RETURN PART OF INITIALIZE	44240660
	INTS:=I + 2;	44240680
	IF (T+NEUP,[FF]-NEUP,[CF])>0 THEN	44240690
	NEUP+NEUP&10[CTC]&(10+T)[CTF];%SAVE # OF EUS ON DKA,	44240695
	I:=NEUP,NEUF;	44240700
	Z:=(Y:=I + EUIOFFSET) + 1	44240800
\$ SET OMIT =	SHAREDISK	44240819
	+ (B:=(I+1) DIV 2 + I + 2)	44240820
\$ POP OMIT		44240821
	J:=GETSPACE(Z,0,1) + 2;	44240840
	MOVE(Z,J-1,J);	44240860
	EUIO:=(J INX M)&Y[8:38:10];	44240880
	PEUIO:=((J:=J+Y) INX M)&I[8:38:10];	44240900
\$ SET OMIT =	SHAREDISK	44240919
	AVTABLE:=((J+I) INX M)&B[8:38:10];	44240920
\$ POP OMIT		44240921
	T1:=GETSPACE(200,0,1)+2; % SPACE FOR INTRNSC	44241000
	CHUNKMAX*CHUNKMAX=1;	44241100
	FOR I=0 STEP 1 UNTIL CHUNKMAX DO	44241110
	IF (TWO(7-(CHUNKZIZE*I+FENCE) DIV 4096)AND MEMASK)≠0 THEN	44241120
	CT[I]:=(NOT 0),[36:12]; % ACTIVE[I]:=TOTAL[I]:=63;	44241130
	ACTDATE:=WEEKDAY:=0;	44241150
	MCP:="SITE ";	44241160
\$ SET OMIT =	NOT(SHAREDISK)	44241179
	SCRATCHVEC:=M[GETSPACE(10,SCRATCHTYPE,SCRATCHSAVE)+2]]&10[TOSIZE];	44241180
	TOGLE:=TOGLE OR SCRATCHDIRECTORYMASK OR USERDISKMASK;	44241190
	IF LDATE THEN	44241200
	BEGIN	44241250
	MOVE(4,[DDD[23]],[MULTITABLE[16]]);	44241260
	DISKWAIT(="EUIO,[CF],EUIO,[8:10],EUIOHOLDER) ;	44241275
	AVS+P(DDD[NEUP,NEUF],DUP),NUMENT+P(XCH),STARTWRD+AVDIFFMIN;	44241330
	AVS+(AVS+IF AVS >AVSMAX THEN AVSMAX ELSE IF AVS LSS AVSMIN	44241335
	THEN AVSMIN ELSE AVS)+30-(IF (AVS+AVS MOD 30)≠0 THEN AVS ELSE	44241336
	30);	44241337
	CLEANOUT(SYSNO);	44241350
	DISKWAIT(="(SCRATCHVEC INX 0),=4,DIRECTORYTOP+1);	44241375
	END ELSE	44241400
	SCRATCHVEC[0]:=SCRATCHVEC[1]:=SCRATCHVEC[2]:=SCRATCHVEC[3]:=0;	44241500
	SCRATCHVEC[SYSNO]:=0&(I:=GETESPDISK)[TOSLINK];	44241550
	DISKWAIT(SCRATCHVEC INX 0,=4,DIRECTORYTOP+1);	44241600
	SCRATCHVEC[0]:=0&1[TOSNUM];	44241650
	SCRATCHVEC[1]:=1&1[TOSLENGTH];	44241700
	DISKWAIT(SCRATCHVEC INX 0,2,I);	44241750
	SCRATCHVEC[0]:=0&(DIRECTORYTOP+1)[TOSLINK]&1[TOSSIZE];	44241800
	SCRATCHVEC[1]:=2&1[TOSNUM]&I[TOSLINK];	44241850



```

IF LDATE THEN MESSAGETABLEBUILDER ELSE 44241900
BEGIN 44241950
$ POP OMIT 44241951
    DIRECTORYBUILDER(A,DDD); 44242000
    FORGETSPACE(P(,DIRECTORYBUILDER,LOD),[CF]); 44242050
$ SET OMIT = NOT(SHAREDISK ) 44242090
    END ; 44242100
$ POP OMIT 44242101
    CREATELOG(DDD); 44242200
    FORGETSPACE(P(,CREATELOG,LOD),[CF]); 44242700
    CANDYINX:=CCTBLWORD:=0; 44242900
    TIMEOUT(SPACE(10)); DATEOUT(SPACE(10)); 44243000
    LASTSEG := FIRSTSEG := P((SPACE(32)+2)OR M) 44395100
        & 32[8:38:10],SFB)); 44395200
    MOVE(32, LASTSEG, [CF]=3, LASTSEG, [CF]=2); 44395220
    M[IOADR] + IOMASK; 44395230
    PROGTANK:=TANKADDRESS:=BASEDISKADR:=0; 44395300
    TOGGLE:=TOGGLE OR HOLDMASK; 44395320
    FIRSTOFFSET + LASTOFFSET + 1; 44395350
    IF CLOCK=0 THEN % CC103F IS INHIBITED 44395500
    BEGIN STREAM(T:=T:=SPACE(10)); 44395550
        BEGIN DS:=19 LIT"#TIMER NOT RUNNING,"; 44395600
            DS:=22 LIT" RESET CC103F INHIBIT="; 44395650
        END; 44395700
        SPOUT(T); 44395750
    END; 44395800
    IF GIVEDATE THEN% 44396000
        BEGIN;STREAM(B:=I:=SPACE(2)); 44397000
            DS + 11 LIT "#DT PLEASE="; % 44398000
            SPOUT(I); % 44399000
            DATE + "1"; % 44400000
        END; 44401000
    IF GIVETIME THEN% 44402000
        BEGIN;STREAM(B:=I:=SPACE(2)); 44403000
            DS + 11 LIT "#TR PLEASE="; % 44404000
            SPOUT(I); % 44405000
            XCLOCK + "5184000"; % 44406000
        END; % 44407000
$ SET OMIT = SHAREDISK 44407999
    DISKWAIT(-KLUMP,3,DIRECTORYTOP+3); 44408000
$ POP OMIT 44408001
    STASUS[0]+READYSTATE; 44408100
    CHANGEDATE(0); 44408200
    KEYBOARDCOUNTER:=0; 44408400
    TOGGLE:=TOGGLE OR HOLDMASK OR CDMASK OR KEYBOARDMASK; 44408500
$ SET OMIT = NOT(AUXMEM) 44408509
    IF (P(RRR),[30:1] AND USEDRE) OR (P(RRR),[31:1] AND USEDRA) THEN 44408510
        BEGIN 44408520
            AUXILIARYTABLEINITIALIZE; 44408530
            T:=SPACE(30); 44408540
        AUXAGN: DISKWAIT(-T,30,AUXMEMDSK); 44408550
            B:=M[T+(SYSNO*4)]; C:=M[T+(SYSNO*4)+1]; 44408560
            IF NOT (M[T+SYSNO+16]="AUXMEM " AND B GEQ 0) THEN 44408570
                BEGIN 44408580
                    AUXMESS: STREAM(I:=I:=SPACE(2)); 44408590
                        DS:= 15 LIT "#CA MCP PLEASE="; 44408600
                        SPOUT(I); CTABLE[4],[2:1] := 1; 44408610
                        SLEEP([CTABLE[4]],NOT CTABLE[4]); 44408620
                        CTABLE[4],[4:1] := 0; 44408630
                        GO AUXAGN; 44408640
                END;
        END;

```

```

END; 44408644
IF B#0 THEN 44408648
IF (I:=DIRECTORYSEARCH(B,C,4))=0 THEN 44408652
BEGIN 44408656
LBMESS(B,C,15,0,0,0,1); 44408660
GO TO AUXMESS; 44408664
END 44408668
ELSE 44408672
BEGIN 44408676
J:=M[(P(,ESPBIT))+1],[CF]; 44408680
TRANSFERMCPTOAUXMEM(I,J); 44408690
HEADERUNLOCK(B,C,I); 44408700
END; 44408710
FORGETSPACE(T); 44408715
END; 44408720
$ POP OMIT 44408721
SLEEP([CLOCK],NOT 0); 44408900
SLEEP([DATE],NOT(-1)); 44409000
WHILE XCLOCK<0 DO SLEEP([XCLOCK],NOT(XCLOCK)); 44409100
DISKWAIT(-A,-30,0); 44410000
IF (T:=DIRECTORYSEARCH(DDD[I:=13+5*SYSNO],DDD[I+1],17))=0 44410100
THEN BEGIN 44410200
NULLINT; 44410300
STREAM(T1); 44410400
DS:=27 LIT"## LOAD INTRINSICS NOW,...,@"; 44410500
SPOUT(T1); 44410600
END ELSE 44410700
IF (NT1:=M[T+4],[36:6])#0 AND NT1#TSSINTYPE THEN 44410800
BEGIN FORGETSPACE(T); 44411000
P(DIRECTORYSEARCH(NABS(DDD[I]),DDD[I+1],16),DEL); 44411200
DDD[I]:=DDD[I+1]:=0; % REMOVE INTRINSICS 44411400
DISKWAIT(A,-30,0); 44411600
GO TO NULLINT; 44411800
END ELSE 44411900
BEGIN INTRINSICTABLEBUILDER(T&T1[CF]); 44412000
FORGETSPACE(T); 44412100
END; 44412199
$ SET OMIT = NOT SHAREDISK 44412200
UNLOCK(0); 44412201
$ POP OMIT 44412300
INTFREE:=1; 44412400
IF (T:=DIRECTORYSEARCH(DDD[I-3],DDD[I-2],17))#0 THEN 44412500
BEGIN 44412599
$ SET OMIT = NOT STATISTICS 44412600
MCPTOP := MCPBASE+M[T INX 7]; 44412601
$ POP OMIT 44412700
FORGETSPACE(T); 44412800
END ELSE 44412900
LBMESS(DDD[I-3],DDD[I-2],-15,0,0,0,1); 44412999
$ SET OMIT = SHAREDISK 44413000
MCPFREE:=1; 44413001
$ POP OMIT 44413500
FORGETSPACE(A); 44414000
LOGOUTMAINT(SHLM); 44414100
MROW:=NABS(MROW); 44414200
FORGETSPACE(SHLM); 44415000
TOGGLE:=TOGGLE OR SHEETMASK OR STATUSMASK; 44417000
READY:=@343600000; % 44417100
SPREADTHEWORD; 44417110
FOR I+20 STEP 1 UNTIL 21 DO 44417120
P(WAITIO(@4000100000,@777,I),DEL);

```

TOGLE := TOGLE OR NSECONDMASK;	44417150
LASTSCHEDESELECT:=(XCLOCK DIV 54000)*54000;	44417200
SCHEDLOOK(0,=1); %REORDER SCHEDULE TASK QUEUE	44418000
	44420000
RRRMECH * RRRMECH AND @763777777777;%	44421000
READY * READY AND @763777777777;%	44422000
\$ SET OMIT = NOT STATISTICS	44422990
INTFINISH;	44423000
\$ POP OMIT	44423010
SWAPEND:=31;	44424000
IF AUTOCE THEN	44425000
BEGIN STREAM(T:=T:=SPACE(8)+2);	44425100
BEGIN DS:=21 LIT"CC RUN CANDE/TSHARER";	44425200
DS:=24 LIT"STACK=200;CORE=4000;END*";	44425300
END;	44425400
CCARD(T&25[3:43:5]);	44425500
END;	44425600
FORGETSPACE(INTS); FORGETSPACE(INTSS);	44437000
GO TO NOTHINGTOD0;%	44439000
END;%	44440000
SAVE REAL PROCEDURE COREND;%	44441000
BEGIN REAL T; P(INI); END;%	44442000
\$ SET OMIT = NOT STATISTICS	44999990
PROCEDURE INTFINISH;	45000000
BEGIN REAL B,T;	45001000
LABEL SS,SS1;	45003200
SS:    B:=0;	45200100
IF (T:=DIRECTORYSEARCH("SYSTEM "	45201000
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	45202000
&(SYSNO+17)[42:42:6]	45203000
\$ POP OMIT	45203001
,"STATS ",5))=0 THEN	45204000
BEGIN	45205000
T:=SPACE(30);	45206000
MOVE(30,T-1,T);	45207000
SS1:	45207100
M[T INX 0]:=00007400074000102;	45208000
STREAM(,X:=T INX 3);	45209000
BEGIN	45210000
SI:=LOC DATE; DS:=8 OCT; DI:=X;	45211000
DS:=2LIT"+#"; SI:=X; SI:=SI+5; DS:=3 CHR;	45212000
END;	45213000
M[T INX 7]:=1;	45214000
M[T INX 4]:=0&1[9:47:1]; %RESET COLD START BIT	45215000
IF NOT B THEN	45215100
BEGIN	45215200
M[T INX 9]:=1;	45216000
M[T INX 10]:=	45217000
GETUSERDISK(=(M[T INX 8]:=200));	45218000
ENTERUSERFILE(="SYSTEM "	45219000
\$ SET OMIT = NOT(SHAREDISK) OR OMIT	45220000
&(SYSNO+17)[42:42:6]	45221000
\$ POP OMIT	45221001
,"STATS ",T-1);	45222000
END ELSE DISKWAIT(T,[CF],30,T,[FF])	45222100
END	45222150
ELSE % FILE PRESENT	45222200
IF B:=((M[T INX 8]=200) AND (M[T INX 9]=1)) THEN	45222250
IF M[T INX 4],[45:1] THEN % JUST COLD-STARTED	45222275
GO TO SS1 % TO FIX-UP THE HDR	45222300

```

ELSE % HDR OK                                45222310
BEGIN                                         45222315
    DISKWAIT("COUNTARRAY,[CF],60,MIT INX 10)); 45222320
    IF COUNTARRAY[28] # DATE THEN COUNTARRAY[29]:=0; 45222325
END                                             45222330
ELSE                                          45222350
BEGIN % START A NEW FILE                    45222400
    FORGETSPACE(T);                          45222450
    P(DIRECTORYSEARCH("SYSTEM "             45222500
$ SET OMIT = NOT(SHAREDISK) OR OMIT          45222550
                                             &(SYSNO+17)[42;42;6] 45222600
$ POP OMIT                                   45222601
                                             ,"STATS ",6),DEL); 45222650
    GO TO SS;                                45222700
END;                                          45223000
SYSTATBASE:=MIT INX 10];                    45224000
INTERVAL:=108000;                           45226000
FORGETSPACE(T);                              45227000
COUNTARRAY[60]:=@37777777777777777;       45228000
END OF INTFINISH;                            45249000
$ POP OMIT                                   45249010
ARRAY LINKR=NT2[*],BAK#+2[*];                45990000
REAL LOGLYNE#+5;                             45990400
REAL CLICKS#+4;                              45990500
REAL LINQ = +1, MIX = +3, MASK = -1, TESTER = -2; 45991000
COMMENT THE ABOVE DEFINE ENTITIES USED BY BEDSEARCH, WHICH (EXCEPT 45991500
FOR "LINKR") EXIST IN THE STACK OF EACH CANDIDATE FOR AWAKENING 45992000
;                                             45992500
REAL FORKER = FORK;                           45993000
COMMENT FORKER IS USED BY SLATE INITIATE CODE TO FOOL FORK, 45993500
A MARK STACK CONTROL WORD IS PASSED AS THE LAST 45994000
PARAMETER, BUT ONLY THE CORE FIELD IS USED, AS LONG AS 45994500
THE HARDWARE WORKS, SO WILL THIS CODE; 45995000
ARRAY BLOB = NT3[*];                          45995500
COMMENT YOU MUST PROTECT "BLOB" THE SAME AS OTHER NT-TYPE 45996000
VARIABLES, IN OTHER WORDS, YOU MAY NOT CALL ANY 45996500
PROCEDURE AND EXPECT IT TO RETURN WITH THE SAME VALUE; 45997000
;16; P(.,COREND,LOD,4,INX,STS); INITIALIZE; % 20=1ST CODE 46000000
% ;17; % 21 = RES FOR NO MEM MSG 46000500
;18; GO TO TIMER; % 22 = TIME INTERVAL% 46001000
;19; GO TO IOBUSY; % 23 = I=O BUSY% 46002000
;20; GO TO KEYBOARDREQUEST; % 24 = KEYBOARD REQUEST% 46003000
;21; PRINTERFINISH(20); % 25 = PRINTER 1 FINISH% 46004000
;22; PRINTERFINISH(21); % 26 = PRINTER 2 FINISH% 46005000
;23; IOFINISH(RESULT1,1); % 27 = CHANNEL 1 COMPLETE 46006000
;24; IOFINISH(RESULT2,2); % 30 = CHANNEL 2 COMPLETE 46007000
;25; IOFINISH(RESULT3,3); % 31 = CHANNEL 3 COMPLETE 46008000
;26; IOFINISH(RESULT4,4); % 32 = CHANNEL 4 COMPLETE 46009000
;27; GO TO P2BUSY; % 33 = P2 BUSY% 46010000
;28; GO TO INQUEST; % 34 = DATACOM INQUIRY 46011000
;29; DO UNTIL FALSE; % 35 = SPECIAL INTERRUPT 46011500
$ SET OMIT = SHAREDISK                        46011990
;30; DO UNTIL FALSE; % 36 = DKA READ CHECK 46012000
;31; DO UNTIL FALSE; % 37 = DKB READ CHECK 46012500
$ SET OMIT = NOT SHAREDISK                    46012750
;30; GO FINDIT; % 36 = FREE ADDRESS 46013000
;31; GO FINDIT; % 37 = ALTERNATE FREE ADR 46013500
$ POP OMIT OMIT                               46013510
;32; P(0); GO TO P2PROCESS; % 40 = P2 MEMORY PARITY% 46014000
;33; P(4,17); GO TO P2PROCESS; % 41 = P2 INVALID ADDRESS 46015000

```

```

134: P(4,1);      GO TO P2PROCESS; % 42 = P2 STACK OVERFLOW 46016000
136: P(6);       GO TO P2PROCESS; % 44 = P2 COMMUNICATE% 46017000
137: P(8);       GO TO P2PROCESS; % 45 = P2 PROGRAM RELEASE 46018000
138: P(10);      GO TO P2PROCESS; % 46 = P2 CONTINUITY BIT 46019000
139: P(18);      GO TO P2PROCESS; % 47 = P2 PRESENCE BIT 46020000
140: P(12,0);    GO TO P2PROCESS; % 50 = P2 FLAG BIT 46021000
141: P(12,1);    GO TO P2PROCESS; % 51 = P2 INVALID INDEX 46022000
142: P(12,2);    GO TO P2PROCESS; % 52 = P2 EXP UNDERFLOW 46023000
143: P(4,9);     GO TO P2PROCESS; % 53 = P2 EXP OVERFLOW% 46024000
144: P(4,11);    GO TO P2PROCESS; % 54 = P2 KINT OVERFLOW% 46025000
145: P(12,3);    GO TO P2PROCESS; % 55 = P2 DIVIDE BY ZERO 46026000
148: P(0);       GO TO P1PROCESS; % 60 = P1 MEMORY PARITY% 46027000
149: P(4,17);    GO TO P1PROCESS; % 61 = P1 INVALID ADDRESS 46028000
STACKOVERFLOW 150: P(4,1);    GO TO P1PROCESS; % 62 = P1 STACK OVERFLOW 46029000
152: P(6);       GO TO P1PROCESS; % 64 = P1 COMMUNICATE% 46030000
153: P(8);       GO TO P1PROCESS; % 65 = P1 PROGRAM RELEASE 46031000
154: P(10);      GO TO P1PROCESS; % 66 = P1 CONTINUITY BIT 46032000
155: P(18);      GO TO P1PROCESS; % 67 = P1 PRESENCE BIT 46033000
156: P(12,0);    GO TO P1PROCESS; % 70 = P1 FLAG BIT 46034000
157: P(12,1);    GO TO P1PROCESS; % 71 = P1 INVALID INDEX 46035000
158: P(12,2);    GO TO P1PROCESS; % 72 = P1 EXP UNDERFLOW 46036000
159: P(4,9);     GO TO P1PROCESS; % 73 = P1 EXP OVERFLOW% 46037000
160: P(4,11);    GO TO P1PROCESS; % 74 = P1 INT OVERFLOW% 46038000
161: P(12,3);    GO TO P1PROCESS; % 75 = P1 DIVIDE BY ZERO 46039000
START:;
TIMER:;
$ SET OMIT = NOT(NEWLOGGING) 48000000
STOPLOG(P1MIX,0); 48000099
$ POP OMIT 48000100
IF CLOCK,[37:5] = 0 OR 48000101
(SECONDCTR + (P2MIX<=0)+SECONDCTR) >= 4 OR% 48000200
XCLOCK GEQ WITCHINGHOUR THEN 48001000
BEGIN IF P(T10) # 0 THEN% 48002000
IF FIRSTWAIT # NEXTWAIT THEN% 48003000
NEWIO;% 48004000
SECONDCTR + 3;% 48005000
IF NSECONDREADY THEN% 48006000
BEGIN TOGGLE+TOGGLE AND NOT NSECONDMASK; 48007000
FORK(P(,NSECOND),0,-1,128,1); 48008000
END 48009000
END;% 48010000
$ SET OMIT = NOT(STATISTICS) 48010999
COUNTUPBY(3,1); 48011000
COUNTUPBY(14,SECONDCTR LEQ 1); 48011100
COUNTUPBY(38,(P1MIX=CANDYINX) AND (CANDYINX NEQ 0)); 48011200
COUNTUPBY(7,NT1:=UNIT[18],[13:5] NEQ 0); 48011300
COUNTUPBY(8,NT2:=UNIT[19],[13:5] NEQ 0); 48011350
COUNTUPBY(17,(NT1+NT2)=2); 48011400
COUNTUPBY(43,(P1MIX=0) AND (NT1:=(NT1 OR NT2))); 48011450
COUNTUPBY(39,(NT2:=(P1MIX NEQ 0)) AND NT1); 48011470
IF (P2MIX GTR 0 AND NT1) THEN COUNTUPBY(39,LEFTHALF1); 48011480
COUNTUPBY(6,PBUSY); 48011500
IF NT2 THEN COUNTUPBY(6,LEFTHALF1); 48011600
COUNTUPBY(31,NT1:=P2MIX GTR 0); 48011700
IF (NT2 AND NT1) THEN COUNTUPBY(31,LEFTHALF1); 48011800
$ POP OMIT 48011801
IF (P(RRR) OR RRRMECH)#READY THEN 48012000
IF STATUSBIT THEN 48012500
BEGIN TOGGLE+TOGGLE AND NOT STATUSMASK; 48013000
FORK(P(,STATUS),0,-1,128,1); 48014000
END;% 48015000

```

```

                IF P2MIX>0 THEN
                    GO TO P2FAKE;
EXTERNAL:;%
                IF P1MIX = 0 THEN GO TO NOTHINGTODO;%
INITIATE:;%
                P(NT1+PRT[P1MIX,8],STS,0,STF);
                IF P2MIX=0 THEN GO TO COMINIT;
                IF (FORKQUE INX 0)=P(,FORKQUE) THEN
                IF BED,[CF]=P(,BED) THEN
COMINIT:;%
                IF NOPROCESSTOG < 0 THEN%
GOGOGO:    BEGIN IF PRT[P1MIX,0] ≠ WORDOFFEASE THEN
                BEGIN P(64,STS);
$ SET OMIT = NOT(NEWLOGGING)
                STARTLOG(P1MIX,0);
$ POP OMIT
                GO TO STACKOVERFLOW;
                END;
                P(IN1);%
                IF PRTRW[P1MIX],[PSF]≠0 THEN
                BEGIN IF (NT3+PRTRW[P1MIX],[PSF])=1 THEN
                    TERMINALMESSAGE(PRTRW[P1MIX],[FF]);
                    IF NT3=2 THEN STOPM ELSE SWAP(FORCESWAP,1);
                    GO TO RETURN;
                END;
                IF OLAYCTR[P1MIX] GEQ 0 THEN
                IF ELAPSEDLIMIT[P1MIX] GTR IOTIME[P1MIX] THEN
                    IF IOTIME[P1MIX]>0 THEN
                        BEGIN
                            TERMINATE(P1MIX&83[CTF]); GO GOGOGO
                        END ELSE
                        BEGIN
                            NT3+PROCTIME[P1MIX];
$ SET OMIT = NOT(NEWLOGGING)
                            IF NOT LOGSTOPPED[P1MIX] THEN
$ POP OMIT
                                NT3+NT3+CLOCK+P(RTR);
                                IF NT3<PROCLIMIT[P1MIX] THEN
                                IF NT3>0 THEN
                                    BEGIN TERMINATE(P1MIX&15[CTF]);
                                        GO GOGOGO;
                                    END ELSE
                                        BEGIN SECONDCTR+0;
$ SET OMIT = NOT(NEWLOGGING)
                                        STARTLOG(P1MIX,0);
$ POP OMIT
                                IF P2MIX≠0 THEN
                                    P(NT1,IP1);
                                    P(NT1,IP2);
                                    P2MIX ← P1MIX;%
                                    LOGLINE2 ← LOGLINE;
                                    GO TO NOTHINGTODO;
                                END;
                                END;
                                IF P(0,RDS) GTR FENCE THEN
                                    FOR NT2+SC[P1MIX] STEP 1 UNTIL LC[P1MIX] DO
                                        IF ACTIVE[NT2] GTR 1 OR OLAYCTR[P1MIX]
                                            LSS 0 THEN
                                            BEGIN SWAP(TIMEND,1); GO TO RETURN END;

```

```

48015100
48015500
48016000
48017000
48018000
48019000
48019500
48020000
48021000
48022000
48023000
48024000
48025000
48025099
48025100
48025101
48025200
48025300
48026000
48027000
48028000
48029000
48030000
48031000
48031100
48031200
XR3848031290
XR5948031300
48031310
48031320
48031330
48031340
48031350
48031360
48031369
48031370
48031371
48031380
48031400
48031600
48031700
48031800
48031900
48032000
48032979
48032980
48032981
48033000
48034000
48035000
48037000
48037100
48038000
48038100
48038110
XR3848038150
48038200
XR3848038300
XR3848038310
48038400

```

```

PROCLIMIT[P1MIX]:=*P(DUP)+208; %DS48038500
ELAPSEDLIMIT[P1MIX]:=*P(DUP)+416; %DS %R5948038600
OLAYCTR[P1MIX]:=ABS(*P(DUP))&28[CTF]; % APPROX 90000048038610
SLN[P1MIX]:=SLN[P1MIX]+(SLN[P1MIX] NEQ 7); %DS48038640
NLS[P1MIX]:=SLN[P1MIX]+2; %DS48038680
IF LOGLINE,[33:7]#0 THEN
    P([STATABLE[LOGLINE,[40:8]]],IOR); 48038700
GO GOGOGO; 48038800
END; % 48038900
P(INI); % 48039000
SLEEP([NOPROCESSTOG],-0); % 48040000
NT1 + PRT[P1MIX,8]; % 48041000
GO GOGOGO; % 48042000
NOTHINGTODO; P1MIX + 0; % 48043000
$ SET OMIT = NOT(STATISTICS) 48044000
PBUSY:=1; 48045000
$ POP OMIT 48045899
P(INI); % 48045900
P(64, STS); 48045901
IF AREASNEEDED THEN 48046000
BEGIN AREASNEEDED:=FALSE; 48046500
    FORK(P(,MOREAREAS),0,-4,96,1); 48046600
END; 48046600
IF (FORKQUE INX 0)#P(,FORKQUE) THEN 48046700
BEGIN IF (NT1 + FORKQUE[4])#0 THEN 48046800
    BEGIN P(NT1,STS,0); 48046900
SLATESTARTER; P(SECONBCTR+0, STF); NT6 + FORKQUE,[CF]; 48047000
    P(FORKQUE[1],FORKQUE[2]); 48047500
    PRIORITY + FORKQUE[0],[9:9]=64; 48048000
    M[NT2 + FORKQUE[0],[CF]],[FF] + P(,FORKQUE); 48048500
    FORKQUE,[CF] + NT2; 48049000
    FORGETAREA(0,NT6); NT4 + P(,BLOB,+); 48049500
    IF (NT1 + P)#0 THEN 48050000
    BEGIN P(NT4,BLOB,NT1,GETSPACE(NT1,12,3)); 48050500
        TOGLE + TOGLE OR STACKMASK; 48051000
        IF (NT1 + P+1)=1 THEN 48051500
        BEGIN P(PRIORITY, XCH, MKS, FORKER); 48052000
            GO TO NOTHINGTODO; 48052500
        END; 48053000
        P(DEL, ,BLOB,+ ,NT4,+ ,NT1,STS); 48053500
    END; 48054000
    LOGLINE:=NT4,[FF]&NT4[1:1:1]; 48054500
P(CLOCK&NT4[8:38:10],MKS,NT4,DIB 0,LOD,BLOB,COC); 48055000
GO TO NOTHINGTODO; 48055500
END; 48055600
IF STACKUSE THEN 48056000
BEGIN TOGLE + TOGLE AND NOT STACKMASK; 48056500
    P(ISTACK, STS, FORKQUE[3]); 48057000
    GO TO SLATESTARTER; 48057100
END; 48057200
END; 48057300
P( 64,STS); % 48057400
IF TOGLE THEN GO TO PROCSWIT; COMMENT TEST HP2TOG; 48057500
LINKR + FLAG(BED); 48057600
WHILE LINKR,[CF]#P(,BED) DO 48058000
BEGIN P(INI,[LINKR[FREG]],STS,DUP,STF); 48059000
$ SET OMIT = NOT(NEWLOGGING) 48060000
P(0,RDS,1,+ ,STS); % FOR LOGTURNEDOFF 48061000
$ POP OMIT 48062000

```

P1MIX ← MIX; % MIX = F+3, MASK=-1, TESTER=-2	48063000
IF (NOT(MASK AND TESTER))≠NOT 0 OR	48064000
CLOCK+P(RTR)>CLICKS THEN	48064100
BEGIN	48065000
P(BAK[0]+P(DUP,LOD)&LINQ[CTC],1,CDC,BAK,XCH,+);	48067000
SECONDCTR ← 0;	48069000
LOGLINE ← LOGLYNE;	48069100
PRIORITY ← LINQ,[FFJ]-64;	48069200
\$ SET OMIT = NOT(NEWLOGGING)	48069299
IF LOGTURNEDOFF THEN	48069300
\$ POP OMIT	48069301
STARTLOG(P1MIX,0);	48069400
P(XIT);	48070000
END;	48071000
P1MIX ← 0;	48072000
LINKR ← FLAG(LINQ);	48073000
END BED SEARCH;	48074000
\$ SET OMIT = NOT(STATISTICS)	48079999
PBUSY←0;	48081000
\$ POP OMIT	48081001
DO DO BEGIN P(INI);	48082000
END UNTIL (P(RRR) OR RRRMECH)≠READY	48087000
UNTIL STATUSBIT;	48088000
TOGGLE:=TOGGLE AND NOT STATUSMASK;	48089000
FORK(P(,STATUS),0,-1,128,1);	48090000
GO TO NOTHINGTODO;	48091000
P2FAKE:	48092000
TOGGLE:=TOGGLE OR HP2MASK;	48092000
\$ SET OMIT = NOT(NEWLOGGING)	48092099
STOPLOG(P2MIX,0);	48092100
\$ POP OMIT	48092101
P(HP2,INI);	48093000
\$ SET OMIT = NOT(NEWLOGGING)	48093099
STARTLOG(P1MIX,0);	48093100
\$ POP OMIT	48093101
PROCSWIT:	48094000
P(16);	48095000
P2PROCESS:;%	48096000
IF P(P1MIX,P2MIX,,P1MIX,+,,P2MIX,STN) ≠ 0 THEN%	48097000
BEGIN	48097010
P(PRT[P2MIX,8],IP2);	48097020
END;	48097100
PRIORITY←PRYOR[P1MIX];	48097200
P(LOGLINE,LOGLINE2,,LOGLINE,+,,LOGLINE2,+);	48098000
TOGGLE←TOGGLE AND NOT HP2MASK;	48099000
P1PROCESS:;%	48100000
P(PRT[P1MIX,8],STS,0,STF);%	48101000
GO TO P(ONEOHONE);%	48102000
GO TO MEMORYPARITY;           % 0%	%WF 48102100
P(NOP,NOP);                   % 2%	48103000
GO TO NORMALERROR;           % 4%	48104000
SHORTCOMMUNICATES;           % 6%	48105000
PROGRAMRELEASE;              % 8	48106000
CONTINUITYBIT;                % 10	48107000
INTERRUPT(ONEOHTWO); P(NOP); % 12	48108000
GO TO INITIATE;               % 16	48109000
MAKEPRESENT(ANALYSIS);       % 18	48110000
RETURN:;           P(NT1←PRT[P1MIX,8],STS,0,STF);	48111000
GO TO COMINIT;	48117000
IOBUSY:;	48117099
\$ SET OMIT = NOT(NEWLOGGING)	48117100
STOPLOG(P1MIX,0);	



\$ POP OMIT	48117101
NT1 ← UNIT[NT2+CHANNEL[0]];	48117200
UNIT[NT2] ← NT1&0[13;5;5];%	48118000
STARTIO(NT2);%	48119000
GO TO EXTERNAL;%	48120000
P2BUSY;:	48121000
\$ SET OMIT = NOT(NEWLOGGING)	48121099
STOPLOG(P1MIX,0);	48121100
\$ POP OMIT	48121101
SAVEMIX(P1MIX,LOGLINE);	48121200
PRIORITY←PRYOR[P1MIX+P2MIX];	48122000
P2MIX ← -1;%	48123000
GO TO EXTERNAL;%	48125000
\$ SET OMIT = NOT(SHAREDISK)	48125099
FINDIT;:	48125100
\$ SET OMIT = NOT(NEWLOGGING) OR OMIT	48125109
STOPLOG(P1MIX,0);	48125110
\$ POP OMIT	48125111
IF NOT FINDINGADDRESS THEN	48125150
BEGIN	48125200
FINDINGADDRESS:=1;	48125250
FORK(P(,FINDFREEADDRESS),1,-2,128,1);	48125300
END;	48125350
GO TO EXTERNAL;	48125400
\$ POP OMIT	48125401
INQUEST;:	48125500
\$ SET OMIT = NOT(NEWLOGGING)	48125509
STOPLOG(P1MIX,0);	48125510
\$ POP OMIT	48125511
\$ SET OMIT = NOT(SAVERESULTS OR DEBUGGING)	48126000
STORAWAY:=UNIT[30];	48126001
\$ POP OMIT	48126002
\$ SET OMIT = NOT SEPTICTANK	48126009
DISPOSAL(P,P,0);	48126010
\$ POP OMIT	48126011
INTRGATCTR:=INTRGATCTR + 1;	48126100
IF (NOT UNIT[30]),[FF]=0 AND REMOTE THEN NEXTDCIO;	48126200
GO TO EXTERNAL;%	48127000
KEYBOARDREQUEST;:%	48128000
\$ SET OMIT = NOT(NEWLOGGING)	48128099
STOPLOG(P1MIX,0);	48128100
\$ POP OMIT	48128101
\$ SET OMIT = NOT DEBUGGING	48128500
NOBACKTALK ← TRUE;%	48129000
IF NOT KEYBOARDREADY THEN GO TO EXTERNAL;%	48130000
\$ POP OMIT	48130500
IF (KEYBOARDCOUNTER:=P(1 INX KEYBOARDCOUNTER)) = 1 THEN	48131000
FORK(P(,KEYIN),1,0,192,0);	48132000
GO TO EXTERNAL;%	48133000
MEMORYPARITY;:%	%WF 48134000
TERMINATE(P1MIX);%	%WF 48135000
TERMINALMESSAGE(32);%	%WF 48136000
NORMALERROR;:%	48137000
IF P1MIX = 0 THEN%	48138000
BEGIN P(@100,STS);%	48139000
PUNT(6);      % INVALID ADDRESS	48140000
END;%	48141000
IF ONEHTWO=1 THEN	48141100
BEGIN P(SINFO[P1MIX],STS);	48141200
INTABLEROW[P1MIX]+0;	48141210

KEYBOARD REQUEST

```
      PRT[P1MIX,15]=M[PRT[P1MIX,8]];
      PRT[P1MIX,8]=-[PRT[P1MIX,15]];
      PRT[P1MIX,3]=FPBD[P1MIX];
      PRT[P1MIX,4]=SEGD[P1MIX];
END;
P(ONEOHTWO);
  IF P(DUP,DUP)=9 OR P(XCH)=11 THEN
    ERRORFIXER((ONEOHTWO=9)+1);
  TERMINATE(P1MIX);
  NT1 = P;
  TERMINALMESSAGE(NT1);
DIFFCOM:;NT4=P;
  P(O,STF,PRT[P1MIX,8],STS,MKS,NT4,DIB O,LOD,XCH,COC);
  GO TO INITIATE;
END,%
```

```
48141300
48141400
48141500
48141600
48141700
48142000
48142100
48142200
48142500
48143000
48143500
48144000
48145000
48146000
48161000
```

