Setup for Mach 3.0

Super root and local root

Mach systems (2.5 or 3.0) are normally set up with a super root that contains one real directory ***RFS***. ***RFS*** contains the directory ***.LOCALROOT*** and links to all other machines that are accessible via RFS. The super root also contains symbolic links to all the standard directories in /, eg. ***/dev, /mnt, /bin, /etc, /lib, /usr, /usr1***. ***/RFS/.LOCALROOT*** is known as the local root. After boot time the local root is named ***/*** and the super root is named ***/../../***. Some files such as the Mach kernel and the Unix-server have hard links in both the local root and super root, so that they can be found by the same name during and after the boot sequence. In the Mach 3.0 case the switching of the root, ***/***, from the super root to the local root is done by the Unix-server or POE rather than the micro-kernel.

Kernel names

In Mach 2.5 the kernel is called both ***mach*** and ***vmunix*** and is hard-linked to by ***/../../vmunix, /../../mach, /mach*** and ***/vmunix***. Some boot code looks for the file ***/vmunix*** by default and some looks for ***/mach*** but always on the super root. As the kernel boots, the root is changed to be the local root, ***/RFS/.LOCALROOT***. After booting some Unix programs read the "Unix namelist" by referencing ***/vmunix***.

In Mach 3.0 the names ***mach*** and ***mach_kernel*** are used for the Mach micro-kernel and ***vmunix*** and ***startup*** are are used for the Unix-server. ***/mach*** is the program that you wish to boot, so if your boot code does not boot mach by default you will have to specify it as the boot file name. The Mach micro-kernel is named ***mach_kernel*** by the build process and in the release distribution. The Unix-server is named ***vmunix*** and ***vmunix.UKnn.<config>*** in the build and ***startup*** in the release directory.

/mach_servers directory

The micro-kernel looks in the ***/mach_servers*** directory on the super root for three files: ***/mach_servers/startup*** which is the program the mach_kernel will start the user task in; ***/mach_servers/emulator*** which the kernel reads to initialize the symbols for the kernel debugger; and ***mach_servers/paging_file*** which is the default paging space. These names may refer to files in that directory, symbolic links to files on the root partition or symbolic links to files on other *unmounted* partitions. See the next section for an example of this last case.

The Unix-server looks in the /mach_servers directory on the local root for ***/mach_servers/emulator***, the emulation library that is loaded with each task and ***mach_servers/mach_init*** which is the first program called by the Unix-server. These programs must reside on the root partition, since the Unix-server only has the root partition mounted and does not read unmounted partitions like the micro-kernel did.

/mach_servers may be a subdirectory in either the local root or the super root with a symbolic link to it in the other root or there can be two different directories. The directory(s) must be on the root partition. NOTE: Some sites have reported that the mach_servers directory must be on the superroot although both locations are used on machines at CMU.

Setting things up to boot Mach 3.0

First you need to install a Mach 3.0 micro-kernel on the root. Copy a "mach_kernel" to ***/mach*** and then hard-link it to ***/../../mach***. The mach_kernel is built from the sources in the mach3.kernel collection or can be found in mach3.release in the directory ***special***.

Now you need to make a ***/mach_servers*** directory, either:

```
mkdir /../../mach_servers
cd /
ln -s ../../mach_servers mach_servers

or

mkdir /mach_servers
cd /../..
ln -s RFS/.LOCALROOT/mach_servers mach_servers
```

Now populate ***/mach_servers*** with:

mach_init is built from sources in the mach3.user collection in the directory ***etc/mach_init*** or can be found in mach3.release ***etc/mach_init***. It must be installed as ***/mach_servers/mach_init***. THIS PROGRAM IS NOT COMPATIBLE WITH THE MACH 2.5 VERSION. You may set ***/mach_servers/mach_init*** to be a symbolic link to the Mach 2.5 version of ***/etc/init***. In this case the system will boot, but Mach servers will not work.

emulator the emulator library which the Unix-server loads with each task and the micro-kernel reads symbols from. This program is built from the sources in the mach3.unix collection in the directory ***emulator*** or can be found in the mach3.release collection in the directory ***special***. It must be accessible from both the local root and the super root.

startup the program that the micro-kernel calls. Usually this is the Unix-server or POE. It could be an OS program of your own. The Unix-server is built from the sources in the mach3.unix release in the directory ***server*** or can be found in the mach3.release collection in the directory ***special***.

paging_file this is usually a symbolic link to a file on an umounted partition in order to avoid using large amounts of space on the root partition. For example if ***/usr1*** is mounted on ***/dev/hd0f*** and has enough free space you could make ***paging_file*** be a symbolic link to "/dev/hd0f/PAGING_FILE". Then you ***cd*** to ***/usr1*** and create a ***PAGING_FILE*** of sufficient size to handle your paging needs. Say 4-20M. The system does not grow this file, so you must set it up in advance as a big file. The easiest way to grow the space is to do:

```
dd if=/dev/rhd0f of=/usr1/PAGING_FILE bs=1024k count=n
```

which gets you a nMeg file.

Mach 3.0 Servers

If it exists ***/mach_servers/mach_init*** is the first program called by the Unix-server. If it does not exist ***/etc/init*** is called and your system will boot but none of the mach servers will work. ***mach_init*** sets up the ports to the netmsgsever, envmgr and service (itself) and then all the rest of the processes on the system inherit these ports. NOTE: do not install this program as ***/etc/mach_init***, if you are also booting the Mach 2.5 kernel. ***/etc/mach_init*** is the first program called by the Mach 2.5 kernel and the Mach 3.0 version of this program does not work with Mach 2.5.

The collection mach3.release contains executables of mach3 programs and servers. They can be installed anywhere on your system and your PATH variable set to find them.

It is assumed below that you have copied the contents of the mach3.release collection (except for the directory ***special***) to ***/usr/mach3***. There are two servers that we normally run:

- ***/usr/mach3/bin/sname*** which sets up a local only name service.
- ***/usr/mach3/bin/machid*** which sets up the machid services.

Accessing AFS

If your site is licensed to run AFS, Mach 3.0 provides access to AFS files using a very old out-of-kernel Venus server. To run Venus you need to take the following steps.

- You need to have the following files in ***usr/vice/etc***:
  - launchvenus.csh
  - runvenus
  - venus2
  - vstab
- `mknod /dev/fs0 c <#> 0`

  where <#> is the major device number for Venus and has the following values:
  - i386 - 17
  - pmax - 12
  - vax - 60
  - sun3 - 34
- Edit ***/usr/vice/etc/vstab*** to specify where you want your cache directory and how big it should be.
- It turns out that for reasons that are now lost in the mists of time, the name that you specify as the cache directory must be a symbolic link to the real cache directory. So now you need to create a cache directory and set its mode to ***700*** and then create the symbolic link that you named in ***vstab***.
- Finally you need to be sure that you have a ***/usr/adm/log*** directory as launchvenus.csh will want to write a ***venus.log*** file there.