MACH 2.5 Sun 3 Installation Notes

Jonathan Chew
Karl Hauth
Carnegie Mellon University
Pittsburgh, PA 15213
21 December 1989

1. Introduction

The distribution of Mach for the Sun 3 contains a whole operating system. It is intended to be loaded onto an empty disk and booted. The tape(s) contains a Mach kernel, the CMU-CS 4.3BSD environment, Mach support programs/libraries, X11, gnu emacs, tools for source control and building from source, and sources for the Mach kernel and environment and the compilation environment to compile them under.

This document assumes at least a working knowledge of the installation and maintenance of a 4.3BSD system. A working knowledge of the installation of a SunOS system is highly desirable, but may not be necessary. If you are unfamiliar with one or both 4.3BSD and SunOS, you should read the 4.3BSD and/or SunOS documentation.

2. Information About the Tape

- 1/2" tape: 1 tape, 6250bpi, device = /dev/rmt{16,20}, blocking factor=40 (This is the PREFERRED tape distribution)
- 1/2" tape: 2 tapes, 1600bpi, device = /dev/rmt{8,12}, blocking factor=40
- 1/4" tape: 3 450ft. tapes, device = /dev/{,n}rst8, blocking factor=1024

2.1. Contents

1. Programs to help set up the disk and load it (tar)

    stand/diag
    etc/{disktab,fsck,mkfs,newfs,restore}
    bin/tar

2. / — 7073K (dump)
3. /sys0 — 22119K (tar)
4. /usr0 (tar)

adm	41K
guest	16K
mach	6631K
spool	9K
tmp	1K

5. Source for Mach kernel and environment, compilation environment, kernel dependent programs, and some useful CMU specific programs (tar)

    build/src — 20739K

6. Miscellaneous software (tar)

    misc — 32869K

.X11	12757K
.gnu-emacs	9503K
.nfs	924K
.rcs	1742K
.sdm	598K

3. Hardware Requirements

To install this distribution, you will need a Sun 3 with:

- A minimum of 4 megabytes of memory (>= 8 meg. is preferred)
- A bwtwo, cgtwo, or cgfour monitor
- At least one formatted 70 megabyte disk

  NOTE: A bootable system consisting of items 2-4 in the previous section will fit on a 70 megabyte disk. 140 megabtyes of disk space will comfortably accomodate everything.

- A server, a 1/4 inch tape drive, or a second disk to boot from

Mach has been tried (and is being used) here on the following models of the Sun 3:

Sun 3/{50,60,75,110,140,160,280}

4. Installation Procedure

Basically, you'll need to boot your machine, read in the programs to help set up and load the disk, partition the disk, make 4.3 file systems on it, and restore and un-tar the contents of the tape(s).

There is a script of a sample installation at the end of this document.

1. Boot your machine with your last version of Mach or SunOS 3.X

   You can boot SunOS on your machine in one of the following ways:

   a. Boot your machine as a diskless node off a server

   b. Load mini-Unix from 1/4 inch tape into the b partition of the local disk and boot it

   c. Boot from a second local disk

   Your local Sun maintainer should be able to help you do one of the above. There is also a Sun manual called "Installing UNIX on the Sun Workstation" that might be helpful.

2. Save your versions of /stand/diag, /etc/{disktab,fsck,mkfs,newfs, restore}, and /bin/tar by renaming them

   NOTE: Of these programs, the only one that you MUST have is /etc/mkfs to make 4.3 file systems instead of Sun file systems. The other programs are optional.

3. Cd to / and un-tar the first file on the tape which contains the programs to help set up and load the disk ie.

   stand/diag
   etc/{disktab,fsck,mkfs,newfs,restore}
   bin/tar

4. Halt the machine

5. Boot /stand/diag to partition your disk(s)

   /stand/diag has been modified to include CMU partition tables as menu items for the disks that we use here. You can use these, the non-CMU tables, or your own. The nice thing about the CMU tables is that they are laid out so everything will fit.

```
      -         -
      |    a    |
      |         |
      |         -
      |    b    |
    c |         -         -         -
      |         |    d    |    f    |
      |         |         -         |
      |    h    |         |         -
      |         |    e    |    g    |
      |         |         |         |
      -         -         -         -
```

   Figure 4-1: Standard CMU disk partition layout

   In the CMU scheme, 70 megabyte disks don't have a "b" partition.
   With big disks, d and e split f instead of h. You can look in
   /etc/disktab for examples of partition tables.

   Here's the contents of /etc/fstab (for a Priam 638) which will give you an idea of how partitions can map to file systems:

```
/dev/sd0a:/:rw:1:1
/dev/sd0d:/sys0:rw:1:2
/dev/sd0b:/tmp:rw:0:5
/dev/sd0e:/usr0:rw:1:3
/dev/sd0g:/usr1:rw:1:4
```

   See Sun's "Installing UNIX on the Sun Workstation" and/or "Diag - A Disk Maintenance Program" if you don't know how to use diag.

6. Reboot machine with your last version of Mach or SunOS 3.X

7. Make the file systems for your disk(s) using /etc/mkfs or /etc/newfs. See /etc/disktab for a list of disks newfs understands.

   NOTE: You should use 8k file systems on your Sun 3

8. Restore the / dump into the "a" partition

9. Un-tar the /sys0 tar file

   NOTE: The files in the this tar file are stored by relative instead of full path names. Also, the tape may not be positioned correctly, so you may get the error message "tar: blocksize 0". To remedy this, either try re-issuing the tar command or rewind the tape and skip forward to the tar file.

10. Un-tar the /usr0 tar file

    NOTE: The files in the this tar file are stored by relative instead of full path names. Also, the tape may not be positioned correctly, so you may get the error message "tar: blocksize 0". To remedy this, either try re-issuing the tar command or rewind the tape and skip forward to the tar file.

11. Un-tar the sources into one of your user partitions

    NOTE: The files in the this tar file are stored by relative instead of full path names. Also, the tape may not be positioned correctly, so you may get the error message "tar: blocksize 0". To remedy this, either try re-issuing the tar command or rewind the tape and skip forward to the tar file.

12. Un-tar miscellaneous programs

    NOTE: The files in the this tar file are stored by relative instead of full path names. Also, the tape may not be positioned correctly, so you may get the error message "tar: blocksize 0". To remedy this, either try re-issuing the tar command or rewind the tape and skip forward to the tar file.

13. Use /usr/mdec/installboot to install a boot block on your disk
14. Halt the machine
15. Boot Mach single user

    NOTE: Mach will come up running the default kernel without NFS or AFS support. See the section on Changing Kernels, for information on how to get a different kernel.

16. Fsck the root partition
17. Su to root (the root password is "password")
18. Edit /etc/THISHOST to contain your machine name
19. Edit /etc/NETWORKS to contain your IP address
20. Create /etc/nmbroadcast with a single line containing your local broadcast address.
21. Edit /etc/attributes, changing "changeme.mach.cs.cmu.edu" to the name of your machine

    This is a control file unique to CMU which contains a single line entry for each machine in the local CMU network environment specifying the particular characteristics of that machine. It is mainly used by /usr/cs/etc/nanny, a CMU developed utility for starting and managing servers, to determine which servers need to be started for a particular machine.

22. Fsck /sys0
23. Mount /sys0 since /etc/hosts is linked to /usr/cs/lib/hosts/hosts in /sys0
24. Edit /etc/hosts to contain your machine name and IP address
25. Edit /etc/neton

    /etc/neton sets the host name and address of the machine, and configures the network interface. It needs to be changed to specify your network broadcast address and add your gateway to the routing table. If your machine won't be on the network, then everything except the first few lines which set the host name and host ID should be commented out.

26. Edit /etc/fstab to contain your disk partitions and mount points
27. CTRL-d out of your root shell
28. CRTL-d out of the opr shell
29. Hit return when asked 'Do you wish to run a "fsck -p"? [yes] '
30. When it comes up multi-user, login as guest (no password)

    The guest account was set up to show you how an account might be set up. It execs X when you login.

5. References

- 4.3BSD Unix Programmer’s Manual
- Diag - A Disk Maintenance Program
- Installing UNIX on the Sun Workstation
- Sun Commands Reference Manual

6. Sample Installation Session

Sample session for installing Mach on a Sun 3/160 on the network with a formatted Priam 638 disk, a 1/4" Mach tape distribution, and a SunOS 3.X boot tape. User input is in **bold face**.

Session output	Comments
	Start with a halted machine
>**b st()**	*Boot from the 1/4" tape*
Boot: st(0,0,0)	
Boot:**st(0,0,4)**	*Load the copy program*
Standalone Copy	
From:**st(0,0,5)**	*Copy mini-unix*
To:**sd(0,0,1)**	*into the b partition*
Boot:**sd(0,0,1)vmunix -as**	*Boot mini-unix*

[... about 20 lines of mini-unix messages ...]

root device?**sd0***

[... more mini-unix messages ...]

# **mv /stand/diag /stand/diag.old**	*save our versions*
# **mv /etc/disktab /etc/disktab.old**	
# **mv /etc/fsck /etc/fsck.old**	
# **mv /etc/mkfs /etc/mkfs.old**	
# **mv /etc/newfs /etc/newfs.old**	
# **mv /etc/restore /etc/restore.old**	
# **mv /bin/tar /bin/tar.old**	
# **cd /**	
# **/bin/tar.old xbf 1024 /dev/rst8**	
# **sync**	
# **sync**	*Get ready to halt*
	We have to use L1 a to halt since
Abort at *some-number*	*mini-unix doesn't have halt*
> **b st()**	
Boot: st(0,0,0)	
Boot: **sd(0,0,1)stand/diag**	*partition the disk*
Version 1.1 86/02/03	
Disk Initialization and Diagnosis	

When asked if you are sure, respond with 'y' or 'Y'

specify controller:
    0 - Interphase SMD-2180
    1 - Xylogics 440 (prom set 926)
    2 - Xylogics 450/451
    3 - Adaptec ACB 4000 - SCSI/ST506
    4 - Emulex MD21 - SCSI/ESDI
which one? **4**
Specify controller address on the mainbus (in hex): **200000**
Device address: 200000

Which target? **0**

Which unit? **0**

```
Specify drive:
      0 - Micropolis 1355
      1 - Toshiba MK 156F
      2 - Priam 638
      3 - Other
which one? 2

ncyl 1218 acyl 6 nhead 15 nsec 34 interleave 1
diag> partition
Select partition table:
      0 - CMU Micropolis 1355
      1 - Toshiba MK 156F
      2 - CMU Priam 638
      3 - Other
Which one? 2
Do you wish to modify this table? no
Verify partition table 'CMU Priam 638':
      Partition a: starting cyl=0, # blocks=28560
      Partition b: starting cyl=56, # blocks=32640
      Partition c: starting cyl=0, # blocks=621180
      Partition d: starting cyl=120, # blocks=932330
      Partition e: starting cyl=303, # blocks=186660
      Partition f: starting cyl=120, # blocks=279990
      Partition g: starting cyl=669, # blocks=279990
      Partition h: starting cyl=120, # blocks=559980
OK to use this partition table? y
Use the label command to write out the partition table.
diag> label
label this disk...
OK to use logical partition map 'CMU Priam 638'? y
Are you sure you want to write? y
verify label
id: <Priam 638 cyl 1218 alt 6 hd 15 sec 34>
      Partition a: starting cyl=0, # blocks=28560
      Partition b: starting cyl=56, # blocks=32640
      Partition c: starting cyl=0, # blocks=621180
      Partition d: starting cyl=120, # blocks=932330
      Partition e: starting cyl=303, # blocks=186660
      Partition f: starting cyl=120, # blocks=279990
      Partition g: starting cyl=669, # blocks=279990
      Partition h: starting cyl=120, # blocks=559980
diag> q
>b sd(0,0,1)vmunix -as                          bring mini-unix back up

[... messages from mini-unix ...]

# newfs -n /dev/sd0a 638                        Make some file systems
[... messages from newfs ...]
# newfs -n /dev/sd0d 638
[... messages from newfs ...]
# newfs -n /dev/sd0b 638
[... messages from newfs ...]
# newfs -n /dev/sd0e 638
[... messages from newfs ...]
# newfs -n /dev/sd0g 638
# mkdir /mnt
# mount /dev/sd0a /mnt
```

# **cd /mnt**	
# **restore rbf 1024 /dev/nrst8**	*Pull in the root partition*
# **mkdir /sys0**	
# **mount /dev/sd0d /sys0**	
# **cd /sys0**	
# **tar xbf 1024 /dev/nrst8**	*Now for /sys0*
# **mkdir /usr0**	
# **mount /dev/sd0e /usr0**	
# **cd /usr0**	
# **tar xbf 1024 /dev/nrst8**	*And now /usr0*
# **tar xbf 1024 /dev/nrst8**	*source files*
# **cd /usr0**	
# **tar xbf 1024 /dev/nrst8**	*miscellaneous programs*
# **cat > /mnt/etc/fstab**	*Put an fstab on the new disk*
/dev/sd0a:/:rw:1:1	
/dev/sd0d:/sys0:rw:1:2	
/dev/sd0b:/tmp:rw:0:5	
/dev/sd0e:/usr0:rw:1:3	
/dev/sd0g:/usr1:rw:1:4	
^D	
# **/usr/mdec/installboot bootsd /dev/sd0a**	
# **sync**	
# **sync**	
Abort at *some-number*	*Halt with L1 a*
> **b sd(0,0,0)vmunix -s**	*Boot mach single user*
[... lots of output from the mach kernel ...]	
-OPR- **fsck -p /dev/sd0a**	
[... fisk output ...]	
-OPR- **su**	
Password:	
# **rm /etc/THISHOST**	*Set up the host's name*
# **cat > /etc/THISHOST**	
ZION.MACH.CS.CMU.EDU	*Use YOUR host name here*
^D	
# **rm /etc/NETWORKS**	*Set up the IP address*
# **cat > /etc/NETWORKS**	
128.2.222.192 on ie0	*Use YOUR IP address here*
^D	
# **fsck -p /dev/rsd0d**	*Need to mount sys0 since /etc/hosts is*
# **mount /dev/sd0d /sys0**	*linked to /usr/cs/lib/hosts/hosts*
# **cat >> /etc/hosts**	*Add ourself to the host table*
128.2.222.192 zion.mach.cs.cmu.edu zion	*Use YOUR information here*
^D	
# **^D** -OPR- **^D**	

You should not bring the system up for users unless you know
that a "fsck -p" has been run and has produced no errors.

Do you wish to run a "fsck -p"? [yes]	*Just hit return*

7. Post-Installation Considerations

7.1. How to Make Accounts

1. Edit /etc/{passwd,group}

   If you don't know what the passwd and group files are, you should look at their manual pages in the "Unix Programmer's Manual".

2. Edit /etc/account

   /etc/account is a CMU-CS specific file that tells when your account starts and expires for each group that you are in. As an example, here are a few lines from it:

   ```
   jjc::477:3:01-Dec-86:01-Jan-89:root:
   jjc::477:7:03-May-86:01-Jan-89:root:
   jjc::477:377:17-May-87:01-Jan-89:root:
   jjc::477:408:01-Sep-87:01-Jan-89:root:
   ```

3. Make a home directory for the account

4. Make a symbolic link from /usr to the home directory

5. Change the ownership of the home directory to the user and his/her group.

Each line of /etc/account contains the account name, user ID, group ID, starting date, expiration date, and sponsor.

7.2. Nanny

/usr/cs/etc/nanny is a server started in /etc/rc that starts and babysits other servers. It will restart any of the servers that it babysits if they die. The servers that it babysits are listed in /usr/cs/etc/nanny.config. The description of each server in nanny's configuration file consists of the full path name of the server, what user and group IDs to run it as, the value to nice it at, and the attribute that the machine must have for nanny to run it. Nanny looks for the machine's attributes in /etc/attributes.

To add or delete a server from nanny's babysitting service, add or delete the server description to or from /usr/cs/etc/nanny.config. If you add a server, check to make sure that your machine has the attribute needed to run the server in /etc/attributes. If you don't want to run nanny, comment out the invokation of it in /etc/rc and start the servers in /etc/rc or /etc/rc.local instead.

See nanny(8) for more information.

7.3. Netmsgserver

/etc/nmbroadcast needs to be edited to contain one line containing an IP address to be used as the IP broadcast address. Omission of this file will cause Mach's netmsgserver to use CMU's IP broadcast address as a default.

7.4. Root, Super-Root and RFS

To clarify upfront what may be the cause of some confusion regarding the software system you have just installed, as distributed your system includes 3 root directories. First, there is the physical root which is /../../. The second is the remote root which is /../ and the third is the local root which is / and appears as a typical Unix root.

The reason for this is that the kernel as distributed supports a remote file system (i.e. RFS) first developed at CMU in 1982. Mach systems at CMU are usually "rooted" at /RFS/.LOCALROOT, allowing for a /../**hostname** syntax and a /../../ super-root.

RFS is easy to use. The syntax is /../**hostname or alias**. For example, the following command would list the contents of /tmp on wb1:

```
ls /../wb1/tmp
```

The local machine must have an RFS link to the remote machine for you to access the remote machine through RFS. To make a link in /.. for a particular machine:

1. Cd to /..
2. Run /usr/cs/etc/remhost giving it the alias of the machine.

   NOTE: The host name and alias must appear in /etc/hosts with its corresponding IP address for this to work.

The above procedure will create two binary files, /../**alias** and /../.**host name**, which contain the IP address of the machine. To refer to these files without going across RFS, you must quote the file by using the syntax /..///. For example, the following command would show the contents of /../**alias** in hexadecimal:

```
od -x /..///alias
```

For RFS to work, the remote machine must be running the RFS daemon, /usr/cs/etc/rfsd, which is started by nanny. The remote machine must also list the IP address of any machines which are allowed to access it in the file **/usr/cs/lib/hosts/local-hosts**.

By default, RFS accesses files as if you had the user and group, rfsd and system. To authenticate yourself to RFS as someone different, you should use the program /usr/cs/etc/rfs. It will prompt you for a user, group, group account, and password. Give the default answer when it asks "Identify?". After running /usr/cs/etc/rfs, you will have a shell that will allow you to access files on remote RFS hosts that you have a corresponding user account, group, and password for.

7.5. Paging

The kernel pages out to inodes. There is no need for special dedicated paging area, but there must be free space on the disk to page out to. The program /usr/mach/etc/swapon can be run at boot time to specify partitions to perfer to swap to and those to never swap to. (see mach_swapon.8)

7.6. System Log Files

System log files are located under /usr/adm. Should any unusual system behavior be noticed, looking at the log files located here is a good place to start looking for clues as to what the cause may be. An unusually large log file can be an indication that system errors are occurring. It's a good idea to occassionally check the size of this directory by simply doing a du /usr/adm. Also, these log files should be periodically checked as some have a tendency to become excessively large over time. Most are regularly removed when they exceed a specified age, see /usr/lib/crontab.

7.7. Changing Time Zones

As distributed the time zone is set to eastern standard time, i.e. EST5EDT. If you need to change time zones you will need to perform two tasks. First, you will need to re-link */etc/zoneinfo/localtime* as follows:

```
login as root
cd  /etc/zoneinfo
rm  localtime
ln  xxx  localtime   (substitute appropriate timezone file for xxx)
```

Second, you will need to patch time zone variable (i.e. _tz) in the kernel. Prior to doing this you will need to calculate the appropriate patch value in hex of the difference in minutes between your local zone time and GMT (universal time). The value of this constant for eastern standard time, for example is x12c hex (i.e. 300 decimal). This is the value that you will initially see for _tz. To change to Pacific Daylight Time, as an example, _tz should be patched to x1e0 hex (i.e. 480 decimal). The following example illustrates patching the vmunix kernel to PDT.

```
login as root
/bin/adb  -w /vmunix
   tz?W  1e0
   $q
/etc/reboot
```

An alternative to patching the kernel is to configure a new kernel with the right timezone, recompile it, and install it.

8. How to make changes and upgrades

8.1. Compilation Environment

Building Mach software (including the kernel) requires a modified compilation environment. This environment includes a path searching version of the C compiler. The C compiler resides in /usr/cs/bin/cc and requires the following paths (which are environment variables):

CPATH	Path to search for include files.
LPATH	Path to search for library files.

These paths usually contain at least:

CPATH	:/usr/mach/include:/usr/cs/include:/usr/include:
LPATH	:/usr/mach/lib:/usr/cs/lib:/lib:/usr/lib:

A script that sets these environment variables to reasonable values is in /usr/mach/lib/machpaths.

The Makefiles that are supplied with the sources in /usr/build/src assume that the program /usr/misc/.sdm/bin/build is being used to execute them and need the CMU-modified version of make found in /usr/cs/bin/make. The files Makefile-common, mach/Make.inc and {.,mach,kernel}/Makeconf are also required parts of the make process. Makefile-common is a set of generalized make rules that is included by the Makefiles in the bin, etc and usr subdirectories. Mach/Make.inc is the set of general rules included by most of the Makefiles in the mach subdirectory. The respective Makeconf files are also read by make and specify an object base directory into which the .o files will be placed. The mach/Makeconf and kernel/Makeconf files also define a few other parameters that the make process depends on.

8.2. Rebuilding from the sources

The build program (see /usr/misc/man/man8/build.8) reads a project description file from which it gets information about switches to make, where to put the executables, and lists of directories to be used for the CPATH and LPATH variables. It will set the correct switches, paths and targets and then call the make program. If you wish to be sure that you are getting the correct make program, set the environment variable **MAKE** to the complete pathname. The project description files are in /usr/build/lib/project.

To build programs in bin,etc, or usr

```
setpath LPATH -i99 /usr/build/lib
setenv PROJECT sys.2.5
cd /usr/build/src
#build <program> but do not install it
build /usr/cs/bin/<program>

#build ps and install it in /usr/bin
build -fromsource -todefault /bin/ps

#build and install all the programs in bin and its subdirectories.
build -fromsource -todefault -here /bin
```

To build programs in mach

```
setpath LPATH -i99 /usr/build/lib
setenv PROJECT sys.2.5
cd /usr/build/src/mach
#build md but do not install it
build /bin/md

#build md and install it in /usr/mach/bin
build /bin/md.install

#build and install all the programs in /lib and it
#subdirectories libthreads and libmach
build /lib/install
```

In the mach area, build is only used to set the path values correctly. If you have CPATH and LPATH correctly set you can use make instead of build.

To build a kernel

```
cd /usr/build/src/kernel
make   - builds the kernel specified by CONFIG in kernel/Makeconf
make CONFIG=STD+MP-afs - builds the specified configuration.
```

Build is only useful with the kernel sources if you wish to make a shadow area in which you place a few modified sources and get the rest of the sources from the standard source area. Build can deal with sources in multiple directories where make cannot. See /usr/misc/man8/build.8 for details on how to do this.

8.3. Changing Kernels

If your license allows it, you will have received extra versions of the kernel in the /usr/build/obj/kernel directory. These kernels are in the subdirectories STD+WS and STD+ANY. The versions STD+ANY or STD+WS have both NFS and AFS support. The versions STD+ANY-afs and STD+WS-afs have NFS support. The versions -afs-nfs have neither. The ANY configurations support more drivers. The vax MP configurations are for multi-processors vaxes with BI buses. The conf/MASTER.* files in the kernel source area define the contents of the various configurations.

The kernel must be in two roots, the physical root (/../..) where the boot program is expecting to find it and the local root (/) where the kernel is found when the system is up. To make the kernel be in both roots, a hard link is made between /vmunix and /../../vmunix. Even though this link has to be made, booting a new kernel is very straightforward:

1. Su to root
2. Copy your new kernel into /vmunix.reboot
3. Change its protection to 640
4. Change its user and group ownership to sys and kmem
5. Reboot

There are lines in /etc/rc that save /vmunix in /vmunix.old, move /vmunix.reboot to /vmunix, make the necessary links from the local to the physical root, and then reboot.

Two shell scripts, /etc/oldvmunix and /etc/newvmunix, are useful for changing kernels. /etc/oldvmunix will save /vmunix in /vmunix.new, move /vmunix.old to /vmunix, and change the links. /etc/newvmunix will do the opposite, moving /vmunix to /vmunix.old and /vmunix.new to /vmunix.

8.4. Supping updates from CMU

Both the sources and the binary for sup (Software Update Protocal) are included in this distribution. With the installation of some control files at your site and at CMU, this program can be used to get updated versions of the kernel or mach sources, and the BSD binaries and selected sources. It can also be used at your site to automatically keep software consistent on all your Mach workstations.

The sup program is documented both in /usr/mach/doc/sup.ps and in /usr/cs/man/man8/sup.8. Send mail to mach@cs.cmu.edu to request more information about setting your site up to sup upgrades from CMU. The machine that you wish to sup files to must be on the INTERNET.

9. Differences between Mach 2.0 and Mach 2.5

9.1. NFS and in-kernel AFS support

If you have the appropriate NFS and AFS licenses, your distribution includes kernel support for Sun's NFS and possibily the Andrew File System (AFS) as well. The few user level programs need to run NFS and AFS are found in /usr/misc/.nfs. The programs to set up an AFS server or to maintain a complete NFS are not provided. It is assumed that you can get them from Sun and Transarc respectively. The necessary daemons are started by /etc/rc.

9.2. Mach IPC changes

The Mach 2.5 kernel has two compile-time options (MACH_IPC_SIGHACK, MACH_IPC_XXXHACK) that can be used to enable backwards-compatibility code. The kernel sources currently have these options enabled but once you change your sources, you may want to turn them off.

In general, Mach 2.5 is not binary compatible with Mach 2.0, because traps and kernel calls have changed, but with some help from libmach it is reasonably source compatible.

1. Ownership rights are no longer supported. MSG_TYPE_PORT_OWNERSHIP and MSG_TYPE_PORT_RECEIVE are invalid. ownership-rights and receive-rights notifications are never generated. (MACH_IPC_XXXHACK enables ownership rights.)
2. Emergency messages are no longer supported. A port has only one queue of messages.
3. PORT_ENABLED is no longer supported. Instead, Mach 2.5 provides "port sets", which generalize the enabled/disabled concept. (MACH_IPC_XXXHACK enables support for PORT_ENABLED. Note that libmach also has some backward-compatibility support for PORT_ENABLED, port_enable, and port_disable.)
4. Mach 2.0 allowed a task to receive messages directly from an enabled port, in addition to receiving messages directed to enabled ports by using PORT_ENABLED. Mach 2.5 does not allow messages to be received directly from ports that are members of a port set.
5. SIGMSG and SIGEMSG are no longer supported. Mach 2.0 signalled the receiver when a message was sent to an enabled port. Mach 2.5 IPC never generates signals. (MACH_IPC_SIGHACK enables SIGMSG/SIGEMSG support. CMU Common Lisp relies on these signals.)
6. The IPC traps (msg_send/msg_receive/msg_rpc) have different numbers and arguments in Mach 2.5. (However, note that the wrapper functions in libmach continue to have the same arguments as in Mach 2.0, so this change is transparent to user code and only impacts binary compatibility.) (MACH_IPC_XXXHACK enables the old traps.)
7. Mach 2.5 has a completely different set of port-manipulating kernel calls. The old port_allocate, port_deallocate, port_enable, port_disable_select, port_select, port_set_backlog, and port_status are gone. task_ports is gone, but it wasn't implemented in Mach 2.0. (MACH_IPC_XXXHACK enables these old kernel calls, except for port_select and task_ports.)
8. Mach 2.5 has new versions of port_allocate, port_deallocate, port_set_backlog, and port_status that have basically the same arguments. The subtle difference is that they refer to ports by name instead of passing send rights. port_status also returns a port-set-membership indication in place of the enabled/disabled indication. It still returns separate ownership/receive rights indications, also they will never have different values (unless MACH_IPC_XXXHACK).
9. Mach 2.5 has new kernel calls: port_names, port_type, port_rename, port_set_allocate,

port_set_deallocate, port_set_add, port_set_remove, port_set_status, port_insert_send, port_extract_send, port_insert_receive, port_extract_receive, port_set_backup.

10. Mach 2.5 supports the use of TASK_KERNEL_PORT with task_{get,set}_special_port, and THREAD_KERNEL_PORT with thread_{get,set}_special_port.
11. Mach 2.5 has a "backup port" concept to replace the most common use of ownership rights. Using port_set_backup, a task can specify a backup port for a port that it owns. Normally when receive rights for a port are deallocated, the port is destroyed. If the port has a backup, instead they are sent in a port-destroyed notification to the backup port.
12. All notifications in Mach 2.5 have a PORT_NULL msg_remote_port field. Notifications in Mach 2.0 carried send rights for the task's kernel port in the msg_remote_port (reply port) field.

9.3. External Pager interface changes

In Mach 2.0 the External Pager was only provided in +EXP configurations of the kernel. It is now standard and cannot be disabled. The names and parameters of most of the calls have changed. The old names began with **pager_** the corresponding new ones begin with **memory_object**. For example,

```
pager_data_provided(pageing_object_request,
        offset,data,data_count,lock_value)
```

has been replaced with

```
memory_object_data_provided(memory_control,
        offset,data,data_count,lock_value)
```

and the type of the first parameter changed from **port_t** to **memory_object_control_t**.

The new calls are documented in the Kernel Reference Manual included with this distribution.

9.4. Kernel include files reorganized

The kernel include files were reorganized in the following ways:

- sys now includes only files from Berkeley. Mach files were moved from here to mach, kernel and vm directories.
- mach and mach/<machine> contain all the Mach specific include files that are to be exported to /usr/cs/include. Files were moved here from sys, kern and vm.
- mach_debug and mach_debug/<machine> contain the debugging interface to the kernel. This interface is only included in +EXP configurations.
- include files in kern and vm directories are no longer exported to /usr/cs/include.
- The kernel include files are no longer placed in /usr/mach/include but only in /usr/cs/include with symbolic links to there from /usr/include.

The sed script in /usr/build/src/mach/FIX-INCLUDES.sed can be used to modify your source files to include the header files from the new locations.

10. Known Problems/Bugs

11. Mach Feedback

We expect to get three kinds of feedback:

1. Bug reports/fixes and suggestions for improvements
2. Code
3. General enquiries

11.1. Reporting bugs/ General enquiries

The Mach Project may be contacted either by US Mail, Arpanet mail or telephone.

US Mail Address

Mach Project
c/o Richard F. Rashid
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA

Arpanet mail addresses

mach@wb1.cs.cmu.edu
info-mach-request@cs.cmu.edu
info-mach@cs.cmu.edu

mach is read by the Project Manager and the Distribution Coordinator. Problems pertaining to distribution problems should be sent here.

info-mach-request is used to accept requests to be added to the info-mach mailing list.

info-mach is a mail distribution list consisting of the mach developers and mach users. Bug reports, technical questions and suggestions on how to use mach should be sent to this list. Also replies to questions raised here should be cc'ed to this list.

Telephone

412 268-8166 (Lori Iannamico - Distribution Coordinator)

The telephone number should only be used when absolutely necessary and then only to leave a message. Please keep in mind that we do not have a support staff and so cannot provide more than minimal help with problems.

11.2. Returning code to CMU

We encourage you to send back to us any code which may improve the system. Please follow the conventions we have used in the kernel. In particular, you should make your changes under specific compilation switches and you should add the switches to the configuration file. All changes (even bug fixes) should leave the original code intact such that turning off the appropriate option will return the system to its original state. By following these conventions we will be able to more quickly incorporate your modifications. This is especially important if you are porting Mach to a new machine and want CMU to include that software in a future release.

Code may be returned in a tape (similar to the kind of tape you received with this release) or via the

Arpanet. If you have network access and intend to work heavily with Mach it may be desirable to access Mach software using a software update program (called SUP) which is supplied with the release tape. This program allows network updating of software modifications between sites. If you feel this is important to you or your group, you should contact the Mach group as described above to make the necessary arrangements.

12. References

- 4.3BSD Unix Programmer's Manual
- Installing UNIX on the Sun Workstation

- Unix is a trademark of AT&T Bell Laboratories

Table of Contents

List of Figures