2 Installation Notes

2.1 Introduction

This chapter describes the procedures for installing and configuring the MK7.3 binary distribution and compiling the source code on the supported reference platform. The distribution tape only includes the source code needed to build the MK7.3 microkernel and OSF/1 server, together with supporting MK-related software. This software must be built on a machine that is already running MK7.1 or later. Once you have installed the binary portion of the MK7.3 release, you can then use the same machine to compile the sources.

Updated information about the release can be found on the World Wide Web by following the “MK7.3 Release Additional Information” link on the MK page (accessible from the RI home page). Refer to this page for additional MK7.3 information that may have been added after these release notes were prepared. The complete URL is:

http://www.opengroup.org/RI/PubProjPgs/MK73.htm

2.2 Reference Platform Definition

The following list constitutes the supported hardware configurations for MK systems release 7.0 and later (as well as AD systems release 3.0

and later). There are significant differences for earlier MK and AD systems. The software described by these release notes is designed for (and has been tested on) these configurations.

There are required minimum configurations, recommended minimum configurations, and obsolete configurations. The obsolete configurations are believed to work but have not necessarily been tested recently. If an obsolete configuration does not work, we may decide not to fix the problem.

2.2.1 System

PC/AT compatible with i486 or Pentium processor and ISA/EISA/PCI bus. Hardware floating point is required (486DX for 486 machines).

Obsolete: PC/AT machines based on the i386.

2.2.2 Video/Monitor

Required: VGA display in 25 x 80 character mode

We use color monitors due to historical problems with monochrome monitors.

2.2.3 Memory

MK 7: 16 MB required, 32+ MB required for use with X11, and otherwise recommended. (We only regularly test on machines with 32 MB or more.)

AD 3: 32 MB required, 36+ MB recommended for server development.

2.2.4 Disk Controllers/Host Adapters

The following hardware configurations are supported (only one controller/adapter at a time, however):

SCSI Disk with Adaptec AHA-1542B/1542C adapter (ISA bus)

SCSI Disk with Adaptec AHA-1542CF adapter (ISA bus) (MK 7.0a+)

SCSI Disk with Adaptec AHA-1542CP adapter (ISA bus) (MK 7.2+)

SCSI Disk with Adaptec AHA-1742 adapter (EISA bus)

SCSI Disk with NCR/Symbios 53C810 or 53C825 PCI adapters (825 in narrow SCSI only). Note that 810 adapter boards require the BIOS to be present on the motherboard. (MK 7.2+) (AD 3.0+)

Obsolete:

ESDI Disk with Western Digital 1007 controller

Note: Some IDE drives may work with WD1007-compatible IDE disk controllers.

We strongly suggest SCSI disks in all configurations. That is what we use (almost exclusively) for our own workstations.

SCSI disks are required for use in real-time configurations.

2.2.5 Disk

Currently only one disk may be used at a time.

Required:

300 MB disks for minimal test machines.

600 MB disks on program development machines.

Recommended:

1 GB disks.

2.2.6 Tapes

SCSI DDS-1 (4mm DAT) tape drives. (MK 7.2+ only)

SCSI QIC-150 is known *not* to work on MK 7.2.

2.2.7 Ethernet

Required: The following 10Mbps Ethernet boards are supported:

SMC WD8003/8013 EP, Elite, Elite 16, Ethercard plus family (e.g., SMC 8013EWC, which supports thinnet, thicknet and UTP or SMC 8013EPC which supports thinnet and thicknet).

DEC Etherworks PCI Ethernet card (DE435-AA/DE450). (MK 7.2+ only) (AD 3.1+)

SMC 8216C (EliteUltra) seems to work in a compatibility mode. (MK 7.2+) (AD 3.1+)

Obsolete:

3Com Etherlink II cards (3C503) may work, but cannot be used for initial network loading.

2.2.8 FDDI

Required only for use with FDDI. FDDI does *not* eliminate the requirement for Ethernet. The following FDDI board is supported:

Digital PCI FDDI Single Attach (DEFPA-AA) (MK 7.0a+) (AD 3.1+)

2.2.9 Myrinet

Required only for use with Myrinet. Myrinet does *not* eliminate the requirement for Ethernet. The following Myrinet board is supported:

> Myricom M2-PCI32 and M2F-PCI32 PCI boards. (MK 7.2+ only)

2.2.10 Floppy

One 3.5” (1.44 MB) drive is required as Drive A:.

One 5.25” (1.2 MB) drive is supported as Drive B:.

2.2.11 Graphics Controllers

The X11 window system is supported only on MK systems.

MK uses the XFree86 distribution for X11. The following graphics controllers are supported:

> #9 GXE (currently used in many RI workstations) (make sure you get revision level 12 or higher)
>
> Diamond Stealth PCI 64 bit graphics card, with at least 2 MB of video RAM, S3 Vision-968 controller.

Obsolete:

> Paradise P1A VGA (not been tested recently)
>
> Sigma Legend
>
> Sigma Legend II
>
> Tseng Labs ET4000-based boards

2.2.12 High resolution Monitors

We don’t know that there should be particular issues with the graphics monitors — as long as the monitor is compatible with the graphics controller that you use. The RI happens to use Mitsubishi Diamond Scan 20, NEC Multisync 6FG, and NEC XP21 color monitors.

2.2.13 “In-kernel” CORDS: (MK 7.2+ only)

Only the following LAN cards are supported for use with In-kernel CORDS:

> SMC WD8003/8013 EP, Elite, Elite 16, Ethercard plus family.
>
> SMC 8216C (EliteUltra).
>
> Digital PCI FDDI Single Attach (DEFPA-AA).
>
> Myricom M2-PCI32 and M2F-PCI32 PCI boards.

2.2.14 Miscellaneous

2 RS232 serial ports (COM1 and COM2)

1 parallel port (unidirectional) (MK 7.0a+ only)

Soundblaster (MK 7.2+ only)

3-button bus mouse, using the Logitech Protocol (MK 7.0+ only)

MK is being used at The Open Group Research Institute on the following models of x86 PC/AT compatible systems:

- Digital Celebris XL 5100 — 100 MHz Pentium with PCI/ISA bus, 32+ MB RAM, Diamond Stealth graphics card, and Symbios 53c810 SCSI chip on motherboard.
- Digital DECpc 450ST — 50 MHz i486 with EISA bus, 36+ MB RAM, #9 GXE graphics card, and Adaptec 1742 SCSI adapter.

Most PC/AT (386+) class machines can be expected to work provided the peripherals are in the list of supported equipment presented above.

2.3 MK 7.3 Distribution Tape

The MK7.3 distribution tape contains two *dd*(1) volumes and four *tar*(1) volumes, as shown in Table 1 below.

Volume	Description	Size
1	MK7.3 binary image (root)	12 MB
2	MK7.3 binary image (usr filesystem)	37 MB
3	MK7.3 boot floppy disk images	5 MB
4	MK7.3 documentation	3 MB
5	MK7.3 sources + build logs	39 MB
6	MK7.3 build tools	45 MB

Table 1: Disk Space Requirements

The first three volumes of the tape contain the *MK7.3 binary release*, while the remaining volumes comprise the source distribution.

The first and second volumes of the tape contain compressed *tar* images of the *root* and *usr* filesystems, which were written onto the distribution tape by *dd*. These two binary volumes contain a basic MK7.3 environment with standard commands and libraries. The third volume contains *tar* images of the floppy disks that are used to bootstrap MK7.3 onto a new system.

The *MK7.3 source distribution* consists of the last three volumes of the tape, each of which is in *tar*-format. The fourth volume contains documentation, including these release notes and manual pages. The fifth volume contains the source files for MK7.3 and reference build log files. The last volume provides the (binary) tools needed to build the release.

Be sure that you have enough disk space before attempting to extract the contents. Table 1 shows the size of each volume. If you are only interested in installing the MK7.3 binary distribution, follow the instructions provided in Section 2.4. If you wish to install the source code, build the system, and install the resultant binaries, follow the instructions provided in Section 2.5.

2.4 Binary Installation

The first and second volumes of the tape contain compressed *tar* images of the *root* and *usr* filesystems, which were written onto the distribution tape by *dd*. These two volumes contain the installed image of the basic MK7.3 binary environment. The third volume contains *tar* images of three floppy disks that are used to boot an MK7.3 system. After extracting the floppy images onto your system, you will need to copy them onto blank double-sided high-density 3.5-inch floppy disks (1.44 MB capacity). On a UNIX system, you can use the *dd* command to make a bit-for-bit copy of the file onto the floppy disk device.

The third volume also includes a shareware DOS utility (rawrite) that can be used on any PC to write a file onto a floppy disk. If you are running MK7.X on a system, you can use the *mkbootfloppies* script to create bootable floppy disks for MK7.3. The script can be easily adapted for use on other UNIX systems.

The binary distribution can be installed using either the network or a local SCSI DDS-1 (4mm DAT) tape drive. Throughout this chapter, the machine on which you are going to install MK7.3 is referred to as the *target* machine. If you intend to use the network for installation on the *target* machine, you will need to use a *remote* machine which has either the remote tape drive from which you will read the MK7.3 distribution tape, or the first two volumes of this tape previously unloaded onto a local disk. You will need an account on this *remote* machine with access to the tape drive, or to the location on the local disk where the contents of the tape reside. Moreover you should enable remote access for this account for the *root* user from the *target* machine through *rsh* by modifying the *.rhosts* file in the HOME directory of this account.

Section 2.4.1 describes the steps that may be necessary to prepare a new system for a binary installation. It is not necessary to reformat the hard drive if you are upgrading an existing MK7.X system. Section 2.4.2 steps through the installation process once the preparations have been completed.

2.4.1 Preparation for Binary Installation

In order to use a SCSI drive as the system disk for MK7.3, you must set it as SCSI unit 0. Similarly if you intend to use a local DDS-1 tape drive to install the binary distribution, set the tape drive to SCSI unit 4.

2.4.1.1 Preparing the Target Machine for Network Access

You will need to choose a host name for your *target* machine along with an IP (Internet Protocol) network address. (Please consult with your network administrator). You will also need to know the *netmask* for your local network. Before installing an ISA bus network adapter in your target machine, you should configure the I/O port address, interrupt level and the RAM base address (if applicable) so the network interface can be recognized by the microkernel. See the Ethernet Board Configuration description in Section 2.7.1 for detailed information on how to properly configure your ISA network adapter.

2.4.1.2 Preparing Remote Machine Access

NOTE: Skip this section if you are installing MK7.3 using a local SCSI tape drive.

The *remote* machine is the one from which you will read the MK7.3 tape; so you must know the name of the host and its IP address. We require that the *target* and *remote* machines be connected to the same subnet. On the *remote* machine, you should add an entry in the */etc/hosts* file containing the IP address and hostname of the *target* machine. For instance, if the hostname of the *target* machine is *hal* with the IP address 123.45.67.89, you should have the following entry in the */etc/hosts* file on the *remote* machine:

```
123.45.67.89    hal
```

Similarly, you will need to have an account on the *remote* machine that you can use for access to the SCSI tape drive (or remote disk storage with the tape contents) located on the *remote* machine. In the HOME directory of this account, you should create a *.rhosts* file granting access to the *root* user from the *target* machine. If the hostname of the *target* machine is *hal*, you should have the following entry in the *.rhosts* file of the HOME directory for the account you will use. This user must have read access to the remote tape drive (or remote disk storage with the tape contents) on the *remote* machine:

hal root

You should note the raw device name of the SCSI tape drive (specifically the non-rewinding tape device) or the pathnames for the compressed *root* and *usr* file systems extracted from the tape. Depending upon the type of the *remote* machine, the tape device may have different names, such as:

/dev/nrmt0h, /dev/nrct0, /dev/rmt/0hn, /dev/nrst4, ...

2.4.1.3 Disk Formatting

You may need to perform a low-level “surface format” of the hard disk on your *target* machine. This procedure partitions the disk into sectors and then checks for bad areas of the disk. This process can require up to an hour to complete, depending upon the size of the disk and speed of the PC. Most disk manufacturers (HP, IBM, Seagate, Quantum, etc.) ship hard drives that are preformatted. Also, some vendors will format the disk for you prior to shipment. The documentation you received with your disk, computer, or DOS will explain how to format your disk. The procedures vary widely and are beyond the scope of this document. Please contact the disk drive manufacturer if you have low-level formatting problems.

Note that almost all SCSI disk drives are already preformatted. You do not need to reformat a disk when upgrading an existing MK7.X system.

2.4.2 Binary Installation Process

There are five steps necessary to install the binary MK7.3 distribution, each of which is described below. First you must create a set of bootstrapping floppies and use them to boot the computer (Section 2.4.2.1). Next you need to prepare the hard disk drive for the binary installation (Section 2.4.2.2). If you are installing the release via the network, you will also need to initialize the network adapter (Section 2.4.2.3). Next you can extract the binary images onto the hard disk (Section 2.4.2.4). Finally, you will need to customize certain system configuration files (Section 2.4.2.5).

Installing MK7.3 is intended to be a simple, cookbook process.

In the following paragraphs, all arguments enclosed within square brackets indicate optional parameters.

2.4.2.1 Booting the Target Machine from Floppy Disks

We will assume for the discussion below that you are performing a "simple" installation. In particular, we assume that you do not want to reserve a portion of the disk for other operating systems (such as DOS or Linux), that you do not have excessive bad blocks in inopportune locations (where the system would put the bad block table), and that you do not need to override the disk geometry parameters that are maintained internal to BIOS.

You will use the boot floppies that you created earlier (by copying the contents of the third volume of the MK7.3 distribution tape) to bring up the system in "single user" mode. (See the discussion in Section 2.4 above concerning the creation of these floppy disks.) Depending on the BIOS of your machine, loading from these floppies may be rather slow.

Insert the first floppy and reboot (or power toggle) the machine. The system should print out a few lines — indicating the amount of memory and the bootstrap version — and then print

```
boot:
```

At this prompt, press the "Enter" (or "Return") key. Then, you will see a number printed out, a pause, a "+", etc. While the bootstrap is reading from the floppy, it prints sizes for this microkernel (text size, data size, bss size, symbols size). Next, you will see several lines of printout as the MK7.3 microkernel describes the hardware configuration that it finds.

Shortly after, the bootstrap will prompt:

```
configuration file: /dev/boot_device/mach_servers/boot-
strap.conf
```

Press the "Enter" key after this and the following prompt will appear:

```
default_pager: /dev/boot_device/mach_servers/default_pager
```

Press the "Enter" key again and the next prompt will appear:

```
startup: /dev/boot_device/mach_servers/startup -s -a fd0b
```

At this point, remove the first floppy and insert the second boot floppy. Once you have switched floppies, you can proceed with the boot process by pressing the “Enter” (or “Return”) key. There will be several lines of output as the server starts. Then you will see the following prompt:

```
root device: [fd0b]
```

Remove the second floppy, replace it with the third floppy disk and press the “Enter” key. Finally, you should see the “#” prompt, indicating that the server is running in “single user” mode with root privileges.

If nothing is displayed after the “`boot:`” prompt then there may be a display/graphics adapter problem (e.g. the board may have switched to an SVGA mode depending upon your monitor). In this case, you may need to switch to a standard VGA graphics adapter board to avoid the problem, unless your display adapter board can be configured to run in standard VGA mode.

2.4.2.2 Disk Setup

This step prepares the hard disk on the *target* machine. This is accomplished by executing the *disk_setup* command on the target machine:

```
# disk_setup [ide]
```

If you are using a standard SCSI disk, you only need to type:

```
# disk_setup
```

By default, *disk_setup* assumes that you have a SCSI disk attached to an Adaptec or NCR host adapter from which you wish to boot.

disk_setup is an interactive script that initializes the disk for use under MK7.3. It can optionally perform three disk initialization steps:

- Low-level formatting, which will destroy any/all data on the disk. Using IDE, EIDE or ESDI disks, this procedure takes roughly 15 minutes per 100MB. On a Pentium system using a SCSI disk drive, formatting the disk requires approximately 15 minutes per gigabyte.
- Partitioning the disk (using the *vtoc* utility). You will be asked to specify parameters for each disk partition interactively. Refer to Section 2.7.4 for an example.
- Creating new filesystems with *newfs*. This is required if you reformatted or re-partitioned the disk.

If an earlier MK7.X system was previously installed, it is not necessary to perform any of the three preceding steps when upgrading to MK7.3. (If you choose to, you will lose any information that is currently stored on the disk.) Nonetheless, you should still run *disk_setup* to extract the tools needed to complete the binary installation. If you are upgrading an MK7.X system and wish to preserve disk data, you can answer “no” to each of the disk initialization options described above.

Finally, after initializing the disk as appropriate, *disk_setup* extracts a set of tools from the third boot floppy that will be used to complete the binary installation. (The boot floppies were described earlier in Section 2.4.)

2.4.2.3 Network Setup

NOTE: **Skip to Section 2.4.2.4 if you are installing MK7.3 using a local SCSI tape drive.**

This step defines the network configuration on the *target* machine by means of the *net_start* command:

```
# net_start targ_ip_addr device_name netmask
```

where the *net_start* parameters are defined as:

> `targ_ip_addr` — the Internet address of the *target* machine.
>
> `device_name` — the Ethernet adapter on the *target* machine. The name is *wd0* for the SMC WD8013EP family of ISA boards (nee Western Digital) or *tu0* for the DEC DE-450 “TULIP” PCI Ethernet adapter.
>
> `netmask` — the network mask for the LAN to which the *target* machine is connected (cf IP address of your *target* machine).

For example, assuming your *target* machine IP address is 123.45.67.89, its Ethernet board is an SMC WD8013EP (or SMC Elite8), and the netmask of your class C network is 255.255.255.0, you would type:

```
# net_start 123.45.67.89 wd0 255.255.255.0
```

Note that the length of the netmask field will vary according to the type of network number assigned to your LAN (class A, B or C IP addresses).

2.4.2.4 Download the Binary Images

After having rebooted your machine to single user mode, you should perform the final step of the installation process by issuing the *mk_install* command. This command is used for all binary installations (both local tape and network installations).

If the installation procedure fails during this step (software downloading), you should correct the problem and rerun the *mk_install* script.

Eventually, even if you need to reboot your *target* machine, *mk_install* will be the only step you will need in order to complete the installation process after rebooting with the floppies (unless you provided incorrect parameters when running *net_start* in the previous step, described in Section 2.4.2.3).

Network installations

If you are using a *remote* machine to supply the distribution tape files to be installed on the *target* machine, you should use the following form of the *mk_install* command:

```
# mk_install rem_ip_addr rem_account rem_device
```

where the *mk_install* parameters are defined as:

> `rem_ip_addr` — the Internet address of the *remote* machine.
>
> `rem_account` — the username you will use on the *remote* machine.
>
> `rem_device` — the remote non-rewinding tape drive on the *remote* machine that you will use to read the tape. If the `rem_device` value is "`prompt`", then you'll be prompted to provide full pathnames for the volumes you retrieved from the distribution tape onto the local disk of the remote machine. These two compressed *tar* files contain the *root* file system, and the *usr* file system.

Thus, if the IP address of the *remote* machine is 123.45.67.89, the account you will use on it is *root* (the account you use must permit read access to the tape drive, or to the local storage containing the binary images), and the no-rewind tape drive device is */dev/nrst4*, then you would type:

```
# mk_install 123.45.67.89 root /dev/nrst4
```

If the binary images reside on a local disk of the *remote* machine then use:

```
# mk_install 123.45.67.89 root prompt
```

You will then be prompted for the absolute pathname on the remote system of the compressed *tar* file containing the *root* file system, and later for the file containing the *usr* file system.

If this step fails using a network install, it is most likely for one of the following reasons:

- you cannot access the *remote* machine
- access is not granted in the *.rhosts* file of the account used to access the *remote* machine

- you specified an invalid remote special device name or incorrect pathname to the *install* images.

If a failure occurs and the tape has been accessed (e.g., a failure occurs during the installation process), remember to rewind the tape before resuming the installation process (e.g., by using the *mt(1)* command).

Local tape drive installations

If you use a local SCSI tape drive to read the distribution tape on the *target* machine, you should insert the tape in the tape drive, wait for the tape to rewind, and then type:

```
# mk_install local
```

Note that the *mk_install* script requires the tape drive to be set for SCSI unit 4. You should see the names of the loaded files displayed on the console as the installation proceeds.

If this step fails using a local DDS-1 SCSI tape, you should verify that the SCSI tape drive is set as SCSI unit 4.

Also, if the tape has been accessed and a failure occurs during the installation process, remember to rewind the tape before restarting the installation process (e.g., by using the *mt(1)* command).

2.4.2.5 Reboot the Target Machine

When software downloading is completed, use *sync* to update the contents of the hard disk on the *target* machine. Then issue a *reboot* command to restart the *target* machine. Once the reboot has begun, remove the floppy from the floppy drive so that your machine will boot from the hard disk.

After the reboot you need to login as *root*. The first time you boot your target machine with MK7.3 you must run the */postboot* script. In addition, standard customization is necessary, and you will need to modify the following site-dependent files:

- */etc/hosts* — to set at least the IP address of the target machine
- */etc/named.boot*, */var/named/hesiod.ca* — for the DNS nameserver
- */etc/resolv.conf*
- */etc/rc.config* — to set the target machine hostname, IP address, netmask, network interface and gateway variables
- */etc/passwd*
- */etc/group*
- */etc/printcap* — to specify network printers and print servers

- */sbin/init.d/inet* — to configure your local network IP and gateway
- */etc/ntp.hosts*, */var/adm/ntp/ntp.conf* — for network time protocol configuration
- */var/adm/sendmail/sendmail.cf* — for sendmail configuration

If you are upgrading an existing MK7.X system, most of these local files will already have been configured.

If the hard disk is larger than 300 MB, there will be an additional partition (*/dev/sd0e* for SCSI, or */dev/hd0e* otherwise) that you may use. You must run the *newfs* command on this partition before it can be mounted as an additional file system. If you want to mount it automatically at boot time, just add an entry to the */etc/fstab* file.

2.5 Source Installation

The MK7.3 source distribution consists of the last 3 volumes of the tape. The fourth volume contains documentation including the MK7.3 release notes and manual pages. The fifth volume contains the source files for MK7.3 and reference-build log files. The last volume provides the (binary) tools needed to build the release.

2.5.1 Extracting Files from the Distribution Tape

Be sure that you have enough disk space before attempting to extract the contents. Table 1 shows the size of each tape volume, while Table 2 lists the approximate disk space required to build the MK7.3 microkernel and server. (The exact size will vary depending upon the configuration selected).

Description	Size
MK7.3 microkernel (object tree)	5 - 7MB
OSF/1 server (object tree)	8 - 10 MB

Table 2: Disk Space Needed for Objects

In order to read the MK7.3 distribution, *cd* to the intended build directory and extract the desired volumes using the non-rewinding tape device (e.g., */dev/nrst4* for a DDS-1 (4mm DAT) drive set for SCSI unit 4 using MK7.3):

$ `cd` <*build_path*>

$ `tar xvf` *non-rewinding-tape-device*

Repeat the *tar* command to extract volumes four through six from the tape. (For some tape devices, you may need to issue an extra *tar* command to skip beyond the end of the current volume).

For the remainder of this section, assume that the source tree has been extracted and placed under the *./mk7.3* subdirectory of the desired build directory:

<build_path>/mk7.3

That is, *<build_path>* is the directory from which you issue the *tar* command to extract the tape archives.

The extracted MK7.3 tree will contain the following subdirectories:

./doc – directory containing release-related documentation

./export – tree for header files & libraries needed when building

./obj – tree for object files, and executables made when building

./src – tree containing source code

./tools – native tools, scripts, and compiler binaries for building

./logs – reference build logs for source (as built by the RI)

./boot - bootstrap floppy disk images

The source code provided under *./src* includes only the source files required to build the MK7.3 microkernel and server, the associated software needed for the system to boot, plus related MK commands and utilities. Note that the source tree does not include those commands and libraries that are unchanged from the OSF/1 1.3 release.

2.5.2 Building the Release

This build process should work on a DEC PC/AT class machine running MK7.2 or later release (see Section 2.2 above). The *./src* tree contains all the source files needed to build the microkernel, server and related MK utilities. However the MK7.3 source distribution does not include source code for the entire set of commands and libraries from OSF/1 1.3. In order to build the full collection of UNIX commands and libraries, you will require the complete OSF/1 source release and should contact The Open Group Direct (direct@opengroup.org).

The remainder of this section assumes that the contents of the source archive reside under the directory *<build_path>/mk7.3* as described above.

There are a few simple steps to follow when building the source tree. The build process is divided into steps that create MK7.3 components in the required order.

In order to build a complete bootable MK7.3 system, you must build all of the following system components (listed with their default pathnames):

Primary bootstrap	*/stand/boot*
Secondary bootstrap	*/mach_boot*
Bootstrap task	*/mach_servers/bootstrap*
MK microkernel	*/mach_kernel*
Default pager	*/mach_servers/default_pager*
Name Server	*/mach_servers/name_server*
MK Server	*/mach_servers/startup*

There are a variety of different kernel and server configurations you can build, and you should be sure you understand the ramifications of the options you select. The steps below describe the build procedure for compiling the entire MK7.3 system.

If any step fails in the build process below, it is important to identify the problem, fix the error, and rebuild, before proceeding to the next step. Any errors will appear in the log files preceded by the string "*** Error". Reference logs from each step (as built by the RI) are provided under *<build_path>/mk7.3/logs*. Although pathnames to the build tree and tools will differ, you can generally compare the reference logs to the output from your build logs below.

The build process attempts to use several utilities and libraries installed in default locations (*/bin*, */lib*) on your target machine. In particular, the process expects to find the following:

Utilities:

cc
lex

yacc

Libraries:

libc.a
libl.a
libln.a

The following steps assume you're using a *sh*(1)-like shell; the example target machine to build for is a PC/AT platform running the MK7.3 binary distribution.

Sources for the FSF *gcc* and the ODE minimal tools set are *not* included here. Only the binaries for a PC/AT platform running an MK-based OSF/1 are included (in volume 6 of the distribution tape). If you need to re-build or port these tools for different platforms, please request a tools distribution from The Open Group Research Institute.

1. Set your path to pick up the build tools:

```
$ PATH="<build_path>/mk7.3/tools/at386/hostbin:$PATH"
$ export PATH
```

2. Copy the build.sh script from the setup area, and build header and MIG definition files:

```
$ cd <build_path>/mk7.3/src
$ cp setup/AT386/build.sh .
$ chmod +x build.sh
```

This step populates the *./export* subtree with header files:

```
$ sh -f build.sh AT386 osf1 . \
  MAKEFILE_PASS=FIRST 2>&1 | tee first.log
```

The reference log is *<build_path>/mk7.3/logs/first.log*.

3. Build the microkernel executable file:

The default configuration for the MK7.3 microkernel is *PRODUCTION*. You can change this configuration by setting the environment variable *MACH_KERNEL_CONFIG* to one of the other available configurations before running *build.sh*. Common configurations include:

- *PRODUCTION* – maximal performance, with no debugging capabilities.
- *FAST* – almost-comparable performance, but with kernel debugger (*ddb*) and all task symbol tables provided.

- *FAST+RT* – development-oriented, with real-time features enabled.
- *FAST+RT+ETAP* – like *FAST+RT*, but with ETAP-related code enabled.

Those configurations that include “RT” in their name are referred to as real-time configurations. They provide kernel features, such as kernel preemption, that are intended to support real-time applications. (N.B.: Currently preemption only works on uniprocessor systems).

```
$ cd <build_path>/mk7.3/src
```

```
$ sh -f build.sh AT386 osf1 \
  mach_kernel build_all 2>&1 | tee ukernel.log
```

The reference logs are *<build_path>/mk7.3/logs/ukernel*.log*.

4. Build all libraries used by the server, default_pager and commands:

```
$ cd <build_path>/mk7.3/src
```

```
$ sh -f build.sh AT386 osf1 . \
  MAKEFILE_PASS=SECOND 2>&1 | tee second.log
```

After this step, you should find the following files under *<build_path>/mk7.3/export/at386/usr/ccs/lib*:

```
$ ls <build_path>/mk7.3/export/at386/usr/ccs/lib
```

```
crt0.o          libgipc.a       libpthread.a
gcrt0.o         libm.a          librthreads.a
libboot.a       libmach.a       libsa_fs.a
libc.a          libmach_p.a     libsa_mach.a
libcthreads.a   libmachid.a     libservice.a
libcurses.a     libnetmemory.a  mcrt0.o
libflipc.a         libnetname.a
```

(There is a separate subdirectory named *posix_draft_4* that contains an earlier version of the POSIX threads library, *libpthreads.a*).

This step will also build all necessary libraries and exported header files. The reference log is *<build_path>/mk7.3/logs/second.log*.

5. Build the bootstrap task:

```
$ cd <build_path>/mk7.3/src
```

```
$ sh -f build.sh AT386 osf1 \
  bootstrap build_all 2>&1 | tee bootstrap.log
```

The reference log is *<build_path>/mk7.3/logs/bootstrap.log*.

6. Build the default_pager:

```
$ cd <build_path>/mk7.3/src
```

```
$ sh -f build.sh AT386 osf1 \
  default_pager build_all 2>&1 | tee pager.log
```

The reference log is *<build_path>/mk7.3/logs/pager.log*.

7. Build all remaining MK commands and services:

```
$ cd <build_path>/mk7.3/src
```

```
$ sh -f build.sh AT386 osf1 \
mach_services build_all 2>&1 | tee services.log
```

The reference log is *<build_path>/mk7.3/logs/services.log*.

8. Build the OSF/1 server:

The default configurations for the MK7.3 server are *PRODUCTION* and *DEBUG*. You can change the configuration by setting the environment variable *SERVER_CONFIG* to one of the alternate configurations before running *build.sh*:

- *PRODUCTION* – maximal performance, meant for use in production systems.
- *FAST* – almost-comparable performance, but with development aids provided.
- *DEBUG* – development-oriented, with (e.g.) server assertions and lock debugging enabled.

```
$ cd <build_path>/mk7.3/src
```

```
$ sh -f build.sh AT386 osf1 \
  osf1_server build_all 2>&1 | tee server.log
```

Note that by default the server is built in two flavors: both collocated, as *osf1_server.<config>.INK* (loadable into the kernel address space) with an *.INK suffix, and as *osf1_server.<config>* built to run in user space.

Also note that the *FAST* and *PRODUCTION* configurations are currently identical, although this may change in future releases. The reference logs are *<build_path>/mk7.3/logs/server*.log*.

9. Build the primary and secondary boot programs:

```
$ cd <build_path>/mk7.3/src
```

```
$ sh -f build.sh AT386 osf1 \
stand build_all 2>&1 | tee stand.log
```

The reference log is *<build_path>/mk7.3/logs/stand.log*.

Copy the microkernel image to the root directory with a name that identifies the kernel configuration. See the next section for directions on how to boot this image on a machine.

2.5.3 Installing and Booting the Built Images

You must install the microkernel and server images built during the previous steps before booting them. We suggest giving images names that reflect the configuration with which they were built, and the purpose for which the build was done. For instance, you might copy the bootable kernel produced into the root directory as */mach_kernel.MK7.FAST*, and copy the collocated server produced into the */mach_servers* directory as *osf1_server.MK7.FAST.INK*.

Now the new images can be booted by specifying a string like this one at the boot prompt (see the next section):

```
boot: mach_kernel.MK7.FAST -- /mach_servers/bootstrap
```

To set these images to be the defaults, use hard links (or a symbolic link for the server) to install them with the pathnames */mach_kernel* and */mach_servers/startup.INK*. Naturally, you should save the initial default images under different names before doing this.

```
$ ln /mach_kernel.MK7.FAST /mach_kernel
$ ln -s /mach_servers/osf1_server.MK7.FAST.INK \
  /mach_servers/startup.INK
```

You might also create the file */mach_servers/bootstrap.conf*. After the boot loader loads the kernel and the bootstrap task, the kernel is given control and launches the pre-loaded bootstrap task as the first task in the system. The bootstrap task, in turn, is responsible for loading and starting additional servers. The set of servers it starts are specified in the *bootstrap configuration file*, whose default pathname is */mach_servers/bootstrap.conf* on the *root* filesystem. The bootstrap configuration file allows you to customize the file names and options for each server that gets started as part of the normal boot process.

The following example shows a typical *bootstrap.conf* file for an MK7.3 configuration, using an in-kernel OSF/1 (UNIX) server:

```
name_server /mach_servers/name_server
def_pager /mach_servers/default_pager sd0b
-k startup /mach_servers/osf1_server.INK sd0a
```

Boot Options

During the execution of the bootblock code, the console displays the secondary boot prompt (`boot:`) and pauses for several seconds. By typing a space character (“ ”) at the console at this time, you can cause the bootstrap to suspend execution of its default parameters and to wait for a specific boot command to be entered. This allows the user to override the default parameters, which boot the microkernel named */mach_kernel*, and start the server named */mach_servers/startup*.

The boot command has the following general syntax:

disk(unit, partition)/kernel [k-options] bootstrap [b-options]

where:

a. *disk* is either *hd* for AT, IDE, and ESDI disks or *sd* for SCSI disks
b. *unit* is the logical disk unit number from which you want to boot (usually 0 or 1; default is 0).
c. *partition* is the partition containing the root file system (value between *a* to *p*; default is *a*).
d. *kernel* is the name of the file on the root partition containing the microkernel binary you want to boot (default is *mach_kernel*). If this name contains no pathname prefix, it will be searched for in the root directory.
e. *k-options* is a set of options for the kernel, including:

   -h — kernel will halt and invoke the debugger as soon as possible (also passed to server)

   -s — single-user mode (passed to server)

   -- — indicates end of kernel options
f. *bootstrap* is the name of the file on the root partition containing the bootstrap task binary you want to boot (default is */mach_servers/bootstrap*). If this name contains no pathname prefix, it will be searched for in the root directory.
g. *b-options* is a set of options for the bootstrap task, including:

   -a — ask for the name of the bootstrap.conf file. This option permits non-default servers to be used as name server, default pager, and OS personality.
h. Every field is optional.

For example, if you want to have the boot program prompt you for the names of all configuration files and servers, at the boot prompt, type:

```
boot: -- -a
```

The bootstrap will prompt for the configuration file and each server, printing a default and allowing you to edit each filename. So, for example, to boot files located in the */mach_srv.old* directory (user input shown in bold):

```
boot:-- -a
Loading sd(0,a)/mach_kernel
583059+121020+172244[+59728+59135]
Loading sd(0,a)/mach_servers/bootstrap
90331+5708+204[+12000+10970]

...

Configuration file:/dev/boot_device/mach_srv.old/bootstrap.conf
name_server: /dev/boot_device/mach_srv.old/name_server
default_pager: /dev/boot_device/mach_srv.old/default_pager
startup: /dev/boot_device/mach_srv.old/osf1_server.FAST.INK
```

2.6 Common Problems

Here are some trouble-shooting hints should problems arise during the installation process:

- Unless you are running X11, MK will use a “monochrome 80 x 25” mode character display. Some VGA cards misbehave; you might get better results with a simple monochrome board if you have one. Also, verify that you have initialized your VGA board for this mode; some boards require special setup.
- If you do not have direct access to a UNIX machine with a tape drive, you should transfer (by whatever means you have) the first two volumes of the MK7.3 distribution tape to some disk location on a UNIX system. (These two volumes are copies — written by *dd* — containing compressed *tar* images of the *root* and *usr* filesystems, respectively.) This machine will then become the *remote* machine, and in the *mk_install* command, you can use the *prompt* option, and give pathnames for the extracted compressed *tar* files.

  For instance, on an MK7.X system with a DDS-1 (4mm DAT) tape drive set to SCSI unit 4, you could copy the binary volumes to disk using the *dd* command:

  ```
  $ dd if=/dev/nrst4 of=root.tar.Z
  $ dd if=/dev/nrst4 of=usr.tar.Z
  ```

2.7 Supplemental Information

The final section of this chapter provides detailed information that you may require for configuration and debugging during installation.

2.7.1 Ethernet Board Configuration

This section specifies the proper configuration of the Western Digital(SMC) WD8003/WD8013 EP/WD8216C family of ISA bus network adapter cards for use in MK systems.

You should select 0x280 as I/O port address and IRQ 2 for the interrupt level; the dual RAM base address should be located at 0xD0000. Note that IRQ 2 and IRQ 9 are equivalent for the PC/AT platform. The microkernel also probes for a network adapter at these alternate settings: port 0x240, RAM address 0xCC000 and IRQ 5 (device wd1), and port 0x300, RAM address 0xD8000 and IRQ 5 (device wd0). The valid settings are listed in Table 3 below:

Device	IRQ	I/O base	RAM address
wd0	2 / 9	0x280	0xD0000
wd0	5	0x300	0xD8000
wd1	5	0x240	0xCC000

Table 3: ISA Ethernet adapter settings

It may be necessary to first change these settings for the network adapter using the (DOS) setup utility program that came with the board. The factory default values for SMC / Western Digital network adapters are *not* the proper configuration for use with MK7.3. You should make sure the board jumpers are set for "software" configuration, and then run EZStart or EZSetup to change the default settings.

2.7.2 Debugging {with} MK

To activate the kernel debugger you must press the *CTRL+ALT+d* keys simultaneously. In addition to the *ddb* manual pages shipped with this guide, here are some hints:

If an OSF/1 server under debug is being run in user space, then you may have to tell *ddb* to do "thread-relative" operations when you want to access the server's address space. This tells *ddb* how to access the address space. None of this is relevant to collocated servers (which are the kind produced by default in the build process).

You specify a thread-relative reference by adding the modifier "*t*" to the *ddb* command you use, and adding the name of the thread to the end of the command line. For instance, the syntax of common *examine* and *write* commands becomes:

db> *x/<flag>t unix::<varname> $task<n>.<p>*

db> *write/tu unix::<varname> $task<n>.<p> <value>*

where:

- *<flag>* is the data-type flag, one of: *x*, *d*, *i*, ...
- *<varname>* is the server variable or function name.
- *<n>* is the task number of the server, as printed by "show all threads".
- *<p>* is the number of any thread in the task (0 works).
- *<value>* is the new value to be assigned to the variable *<varname>*.

Again, the "*t*" flag, in both commands, indicates that a thread specifier (of the form *$task<n>.<p>*) appears after the symbol name.

The "*u*" flag in the write command specifies a reference to user space (i.e., the server is running in user mode and is not collocated).

2.7.3 MK Paging

MK uses a paging partition that can be specified in either of two files:

- As a command line argument to the microkernel default pager, as defined in */mach_servers/bootstrap.conf* (e.g., *"sd0b"*)
- Or as a "sw" entry in the */etc/fstab* (used by the *swapon* command)

Any use of anonymous *mmap* will be backed by the vnode pager of the OSF/1 server. Temporary memory created by MK microkernel services, such as *vm_allocate* and *copy-on-write*, will be backed by the default pager.

Since shared-library programs are loaded by use of *mmap*, they will have their memory backed by the vnode pager. Statically bound programs will be backed by the default pager.

2.7.4 Sample Disk Partition Table

The *vtoc* utility is used to partition and format the disk. The MK7.3 binary distribution installation scripts assume that the disk uses standard partition assignments. See Table 4 below for a description of the disk partitions.

Partition	Contents	Overlap	Minimum size
a	*root* filesystem	c	40Mb
b	paging	c	80Mb
c	entire disk	all	350Mb
d	*usr* filesystem	c	100Mb
e	optional 2nd *root*	c	0
f	optional 2nd *usr*	c	0

Table 4: Disk Partitions

By convention, the *b* partition is used for paging space. The minimum size of this partition depends upon the amount of physical memory configured on the system. For reasonable virtual memory performance, we recommend that the size of the paging partition should be at least three to four times the size of memory.

The *vtoc* command for a typical MK7.3 workstation configuration might report the following partition sizes:

```
The static data for the vtoc is:
     cyls 3980/ heads 5/ sectors 104; alt ptr 15360 alt len 2048
     4 UNIX partition entries.
a: start 1040, end 1332239, size 1331200 (650 Meg), flag 200, tag 2
b: start 1332240, end 1639439, size 307200 (150 Meg), flag 201, tag 4
c: start 1, end 2069599, size 2069599 (1010 Meg), flag 200, tag 5
d: start 1639440, end 2049039, size 409600 (200 Meg), flag 201, tag 4
```