ULTRIX-32
Installation Guide

Order No. AA-BG58C-TE

ULTRIX-32 Version 1.2

Digital Equipment Corporation

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

DEC
DECnet
DECUS
MASSBUS
MicroVAX
PDP
Q-Bus
RT
ULTRIX
ULTRIX-11
ULTRIX-32
ULTRIX-32m
ULTRIX-32w
UNIBUS
VAX
VAXstation II
VAXstation II/GPX
VMS
VT
Worksystem
digital™

This manual was written and produced by the ULTRIX™ Documentation Group in Merrimack, New Hampshire.

Contents

About This Manual

3 Phase 1: Bring Up The Preliminary Systems

4 Phase 2: Bring Up the Full System

5 Phase 3: Selecting Software Subsets

6 Phase 4: Install Local Area Network and/or uucp Connection

7 Phase 5: Define Printer Capabilities

C The ed Command Summary

Index

Example

Figures

Tables

About This Manual

This manual tells you how to install Version 1.2 of the ULTRIX-32 Operating System on your VAX-11/730, VAX-11/750, VAX-11/780, VAX-11/785, VAX 8600, or VAX 8200 processor.

Audience

This manual assumes that your VAX system hardware has been installed and checked out, and that you know how to load and prepare the disk and/or tapes used during the installation. (Chapter 1 lists these devices, and the User's Guide for each device describes loading and unloading operations.)

You should be familiar with one of the text editors available on the ULTRIX-32 system, such as ed. If you are not, Appendix C of this manual gives a command summary for ed, the simplest of the text editors.

Organization

Chapter 1 — Introduces the hardware you use to install ULTRIX-32 and the major pieces of the software you install.

Chapter 2 — Gives you an overview of the first four phases of the installation procedure. After you have completed the first four phases, you have a running system. You can perform the last two phases immediately, or at a later time. Chapter 2 contains a checklist of information you need to have at hand before beginning the installation.

Chapter 3 — Describes Phase 1 of the installation procedure, Bringing Up the Preliminary Systems.

Chapter 4 — Describes Phase 2 of the installation procedure, Bringing Up Your Configured System.

Chapter 5 — Describes Phase 3 of the installation procedure, Selecting Software Subsets.

Chapter 6 — Describes Phase 4 of the installation procedure, Installing the Local Area Network and/or uucp Connection.

Chapter 7 — Describes Phase 5 of the installation procdure, Define Printer Capabilities.

Chapter 8 — Describes Phase 6 of the installation procedure, Install sendmail for Networks.

Appendix A — Lists error messages that you may see during the installation and what to do to correct the problems that they denote.

Appendix B — Gives detailed information on the format of the configuration file.

Appendix C — Summarizes commands for the ed text editor.

Conventions

This manual uses the following conventions to illustrate the installation process:

special type — In text, each mention of a specific command, option, partition, path name, directory, or file is presented in this type.

command(x) — In text, some mentions of an ULTRIX command include the section number in the Programmer's Manual where the commands are documented. For example, the cat(1) command appears in Section 1 of the Programmer's Manual.

italicized — In syntax descriptions, this type indicates terms that are variable.

lowercase

UPPERCASE — The ULTRIX system differentiates between lowercase and uppercase characters. In examples, enter uppercase characters only where specifically indicated by an example or a syntax line. Uppercase and lowercase characters are significant

`example` — In examples, input or output text is printed in this type.

color — In examples, system messages and prompts are shown in black. User input is shown in this color.

<KEYNAME> — In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal.

<CTRL/d> — In examples, symbols like this indicate that you must hold down the CTRL key while you type the key that follows the slash.

This combination of keys may appear on your terminal screen as the letter preceded by the circumflex character (^). In some instances, it may not appear at all.

<CTRL/u> Press <CTRL/u> to erase a line of input.

Introduction 1

This manual tells you how to install ULTRIX-32 software on your VAX processor. Figure 1-1 shows the six phases of the installation process. Note that phases five and six are optional.

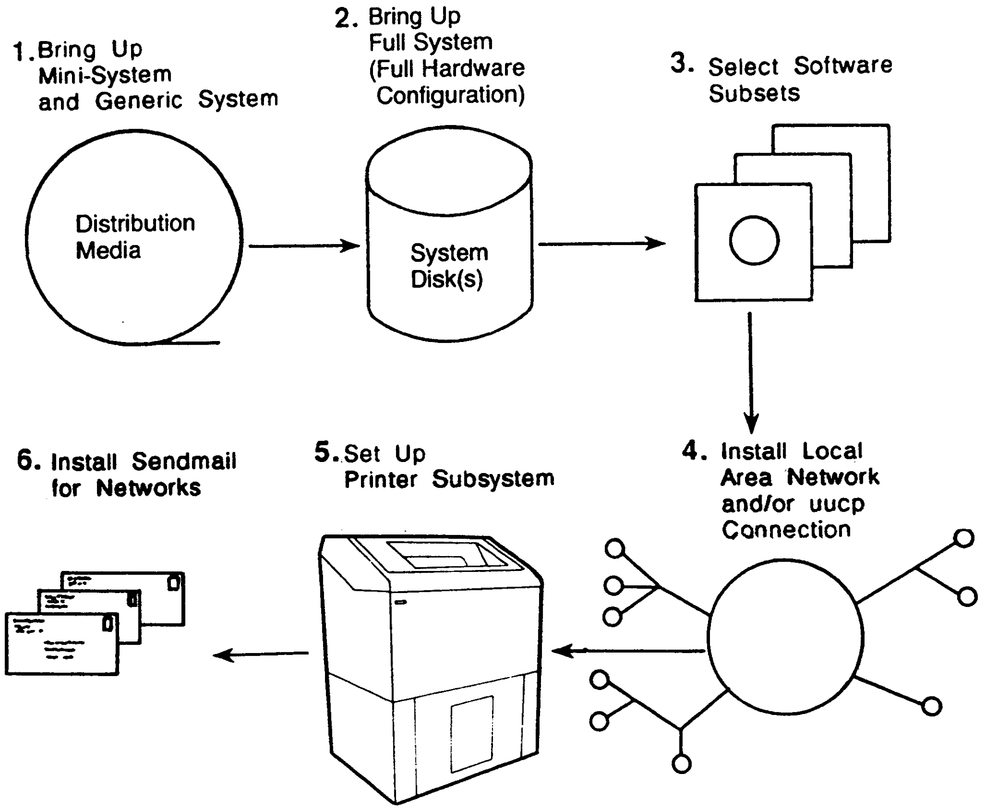

Figure 1-1: ULTRIX-32 Installation Phases

1.1 The Hardware Needed for Installation

During the installation process. you issue commands at the console terminal to:

- Transfer the software from the distribution media (a tape or disk pack) to the disk you select as the system disk (a high-speed input/output device).
- Load and run the software you select in the VAX processor.

Figure 1-2 shows a general picture of the hardware used in the installation.

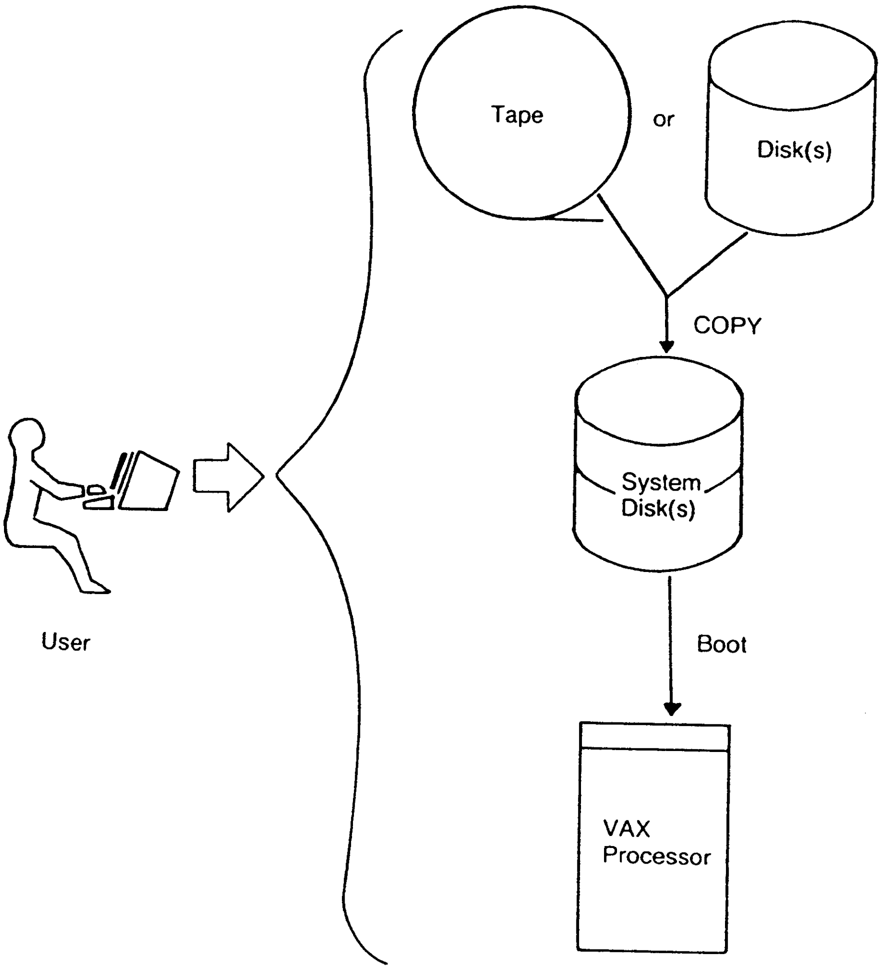

Figure 1-2: Overview of the Hardware Used During Installation

1.1.1 VAX Processor

ULTRIX-32 runs on the VAX-11/730, VAX-11/750, VAX-11/780, VAX-11/785, VAX 8600, and VAX 8200 processors.

1.1.2 Load Device

Table 1-1 lists the devices from which you can read distribution media. If you do not know how to load a tape or disk and prepare it for reading, see the hardware User's Guide for the load device.

1.1.3 System Disk

Table 1-2 shows the system disks that you can use for the various processors. You should know how to load a disk pack (if it is removable) on your disk drive and get the disk ready for read/write operations. If you do not, see the hardware User's Guide for your drive.

1.1.4 Console Terminal

The console terminal, at which you do most of the installation, should be a hard-copy terminal, so you can have a record of the installation. You can use a screen terminal, however.

Figure 1-3 shows three hard-copy terminals you can use: an LA120-DA, an LA100-BA, and an LA12-DB terminal.

Table 1-1: Load Devices

Processor	Load Devices
VAX 8600	TU80 tape drive TU81 tape drive TU77 tape drive TU78 tape drive TE16 tape drive TU45 tape drive TS05 tape drive TS11 tape drive
VAX 8200	TU80 tape drive TU81 tape drive TS05 tape drive TS11 tape drive
VAX-11/780 VAX-11/785	TU80 tape drive TU81 tape drive TU77 tape drive TU78 tape drive TE16 tape drive TU45 tape drive TS05 tape drive TS11 tape drive RA60 disk drive
VAX-11/750	TU80 tape drive TU81 tape drive TU77 tape drive TU78 tape drive TE16 tape drive TU45 tape drive TS05 tape drive TS11 tape drive RA60 disk drive
VAX-11/730	TU80 tape drive TU81 tape drive TS05 tape drive TS11 tape drive

Table 1-2: System Disks

Processor	System Disk
VAX 8600	RA80 121-Mbytes fixed disk RA81 456-Mbytes fixed disk RA60 205-Mbytes removable disk RM05 256-Mbytes removable disk RP07 516-Mbytes fixed disk RM03 67-Mbytes removable disk RM80 12-Mbytes fixed disk RP05 100-Mbytes removable disk RP06 200-Mbytes removable disk
VAX 8200	RA81 456-Mbytes fixed disk RA80 121-Mbytes fixed disk RA60 205-Mbytes removable disk
VAX-11/780 VAX-11/785	RA80 121-Mbytes fixed disk RA81 456-Mbytes fixed disk RA60 205-Mbytes removable disk RM05 256-Mbytes removable disk RP07 516-Mbytes fixed disk RM03 67-Mbytes removable disk RM80 12-Mbytes fixed disk RP05 100-Mbytes removable disk RP06 200-Mbytes removable disk
VAX-11/750	RA80 121-Mbytes fixed disk RA81 456-Mbytes fixed disk RA60 205-Mbytes removable disk RK07 28-Mbytes removable disk RM05 256-Mbytes removable disk RM03 67-Mbytes removable disk RM80 12-Mbytes fixed disk RP05 100-Mbytes removable disk RP06 200-Mbytes removable disk RP07 516-Mbytes fixed disk
VAX-11/730	RA80 121-Mbytes fixed disk RA81 456-Mbytes fixed disk RA60 205-Mbytes removable disk RK07 28-Mbytes removable disk R80 121-Mbytes fixed disk

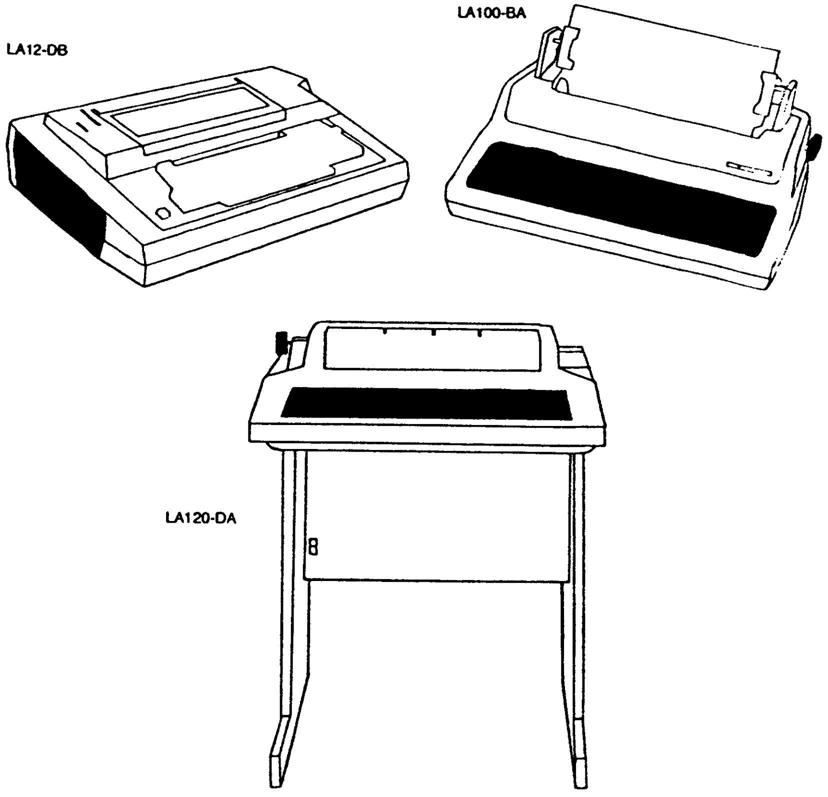

Figure 1-3: Hardcopy Console Terminals

1.1.5 Networking Hardware (Optional)

If your system is to be part of a computer network, you must install the proper hardware. Two network controllers are available:

- DEUNA – This controller provides access to a 10 Mbytes/sec Ethernet network.

- DMR-11 – This controller provides access to a point-to-point communications device that runs at either 1 Mbytes/sec or 56 Kbytes/sec.

1.2 The Software You Install

The ULTRIX-32 system is a multiuser, multitasking operating system. Many users can use the system at the same time, and each user can run more than one program at the same time.

Figure 1-4 shows a simplified view of the general structure of the system software. The software is shown in three layers, each layer indicating a different level of control in the overall system.

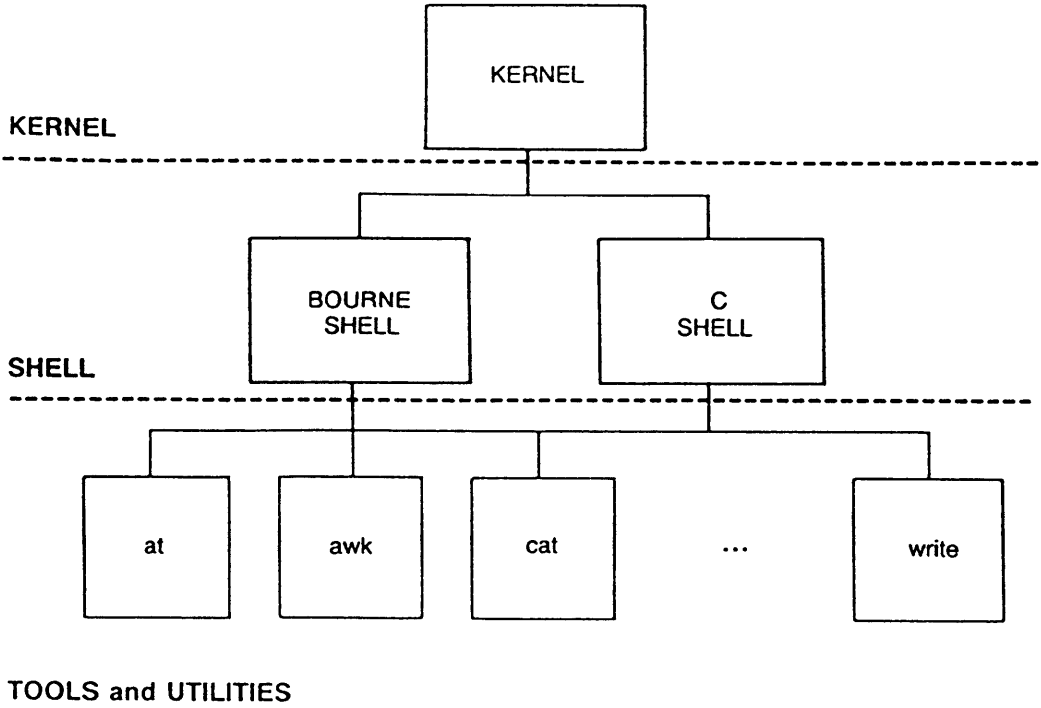

Figure 1-4: A General Diagram of the Software

1.2.1 Kernel

The part of the system called the kernel has control over the rest of the software running on the system. Some important functions the kernel performs are:

- Scheduling – The kernel determines which of the programs operating at different priorities are to run in the processor next.
- Paging and swapping – The kernel loads programs to be run into the processor and returns them to disk after their time slice of execution in the processor has passed.
- Input/output – The kernel contains software drivers that actually read data from and write data to the hardware devices.
- Managing the File System – The kernel maintains the internal directory and subdirectory structure of the ULTRIX-32 file system.

During Phases 1 and 2 of the installation procedure, you install the kernel as the running program in the processor. Once the kernel is installed, the commands you give during the installation procedure cause programs to be scheduled and executed under the control of the kernel.

1.2.2 Shells

A shell is a command language interpreter. This software examines the commands users type and passes control to the appropriate utility program, development tool, or application program to do what the user wants.

ULTRIX-32 software provides two shells: the Bourne shell and the C shell.

Those shells function as both command interpreters or programming languages. Users can write shell scripts to perform several actions with a single command. The C shell offers some capabilities that the Bourne shell does not. For example, the C shell maintains a history or record of commands processed for each user. A user can then reexecute any particular command using fewer keystrokes than needed to retype the original command.

You install both shells in your system. The Bourne shell is installed in Phase 1 of the installation, so that you can use its commands during the installation process. The C shell is installed in Phase 3. Your users can then select which shell they want to use at any given time.

1.2.3 Utilities

These programs provide the command environment that the user works with on an ULTRIX system. They include program development utilities, such as the C compiler and the ld linker. Other utilities provide general-purpose commands that your ULTRIX-32 system users will find useful, such as, the who program that lists the current users on the system.

1.3 The Software Distribution Kit

The ULTRIX-32 software distribution kit contains the ULTRIX-32 software on distribution media such as magnetic tapes or disk pack. The distribution media in your kit can be any of the following, depending on how you ordered ULTRIX-32:

- Two 9-track magnetic tapes, labeled ULTRIX-32 V1.2 BIN MT9 and numbered 1/2 and 2/2.
- One RA60 disk pack, labeled ULTRIX-32 V1.2 BIN RA60 and numbered 1/1.

Media with multiple volumes (except the RA60 disk pack) are also labeled SYSTEM SOFTWARE and OPTIONAL SOFTWARE as appropriate.

The distribution kit (for all processors other than the VAX 8600) also includes software needed to get the system software from the distribution media to the system disk and load it into the computer, or boot it. This software is on either a diskette or tape cassette labelled ULTRIX-32 V1.2 Boot 1 of 1.

In addition, the distribution kit contains the ULTRIX-32 documentation, of which this Installation Guide is a part.

Overview of the Automatic Phases 2

This chapter gives some background detail on the first four phases of the installation process. Most of the steps in these four phases are automated. That is, you answer questions presented by command files called shell scripts. Reading this chapter gives you an understanding of what is happening behind the scenes with each phase.

In addition to general background information, this chapter tells you what you need to know before beginning the installation process. Your installation will proceed more smoothly if you read this chapter and fill out the checklist in the last section, before beginning to install your ULTRIX-32 system.

2.1 Phase 1: Bringing Up the Preliminary Systems

During Phase 1 of the installation process, you install two preliminary systems that you use to build the final system in Phase 2. The final system reflects the actual or planned hardware configuration at your site.

The first preliminary system is called the minisystem, which is a subset of the full ULTRIX-32 system. It consists only of the software needed to bring up the generic system, the second preliminary system installed in Phase 1.

To begin, you load the first tape or disk of the distribution media on the load device and get the system disk ready for read/write operations.

Next, you copy the minisystem from the distribution media onto the system disk, load it into memory in the VAX processor, and execute it. You perform these steps by typing commands at the console terminal. The commands are executed by the VAX console subsystem, which is part of the VAX hardware system.

When you boot the minisystem, it runs a shell script that begins creating the ULTRIX-32 file system, by creating a directory called the root directory on the system disk and certain subdirectories within the root directory.

The script then extracts files from the distribution media and puts them on disk in the appropriate subdirectories:

- /etc – This directory contains executable files for various system management tools, such as backup and shutdown.

- /bin — This directory contains binary (executable) code for the shells, in addition to commonly used utilities such as mail, who, and write.
- /vmunix — This file contains the binary code for the generic system used during Phase 1.
- /sys — This directory contains shell scripts and the kernel object files for procedures related to system management.

At this point, you update the console device and then load and execute the generic system.

2.2 Phase 2: Bringing Up Your Configured System

During the second phase of the installation process, you compile and link your system to reflect the actual or anticipated hardware configuration at your site.

The generic system runs a shell script to request the following information from you at the console terminal:

- The date and time.
- The root (or superuser) password. From this point on, anyone logging in under the name root will be expected to provide this password.
- The field service account password. From this point on, anyone logging in under the name field will be expected to provide this password.

After you supply the necessary information, the shell script copies basic files from the distribution media to the /usr disk directory. It then creates the /usr/users file system on the system disk.

The shell script also copies the kernel code from the distribution media to the system disk. (This kernel code is later used in compiling and linking the final configured system.)

The script then unmounts and checks the consistency of the /usr directory and remounts the /usr directory so it will be ready to access.

Next you indicate if you want crash dumps for the system. If you choose to enable crash dumps, then every time that the system crashes, the contents of memory is written to the system disk, in the area allocated for paging and swapping during normal system operation.

Next you are asked to define the hardware configuration. The shell script polls the system to determine what hardware is currently connected. It lists the hardware it finds, and asks you to define the communications hardware (such as terminal controllers) on the system. You are then asked a series of questions that define the hardware you want to add.

The shell script adds this information to what is called the configuration file for your system. This file is used as a specification to another shell script that compiles and links the proper software (I/O drivers) for the hardware you have or expect to have on your system.

At this point, you load and execute your new system.

During the final part of Phase 2 of the installation process, you configure terminals so that more than one person can access the ULTRIX-32 system simultaneously. This step is not automated; you must edit these fields in the /etc/ttys file:

- **terminal name** – This field defines a name for each terminal special file in the /dev directory.
- **command** – This field defines the command used by the init program to initialize such terminal line characteristics as the baud rate (the speed at which the terminal sends and receives characters).
- **terminal type** – This field defines the type of terminal (such as VT100) associated with each terminal name in /etc/ttys.
- **terminal status** – This field defines a variety of actions permitted at the terminal such as logins and root logins.

2.3 Phase 3: Selecting Software Subsets

Some pieces of the ULTRIX-32 software are optional. If none of your users are programmers, for example, you can save disk space by not installing the Pascal compiler and its associated libraries.

During this phase of the installation process, you run a shell script called instmedia, which presents a menu that lists available optional software. When you select an item from the menu, the program loads that software from the distribution media into its proper directory on the disk partition.

The shell script presents the menu again, and the process is repeated until you select the exit option from the menu.

The optional software includes:

- Optional development software, such as Pascal, Fortran, Modula2, and so forth.
- Local area networking software, such as rlogin, which allows users to log in to remote computers.
- Games. This software includes computer games such as adventure and backgammon.
- Documentation for man. This is the on-line documentation presented by the man utility.

- Optional documentation. These are the nroff source files for the *ULTRIX-32 Supplementary Documents.*
- The learn program. This program presents tutorial information on ULTRIX-32 to users, and its associated text.
- Font support. This software includes the Berkeley font libraries for the troff program.

At this point, you can go to multiuser mode by pressing <CTRL/d>. However, you should remain in single-user mode if you plan to complete Phases 4, 5, and 6 immediately. If this is the case, you can go to multiuser mode after you complete Phase 6.

2.4 Phase 4: Installing the Local Area Network and/or uucp Connection

During this phase of the installation, you indicate whether your system is to be part of either or both of two types of networks:

- Local area network. This type of network connects systems in one area, such as the floor of a building, or a university campus. The systems in the network are usually connected with direct high-speed lines.
- uucp network. This type of network uses uucp for data transfer between computer systems. The systems in such a network are usually connected over indirect lines such as a telephone connection.

2.4.1 Local Area Networks

With local area networks, users give commands such as rlogin (log in to remote system) and rcp (copy file to or from remote system).

During this phase of the installation process, you run a shell script called netsetup that asks if you want to establish your system as part of a local area network. If you do not, no further questions are presented, and you proceed to the shell script for uucp. If you do want to connect to a local area network, you need to know:

- System abbreviation. – Local and remote users can use this abbreviation in referring to the local system. It is usually the first character of the full system name you specify at the end of Phase 2. In general, use the least number of characters needed to make the abbreviation unique on the network.
- Network number. – This number can be between 1 and 126. This number identifies the network in network topologies where multiple networks are connected. The number you give must be the same as the number given for other systems on your local area network.

- System number. – This number can be between 1 and 999. It identifies your system on the local area network and is used as an identifying address by data transfer utilities.
- Network name and alias. – Utilities use the network name and alias to identify the network when it is connected to other local area networks.
- Name, abbreviation, and system number for each computer on the network. You identify the network known to your system.
- Names of trusted systems on the network. Users with valid accounts on these systems can log in to your system without being prompted for log-in name or password, as long as they have the same log-in name and password on your system as they have on the remote system.

2.4.2 uucp Connection

Here you either reject or define your system as part of a network of systems using the uucp utility to access remote systems.

The commands uucp (remote file copy) and uux (remote shell command execution), for example, provide access to remote systems.

You run a shell script called uucpsetup to make your selection. If you reject uucp connections, you are finished with the automated phases of the installation; no further questions are asked.

If you accept uucp connections, the shell script asks you to define outgoing and incoming connections. Outgoing connections are those which are initiated by your system to a particular remote system. The link is usually made through telephone connections initiated by autodial modems. The following information is requested for each outgoing connection:

- The name of the remote system your system is to connect with.
- The time span during which calls to the remote system can be made. Requests like uux (to execute a shell command on a remote system) can only be executed during this time span. Users can request file transfer at any time. The files will be held for transfer, if necessary, until the period you define here.
- The line speed of the remote system.
- The telephone number of the remote system.
- Your system's log-in name and password to be used when logging in to the remote system.
- The type of modem your system uses to connect to the remote system (DF02, DF03, or DF112 modems).

Incoming connections are those initiated by remote systems to your system. The shell script makes some general definitions for local users and for remote systems you do not specifically define (giving such remote systems the lowest access privileges). It then asks for the following information for each incoming connection:

- The name of the remote system.
- A short comment for the password file.
- The password for the remote system.
- The access level for the remote system. This is a number from 0 through 9. Each number has been defined to allow users on the remote system to execute certain commands on your system. The shell script shows you the commands defined for each level, so you can make a choice.
- The callback option. If you specify this option, you indicate that an incoming call from this system will be rejected, but the local system will call back the remote system. This is a security measure, to ensure that incoming calls are really from valid systems. It means, of course, that your site always pays for the telephone connections with this system.
- Directory path for the remote system. This defines the directory for files transferred from the remote system. The default is the directory /usr/spool/uucppublic.

This terminates Phase 4 of the installation procedure. At this point, you have a running system. The *ULTRIX-32 System Manager's Guide* explains how to add user accounts so that users can log in and use the system. You need to perform the steps in Phases 5 and 6 only if:

- You have a line printer.
- You want to use mail to send mail to different sites in a network.

2.5 Checklist for Automated Installation Phases

The following checklists are provided for your convenience. You should have this information available before you start the installation process described in Chapter 3.

2.5.1 Local Area Network Information

Network Number:

System Number:

Network Name:

Alias:

Remote Nodes:

Name Abbreviation Number

Name Abbreviation Number

Name Abbreviation Number

Name Abbreviation Number

Name Abbreviation Number

Name Abbreviation Number

Name Abbreviation Number

Name Abbreviation Number

2.5.2 UUCP Information

2.5.2.1 Outgoing Connections

Remote System Name ______________________________
Time Span ______________________________
Remote System Line Speed ______________________________
Remote System Telephone Number ______________________________
Log-in Name, Password ______________________________
Modem Type ______________________________

Remote System Name ______________________________
Time Span ______________________________
Remote System Line Speed ______________________________
Remote System Telephone Number ______________________________
Log-in Name, Password ______________________________
Modem Type ______________________________

Remote System Name ______________________________
Time Span ______________________________
Remote System Line Speed ______________________________
Remote System Telephone Number ______________________________
Log-in Name, Password ______________________________
Modem Type ______________________________

2.5.2.2 Incoming Connections

Remote System Name ______________________________
Comment for Password File ______________________________
Password for Remote System ______________________________
Access Level ______________________________
Callback Option ______________________________
Directory Path ______________________________

Remote System Name ______________________________
Comment for Password File ______________________________
Password for Remote System ______________________________
Access Level ______________________________
Callback Option ______________________________
Directory Path ______________________________

Remote System Name ______________________________
Comment for Password File ______________________________
Password for Remote System ______________________________
Access Level ______________________________
Callback Option ______________________________
Directory Path ______________________________

Phase 1: Bring Up The Preliminary Systems 3

This chapter describes how to bring up the preliminary systems (the minisystem and the generic system) for your processor. The chapter is divided into these sections:

- Section 3.1, Bring Up the Preliminary Systems for VAX-11/730
- Section 3.2, Bring Up the Preliminary Systems for VAX-11/750
- Section 3.3, Bring Up the Preliminary Systems for VAX-11/780 and VAX-11/785
- Section 3.4, Bring Up the Preliminary Systems for VAX 8600
- Section 3.5, Bring Up the Preliminary Systems for VAX 8200

Each section consists of a sequence of steps. Read only the section and complete only the steps in the section that pertains to your processor. For example, if your processor is a VAX 8600, then go to Section 3.4 and complete the steps in that section.

3.1 Bring Up The Preliminary Systems for VAX-11/730

This section describes the steps you follow to bring up the two preliminary systems, the minisystem and the generic system, needed to generate your fully configured system.

Some steps require you to load and unload tapes, disks, and console cassettes. Other steps require you to answer questions and to make decisions about some of the questions you need to answer. The responses in color next to the question are those from an example installation. Use the example installation as a guide. The example installation has these characteristics:

- Processor: VAX-11/730
- Memory: 2 Mbytes
- System disk: R80
- Distribution media: Magnetic tape mounted on a TS05 tape drive unit 0 (connected to UNIBUS adapter number 0)

Your responses may differ from the ones in the example installation.

Step 1: Prepare the Load Device

Select the first volume of the distribution media, which on a VAX-11/730 is magnetic tape. The label for the first volume of the magnetic tape distribution media is: ULTRIX-32 V1.2 BIN 16MT9 1/2. Mount the tape on the load device, and prepare it for operation. (Remove the write-enable ring before mounting the tape.) If you do not know how to mount a tape, see the User's Guide that was supplied with the device. The tape should be at beginning-of-tape (BOT) and on line.

Note

> You must select unit 0 as the load device for your magnetic tape distribution media.

Step 2: Prepare the System Disk

Prepare the disk you have selected for your system disk. If it has removable packs, load a formatted disk pack into the drive. (DIGITAL disk packs are shipped preformatted.) Even if the pack is nonremovable, make sure that your system disk pack is spinning and that the Ready light is lit.

Step 3: Insert the Distribution Cassette

Insert the distribution cassette into the console TU58 disk drive on the front of the processor (drive 0).

Step 4: Prepare the Processor

Set the restart switch to OFF and the rotary key switch to LOCAL.

Step 5: Load the copy Program

Halt the system and load the copy program from the distribution media into memory:

```
^P                           (CTRL/P to alert the console subsystem)
>>>H                         (halt the system)
       .
       .
       .
>>>I                         (initialize the system)
>>>L DD0:COPY                (load copy from console TU58)
>>>S 2                       (start the copy program)
```

Step 6: Copy the Minisystem to the System Disk

At this point the copy program is running from memory, and it asks for the necessary information to do the copy. The general form of the query and your response is:

From: *d(x,y)*
To: *s(z,1)*

where:

d Is a two- or three-character ULTRIX device name for the distribution media. See Table 3-1 for a list of ULTRIX device names.

x Is the logical unit number used by the copy program to access the distribution media. The logical unit number is the same as the physical unit number shown on the front of the load device.

y Is 1, indicating the second file on the tape.

s Is a two-character ULTRIX device name for the system disk.

See Table 3-1 for a list of ULTRIX device names.

z Is the logical unit number used by the copy program to access the system disk. The logical unit number is the same as the physical unit number shown on the front of the system device.

1 Indicates that the minisystem is to be built in the b partition. Normally, the b partition of the system disk is the swap area. The minisystem occupies this space only temporarily, until the generic system is built.

When the copy program is finished, it displays the number of blocks copied (300).

Note

The CSR address for the load and system devices must be the standard VAX-11 CSR address, or the copy program will not recognize them.

Because the example installation uses a magnetic tape distribution media mounted on a TS05 tape drive and a R80 system disk, the response is:

```
From: ts(0,1)
To: rb(0,1)
```

Keep the system disk name and logical unit number handy. You need to specify them again in Steps 8 and 9.

Table 3-1: ULTRIX Device Names

DIGITAL Name	ULTRIX Name
UNIBUS Disks	
RA60	ra
RA80	ra
RA81	ra
RK07	hk
IDC Disks	
R80	rb
UNIBUS Tapes	
TS05	ts
TS11	ts
TU80	ts
TU81	tms

Step 7: Load the boot Program

At this point, the copy program has copied the minisystem to the system disk and prompts for another copy. You want to stop the copy program, return to the console subsystem, and halt and initialize the system, to prepare to load the boot program:

```
From:^P                  (return to console subsystem)
>>>H                     (halt the system)
>>>I                     (initialize the system)
>>>L DD0:BOOT            (load the boot program)
>>>S 2                   (start the boot program)
```

Step 8: Boot the Minisystem

The boot program prints this prompt:

```
Boot
:
```

You type the ULTRIX device name, logical unit number, and partition number for the system disk you copied to in Step 6, plus the file name vmunix:

*s(z,1)*vmunix

Again, *s* is the ULTRIX name for the system device, and *z* is the logical unit number, as discussed in Step 6. Because the example installation uses a R80 system disk, the response is:

```
Boot
:rb(0,1)vmunix
```

Step 9: Specify the System Disk

The boot program prints several lines of information about the equipment on the system, and then prompts:

```
root device?
```

You type the ULTRIX device name and logical unit number for the system disk you copied to in Step 6, *followed by an asterisk.* This is the name and logical unit number used by the minisystem to access the system disk. Because the example installation uses a R80 system disk, the response is:

```
root device? rb0*
```

The asterisk is very important. It tells the minisystem not to swap. The normal swap area, partition b of the system disk, is where the minisystem was loaded.

The following line then appears:

```
erase ^?, kill ^U, intr ^C
```

At this point, a shell procedure takes control to present you with some questions, and to perform the installation of the generic system.

Step 10: Load the Generic System

During this step, you answer questions to define the load device, from which the procedure reads the generic system, and the system device, on which the procedure creates the root file system.

In this case, however, you specify the DIGITAL names for the devices. A list of possible load and system devices is displayed before each question.

Because the example installation uses a magnetic tape distribution media mounted on a TS05 drive unit 0, the response is:

```
List of recognized distribution media devices:
    ts11 ts05 tu80 tu81

What type of drive is the distribution media
mounted on [no default]: ts05
```

The script then displays a similar set of questions to define the system disk, where the root file system is to be built. Because the example installation uses a R80 system disk, the response is:

```
List of recognized disk drives to contain
root file system:
    r80 ra60 ra80 ra81 rk07
What type of drive is going to contain the root
filesystem [no default]: r80
```

The script displays a message indicating the system disk logical unit number supported by the installation. The script asks whether you want to specify an alternate logical unit number. You should specify the same logical unit number you specified for the system device in Step 6. Therefore, in most cases your response to the alternate unit number question is n, as in the example installation:

```
Unit 0 is the only supported unit number
for this device.  These installation procedures
will use this unit number unless you specify an
alternate.

Do you wish to specify an alternate
unit number (y/n) [default: n]? n
```

The script then displays two messages and a question. The messages indicate the DIGITAL name and the corresponding ULTRIX device name and logical unit number for your distribution media and system disk. If these values are correct, answer y to the question following the messages, as in the example installation. Otherwise, answer n, and the script will prompt you again for the appropriate values.

```
Your distribution media is mounted on
drive type ts05, whose corresponding
ULTRIX device name and logical unit
number are ts0.

Your system disk (the root device) is
mounted on drive type r80, whose
```

(continued on next page)

```
corresponding ULTRIX device name and
logical unit number are rb0.

Are these correct (y/n) [default: n]? y
```

After you respond with a y, the script displays some informational messages:

```
- - - Build root filesystem on rb0 - - -
           .
           .
           .
- - - Check the filesystem - - -
           .
           .
           .
- - - Restoring the dump image of the root - - -
           .
           .
           .
-*-*-* Root filesystem is now extracted onto -*-*-*
-*-*-* the destination disk filesystem -*-*-*
```

Step 11: Update the Console Cassette

The script displays the following messages, waiting for you to perform the action and press the RETURN key to continue:

```
Please remove the distribution cassette from
drive 0 (the front drive) and insert a blank
cassette.  Be sure that this cassette is
write-enabled, with its record tab in the
left-most position.

Press RETURN when ready (or type 'skip' to skip
this step):

Please make sure the local cassette is loaded in
drive 1 (the internal TU58 drive).  To locate this
drive on a 730, pull out the front of the CPU like
a desk drawer, and you will find drive 1 on the
back right-hand side.
```

(continued on next page)

```
Press RETURN when you have checked this:
```

You may want to use the skip option if you have already updated the console cassette. For example, you may have completed this step, and are backtracking because you made an error later on in the installation procedure.

The local cassette referred to in the script is the standard console cassette that comes with the hardware. If you press the RETURN key at the previous prompt, the shell script displays this question:

```
Really initialize directory? y
```

You must type yes before the shell script continues the update on the blank cassette (the cassette in drive 0). Output from the make program is displayed, indicating the update is in process. The script then displays these directions for you to follow:

```
Remove the local console cassette from drive 1 and
store it and the installation cassette in a safe
place.

Remove the new, updated console cassette from
drive 0 and replace it in drive 1.

Press RETURN when ready:
```

You should write-protect the new, updated console cassette before you place it in drive 1.

Step 12: Boot the Generic System

You are now ready to boot the generic system. The shell script displays a notice and asks you to type:

`^P`	(access console subsystem)
`>>>H`	(halt the processor)
`>>>I`	(initialize the processor)
`>>>B ANY`	(load the boot program)

At this point, the boot program prints its prompt, and you respond with the name of the system disk that you placed the software on in step 10 and the file name vmunix. The second prompt is for the device where the root file system is to be placed (again, the system disk). In general, the sequence is:

Boot
:*s(z.0)*vmunix

```
          .
          .
          .
(information display)
          .
          .
          .
Root device? sz
```

Again, *s* is the ULTRIX name for the system device, and *z* is the logical unit number, as discussed in Step 6. The shell script displays the correct names and unit numbers before the sequence starts.

Because the example installation uses a R80 system disk, the response is:

```
Boot
: rb(0,0)vmunix
              .
              .
              .
(information display)
              .
              .
              .
Root device? rb0
```

This completes the steps for Phase 1 of the installation procedure.

Go to Phase 2.

3.2 Bring Up The Preliminary Systems for VAX-11/750

This section describes the steps you follow to bring up the two preliminary systems, the minisystem and the generic system, needed to generate your fully configured system.

Some steps require you to load and unload tapes and disks. Other steps require you to answer questions and to make decisions about some of the questions you need to answer. The responses in color next to the question are those from an example installation. Use the example installation as a guide. The example installation has these characteristics:

- Processor: VAX-11/750
- Memory: 6 Mbytes
- System disk: RK07 mounted on drive unit 0, connected to UNIBUS adapter number 0)
- Distribution media: Magnetic tape mounted on a TU80 tape drive unit 0 (connected to UNIBUS adapter number 0)

Your responses may differ from the ones in the example installation.

Step 1: Prepare the Load Device

Select the first volume of the distribution media. The labels for the first volume of the various media are:

- Magnetic Tape. Label reads: ULTRIX-32 V1.2 BIN 16MT9 1/2.
- RA60 Disk Pack. Label reads: ULTRIX-32 V1.2 BIN RA60 1/1.

Mount the tape or disk on the load device, and prepare it for operation. If you do not know how to do this, see the User's Guide that was supplied with the device. A tape should be at beginning-of-tape (BOT) and on line. A disk should be spinning and have the Ready indicator lit.

Note

> You must select unit 0 as the load device if your distribution media is magnetic tape.

Step 2: Prepare the System Disk

Prepare the disk you have selected for your system disk. If it has removable packs, load a formatted disk pack into the drive. (DIGITAL disk packs are shipped preformatted.) Even if the pack is nonremovable, make sure that your system disk pack is spinning and that the Ready light is lit.

Step 3: Insert the Distribution Cassette

Insert the distribution cassette into the console TU58 disk drive on the front of the processor (drive 0).

Step 4: Prepare the Processor

Set the power-on action switch to HALT and the rotary key switch to LOCAL. It makes no difference how the boot device switch is set during installation.

Step 5: Load the copy Program

Halt the system and load the copy program from the distribution media into memory:

```
^P                          (CTRL/P to alert the console subsystem)
>>>H                        (halt the system)
      .
      .
      .
>>>I                        (initialize the system)
>>>B DDA0                   (boot the console TU58)
%%
=copy                       (the copy program starts)
```

Step 6: Copy the Minisystem to the System Disk

At this point the copy program is running from memory, and it asks for the necessary information to do the copy. The general form of the query and your response is:

From: *d(x,y)mini*
To: *s(z,1)*

where:

d — Is a two- or three-character ULTRIX device name for the distribution media. See Table 3-2 for a list of ULTRIX device names.

x — Is the logical unit number used by the copy program to access the distribution media.

y — Is 1 for tape distribution media, indicating the second file on the tape.

Is 2 for disk distribution media, indicating the c disk partition.

mini Is a file name that you type only if your distribution media is disk. (Do *not* type *mini* if your distribution media is a tape.)

s Is a two-character ULTRIX device name for the system disk. See Table 3-2 for a list of ULTRIX device names.

z Is the logical unit number used by the copy program to access the system disk.

1 Indicates that the minisystem is to be built in the b partition. Normally, the b partition of the system disk is the swap area. The minisystem occupies this space only temporarily, until the generic system is built.

You calculate the logical unit number as follows:

((adapter number) x 8) + (device unit number)

The *adapter number* is the number of the MASSBUS adapter (MBA) or UNIBUS adapter (UBA) to which the device is attached (0, 1, 2, or 3). To identify the adapter number to which your device is attached, check the Site Management Guide or contact your DIGITAL field service representative.

The *device unit number* is the physical device number on the front of the unit.

Note

> The CSR address for the load and system devices must be the standard VAX-11 CSR address, or the copy program will not recognize them.

Table 3-2: ULTRIX Device Names

DIGITAL Name	ULTRIX Name
MASSBUS Disks	
RM03	hp
RM05	hp
RM80	hp
RP05	hp
RP06	hp
RP07	hp
UNIBUS Disks	
RA60	ra
RA80	ra
RA81	ra
RK07	hk
MASSBUS Tapes	
TU45	ht
TE16	ht
TU77	ht
TU78	mt
UNIBUS Tapes	
TS05	ts
TS11	ts
TU80	ts
TU81	tms

Because the example installation uses a magnetic tape distribution media mounted on drive unit 0 (connected to UNIBUS adapter number 0), and a RK07 system disk mounted on drive unit 0 (connected to UNIBUS adapter number 0), the response is:

```
    From: ts(0,1)
    To: hk(0,1)
```

When the copy program is finished, it displays the number of blocks copied (300). Keep the system disk name and logical unit number handy. You need to specify them again in Steps 8 and 9.

Step 7: Load the boot Program

At this point, the copy program has copied the minisystem to the system disk and prompts for another copy. You want to stop the copy program, return to the console subsystem, and halt and initialize the system, to prepare to load the boot program:

```
From:^P                       (return to console subsystem)
>>>H                          (halt the system)
>>>I                          (initialize the system)
>>>B DDA0                     (load the boot program)
>>>%%
=boot
```

Step 8: Boot the Minisystem

The boot program prints this prompt:

```
    Boot
    :
```

You type the ULTRIX device name, logical unit number, and partition number for the system disk you copied to in Step 6, plus the file name vmunix:

*s(z.1)*vmunix

Again, *s* is the ULTRIX name for the system device, and *z* is the logical unit number, as discussed in Step 6. Because the example installation uses a RK07 system disk mounted on drive unit 0 (connected to UNIBUS adapter number 0), the response is:

```
Boot
:hk(0,1)vmunix
```

Step 9: Specify the System Disk

The boot program prints several lines of information about the equipment on the system, and then prompts:

```
root device?
```

You type the ULTRIX device name and logical unit number for the system disk you copied to in Step 6, *followed by an asterisk.* This is the name and logical unit number used by the minisystem to access the system disk. Because the example installation uses a RK07 system disk mounted on drive unit 0 (connected to UNIBUS adapter number 0), the response is:

```
root device? hk0*
```

The asterisk is very important, as it tells the minisystem not to swap. Thc normal swap arca, partition b of thc systcm disk, is wherc the minisystem was loaded.

The following line then appears:

```
erase ^?, kill ^U, intr ^C
```

At this point, a shell procedure takes control to present you with some questions, and to perform the installation of the generic system.

Step 10: Load the Generic System

During this step, you answer questions to define the load device, from which the procedure reads the generic system, and the system device, on which the procedure creates the root filesystem.

In this case, however, you specify the DIGITAL names for the devices. A list of possible load and system devices is displayed before each question.

Because the example installation uses a magnetic tape distribution media mounted on a TU80 tape drive, the response is:

```
List of recognized distribution media devices:
        te16 ts11 ts05 tu45 tu77 tu78 tu80 tu81
        ra60

What type of drive is the distribution media
mounted on [no default]: tu80
```

If your distribution media is disk, the script displays a message indicating which distribution disk logical unit number is supported by the installation. This message is not displayed if your distribution media is tape, because the script always assumes you are loading from physical tape unit 0 (on MASSBUS adapter number 0 or UNIBUS adapter number 0, if appropriate).

For disk distribution media, you are generally assumed to be loading from disk unit 1. However, the script asks whether you want to specify an alternate logical unit number. You should specify the same logical unit number you specified for the load device in Step 6. Therefore, in most cases your response to the alternate unit number question is n.

```
Unit 1 is the only supported unit number
for this device. These installation procedures
will use this unit number unless you specify an
alternate.

Do you wish to specify an alternate
unit number (y/n) [default: n]? n
```

The script then displays a similar set of questions to define the system disk, where the root file system is to be built. Because the example installation uses a RK07 system disk, the response is:

```
List of recognized disk drives to contain
root file system:
    ra60 ra80 ra81 rk07 rm03 rm05 rm80 rp05 rp06
    rp07

What type of drive is going to contain the root
filesystem [no default]: rk07
```

The script displays a message indicating the system disk logical unit number supported by the installation. However, the script asks whether you want to specify an alternate logical unit number. You should specify the same logical unit number you specified for the system device in Step 6. Therefore, in most cases your response to the alternate unit number question is n, as in the example installation:

```
    Unit 0 is the only supported unit number
    for this device.  These installation procedures
    will use this unit number unless you specify an
    alternate.

    Do you wish to specify an alternate
    unit number (y/n) [default: n]? n
```

The script then displays two messages and a question. The messages indicate the DIGITAL name and the corresponding ULTRIX device name and logical unit number for the distribution media and system disk. If these values are correct, answer y to the question following the messages, as in the example installation. Otherwise, answer n, and the script will prompt you again for the appropriate values.

```
Your distribution media is mounted on drive type tu80,
whose corresponding ULTRIX device name and logical unit
number are ts0.

Your system disk (the root device) is mounted on
drive type rk07, whose corresponding ULTRIX device
name and logical unit number are hk0.

Are these correct (y/n) [default: n]? y
```

After you respond with a y, the script displays some informational messages:

```
- - - Build root filesystem on hk0 - - -
       .
       .
       .
- - - Check the filesystem - - -
       .
       .
       .
- - - Restoring the dump image of the root - - -
       .
       .
       .
-*-*-* Root filesystem is now extracted onto -*-*-*
-*-*-* the destination disk filesystem -*-*-*
```

Step 11: Boot the Generic System

You are now ready to boot the generic system. The shell script displays a notice, and if your VAX-11/750 has the proper boot ROM, it asks you to type:

```
^P                          (access console subsystem)
>>>H                        (halt the processor)
>>>I                        (initialize the processor)
>>>B/3 DUA0                 (load the boot program)
```

If your VAX-11/750 does not have the proper boot ROM, the shell script asks you to type:

```
^P                          (access console subsystem)
>>>H                        (halt the processor)
>>>I                        (initialize the processor)
>>>B DDA0
=boot
```

At this point, the boot program prints its prompt, and you respond with the name of the system disk specified in step 10 and the file name vmunix. The second prompt is for the device where the root file system is to be placed (again, the system disk). In general, the sequence is:

```
    Boot
    :s(z,0)vmunix
              .
              .
              .
    (information display)
              .
              .
              .
    Root device? sz
```

Again, *s* is the ULTRIX name for the system device, and *z* is the logical unit number, as discussed in Step 6. The shell script displays the correct names and unit numbers before the sequence starts.

Because the example installation uses a RK07 system disk mounted on drive unit 0 (connected to UNIBUS adapter number 0), the response is:

```
Boot
: hk(0,0)vmunix
         .
         .
         .
(information display)
         .
         .
         .
Root device? hk0
```

This completes the steps for Phase 1 of the installation procedure. **Go to Phase 2.**

3.3 Bring Up The Preliminary Systems for VAX-11/780 and VAX-11/785

This section describes the steps you follow to bring up the two preliminary systems, the minisystem and the generic system, needed to generate your fully configured system.

Some steps require you to load and unload tapes, disks, and console diskettes. Other steps require you to answer questions and to make decisions about some of the questions you need to answer. The responses in color next to the question are those from an example installation. Use the example installation as a guide. The example installation has these characteristics:

- Processor: VAX-11/780
- Memory: 24 Mbytes
- System disk: RA81 mounted on drive unit 0, connected to UNIBUS adapter number 0)
- Distribution media: Magnetic tape mounted on TU77 tape drive unit 0 (connected to MASSBUS adapter number 1)

Your responses may differ from the ones in the example installation.

Step 1: Prepare the Load Device

Select the first volume of the distribution media. The labels for the first volume of the various media are as follows:

- Magnetic Tape. Label reads: ULTRIX-32 V1.2 BIN 16MT9 1/2.
- RA60 Disk Pack. Label reads: ULTRIX-32 V1.2 BIN RA60 1/1.

Mount the tape or disk on the load device, and prepare it for operation. If you do not know how to do this, see the User's Guide that was supplied with the device. A tape should be at beginning-of-tape (BOT) and on line. A disk should be spinning and have the Ready indicator lit.

Note

> You must select unit 0 as the load device if your distribution media is magnetic tape.

Step 2: Prepare the System Disk

Prepare the disk you have selected for your system disk. If it has removable packs, load a formatted disk pack into the drive. (DIGITAL disk packs are shipped preformatted.) Even if the pack is nonremovable, make sure that your system disk pack is spinning and that the Ready light is lit.

Step 3: Insert the Distribution Diskette

Insert the distribution diskette into the diskette drive inside the processor. The diskette label should point toward the handle of the diskette drive.

Step 4: Prepare the Processor

Set the auto restart switch to OFF and the rotary key switch to LOCAL.

Step 5: Load the copy Program

Halt the system and load the copy program from the distribution media into memory:

```
^P                      (CTRL/P to alert the console subsystem)
>>>H                    (halt the system)
>>>U                    (unjam the SBI)
>>>I                    (initialize the system)
>>>L COPY               (load the copy program)
>>>S 2                  (start the copy program)
```

Step 6: Copy the Minisystem to the System Disk

At this point the copy program is running from memory, and it asks for the necessary information to do the copy. The general form of the query and your response is:

From: *d(x,y)mini*
To: *s(z,1)*

where:

d — Is a two- or three-character ULTRIX device name for the distribution media. See Table 3-3 for a list of ULTRIX device names.

x — Is the logical unit number used by the copy program to access the distribution media.

y — Is 1 for tape distribution media, indicating the second file on the tape.

Is 2 for disk distribution media, indicating the c disk partition.

mini Is a file name that you type only if your distribution media is disk. (Do *not* type *mini* if your distribution media is a tape.)

s Is a two-character ULTRIX device name for the system disk. See Table 3-3 for a list of ULTRIX device names.

z Is the logical unit number used by the copy program to access the system disk.

1 Indicates that the minisystem is to be built in the b partition. Normally, the b partition of the system disk is the swap area. The minisystem occupies this space only temporarily, until the generic system is built.

You calculate the logical unit number as follows:

((adapter number) x 8) + (device unit number)

The *adapter number* is the number of the MASSBUS adapter or UNIBUS adapter to which the device is attached (0, 1, 2, or 3). To identify the adapter number to which your device is attached, check the Site Management Guide or contact your DIGITAL field service representative.

The *device unit number* is the physical device number on the front of the unit.

Note

> The CSR address for the load and system devices must be the standard VAX-11 CSR addresses, or the copy program will not recognize them.

Table 3-3: ULTRIX Device Names

DIGITAL Name	ULTRIX Name
MASSBUS Disks	
RM03	hp
RM05	hp
RM80	hp
RP05	hp
RP06	hp
RP07	hp
UNIBUS Disks	
RA60	ra
RA80	ra
RA81	ra
MASSBUS Tapes	
TU45	ht
TE16	ht
TU77	ht
TU78	mt
UNIBUS Tapes	
TS05	ts
TS11	ts
TU80	ts
TU81	tms

Because the example installation uses a distribution tape mounted on TU77 tape drive unit 0 (connected to MASSBUS adapter number 1), and an RA81 system disk mounted on drive unit 0 (connected to UNIBUS adapter number 0), the response is:

```
From: ht(8,1)
To: ra(0,1)
```

When the copy program is finished, it displays the number of blocks copied (300). Keep the system disk name and logical unit number handy. You need to specify them again in Steps 8 and 9.

Step 7: Load the boot Program

At this point, the copy program has copied the minisystem to the system disk and prompts for another copy. You want to stop the copy program, return to the console subsystem, and halt and initialize the system, to prepare to load the boot program:

```
From:^P                         (return to console subsystem)
>>>H                            (halt the system)
>>>I                            (initialize the system)
>>>B ANY
```

Step 8: Boot the Minisystem

The boot program prints this prompt:

```
    Boot
    :
```

You type the ULTRIX device name, logical unit number, and partition number for the system disk you copied to in Step 6, plus the file name vmunix:

*s(z,1)*vmunix

Again, *s* is the ULTRIX name for the system device, and *z* is the logical unit number, as discussed in Step 6.

Because the example installation uses a RA81 system disk mounted on drive unit 0 (connected to UNIBUS adapter number 0), the response is:

```
    Boot
    :ra(0,1)vmunix
```

Step 9: Specify the System Disk

The boot program prints several lines of information about the equipment on the system, and then prompts:

```
    root device?
```

You type the ULTRIX device name and logical unit number for the system disk you copied to in Step 6, *followed by an asterisk.* This is the name and logical unit number used by the minisystem to access the system disk. Because the example installation uses a RA81 system disk mounted on drive unit 0 (connected to UNIBUS Adapter Number 0), the response is:

```
root device? ra0*
```

The asterisk is very important, as it tells the minisystem not to swap. The normal swap area, partition b of the system disk, is where the minisystem was loaded.

The following line then appears:

```
erase ^?, kill ^U, intr ^C
```

At this point, a shell procedure takes control to present you with some questions, and to perform the installation of the generic system.

Step 10: Load the Generic System

During this step, you answer questions to define the load device, from which the procedure reads the generic system, and the system device, on which the procedure creates the root file system.

In this case, however, you specify the DIGITAL names for the devices. A list of possible load and system devices is displayed before each question.

Because the example installation uses a TU77 tape drive as the distribution load device, the response is:

```
List of recognized distribution media devices:
    te16 ts11 ts05 tu45 tu77 tu78 tu80 tu81
    ra60
What type of drive is the distribution media
mounted on [no default]: tu77
```

If your distribution media is disk, the script displays a message indicating which distribution disk logical unit number is supported by the installation. This message is not displayed if your distribution media is tape, because the script *always* assumes you are loading from physical tape unit 0 (on MASSBUS adapter number 0 or UNIBUS adapter number 0, if appropriate).

For disk distribution media, you are generally assumed to be loading from disk unit 0. However, the script asks whether you want to specify an alternate logical unit number. You should specify the same logical unit number you specified for the load device in Step 6. Therefore, in most cases your response to the alternate unit number question is n. The message is not shown here because the example installation is using magnetic tape as the distribution media.

The script then displays a similar set of questions to define the system disk, where the root file system is to be built. Because the example installation uses a RA81 system disk, the response is:

```
    List of recognized disk drives to contain
    root file system:
        ra60 ra80 ra81 rm03 rm05 rm80 rp05 rp06
        rp07

    What type of drive is going to contain the root
    filesystem [no default]: ra81
```

The script displays a message indicating the system disk logical unit number supported by the installation. However, the script asks whether you want to specify an alternate logical unit number. You should specify the same logical unit number you specified for the system device in Step 6. Therefore, in most cases your response to the alternate unit number question is n, as in the example installation:

```
    Unit 0 is the only supported unit number
    for this device.  These installation procedures
    will use this unit number unless you specify an
    alternate.

    Do you wish to specify an alternate
    unit number (y/n) [default: n]? n
```

The script then displays two messages and a question. The messages indicate the DIGITAL name and the corresponding ULTRIX device name and logical unit number for the distribution media and system disk. If these values are correct, answer y to the question following the messages, as in the example installation. Otherwise, answer n, and the script will prompt you again for the appropriate values.

```
Your distribution media is mounted on
drive type tu77, whose corresponding ULTRIX
device name and logical unit number are ht0.

Your system disk (the root device) is mounted on
drive type ra81, whose corresponding ULTRIX device
name and logical unit number are ra0.

Are these correct (y/n) [default: n]? y
```

After you respond with a y, the script displays some informational messages:

```
- - - Build root filesystem on ra0 - - -
          .
          .
          .
- - - Check the filesystem - - -
          .
          .
          .
- - - Restoring the dump image of the root - - -
          .
          .
          .
-*-*-* Root filesystem is now extracted onto -*-*-*
-*-*-* the destination disk filesystem -*-*-*
```

Step 11: Update the Console Diskette

The automated shell script displays the following instructions. After you perform the action, press the RETURN key to continue:

```
Please remove the distribution diskette from
the drive and store it in a safe place. Insert the
VAX-11/780/785 console diskette.

Press RETURN when you have done this or type 'skip'
to skip this step:
```

You may want to use the skip option if you have already updated the console diskette. For example, you may have completed this step, and are backtracking because you made an error later on in the installation procedure.

Otherwise, the shell script now reads the console diskette. Output from the make program is displayed, indicating the update is in process. The shell script then displays a message asking you to remove the console diskette and insert a blank or scratch diskette. This diskette will be overwritten, so the script asks you to confirm that the old information can be destroyed.

```
    Change diskette.  Hit return when done.

    Really initialize directory?
```

You must type yes before the shell script continues the update on the second diskette. Output from the make program is displayed, indicating the update is in process.

At the end of the console update, the script prompts:

```
Do you wish to make another console update (y/n) [n]: n
```

If you want another copy of the console media, change the diskette, and then type y, and the second copy will be made. Otherwise, type n, as in the example installation.

Step 12: Boot the Generic System

You are now ready to boot the generic system. The shell script displays a notice and asks you to type:

`^P`	(access console subsystem)
`>>>H`	(halt the processor)
`>>>U`	(unjam the SBI)
`>>>I`	(initialize the processor)
`>>>B ANY`	(load the boot program)

At this point, the boot program prints its prompt, and you respond with the name of the system disk that you placed the software on in Step 10 and the file name vmunix. The second prompt is for the device where the root filesystem is to be placed (again, the system disk). In general, the sequence is:

```
    Boot
    :s(z,0)vmunix
              .
              .
              .
    (information display)
              .
              .
              .
    Root device? sz
```

Again, *s* is the ULTRIX name for the device and *z* is the logical unit number, as discussed in Step 6. The shell script displays the correct names and unit numbers before the sequence starts.

Because the example installation uses a RA81 system disk mounted on drive unit 0 (connected to UNIBUS adapter number 0), the response is:

```
Boot
: ra(0,0)vmunix
         .
         .
         .
(information display)
         .
         .
         .
Root device? ra0
```

This completes the steps for Phase 1 of the installation procedure.

Go to Phase 2.

3.4 Bring Up The Preliminary Systems for VAX 8600

This section describes the steps you follow to bring up the two preliminary systems, the minisystem and the generic system, needed to generate your fully configured system.

Some steps require you to load and unload tapes and disks. Other steps require you to answer questions and to make decisions about some of the questions you need to answer. The responses in color next to the question are those from an example installation. Use the example installation as a guide. The example installation has these characteristics:

- Processor: VAX 8600
- Memory: 32Mbytes
- System disk: RA60 mounted on drive unit 1, connected to UNIBUS adapter number 0,connected to SBI 0
- Distribution media: Magnetic tape mounted on TU81 tape drive unit 0, connected to UNIBUS adapter number 4, connected to SBI 1

Your responses may differ from the ones in the example installation.

Step 1: Prepare the Load Device

Select the first volume of the distribution media, which on a VAX 8600 is magnetic tape. The label for the first volume of the magnetic tape distribution media is: ULTRIX-32 V1.2 BIN 16MT9 1/2. Mount the tape on the load device, and prepare it for operation. (Remove the write-enable ring before mounting the tape.) If you do not know how to mount a tape, see the User's Guide that was supplied with the device. The tape should be at beginning-of-tape (BOT) and on line.

Step 2: Prepare the System Disk

Prepare the disk you have selected for your system disk. If it has removable packs, load a formatted disk pack into the drive. (DIGITAL disk packs are shipped preformatted.) Even if the pack is nonremovable, make sure that your system disk pack is spinning and that the Ready light is lit.

Step 3: Mount the RL02 Disk

Mount the RL02 disk (if not already mounted) into the disk drive located in the VAX 8600 processor. You should have received this RL02 disk with your hardware. Make sure the READY light is lit and that the console subsystem is running. The console subsystem is running if three angle brackets (>>>) appear on your console terminal. If three angle brackets do not appear on your console terminal, press the RETURN key. The angle brackets should then appear.

Step 4: Prepare the Processor

Set the power-on action switch to HALT and the rotary key switch to LOCAL.

Step 5: Load the copy Program

Halt the system and load the copy program from the distribution media into memory:

```
^P                        (CTRL/P to alert the console subsystem)
>>>H                      (halt the system)
    .
    .
    .
>>>B COPY                 (load the copy program)
```

Step 6: Copy the Minisystem to the System Disk

At this point the copy program is running from memory, and it asks for the necessary information to do the copy. The general form of the query and your response is:

From: *d(x,y)*
To: *s(z,1)*

where:

d Is a two- or three-character ULTRIX device name for the distribution media. See Table 3-4 for a list of ULTRIX device names.

x Is the logical unit number used by the copy program to access the distribution media.

y Is 1, indicating the second file on the distribution tape.

s Is a two-character ULTRIX device name for the system disk. See Table 3-4 for a list of ULTRIX device names.

z Is the logical unit number used by the copy program to access the system disk.

1 Indicates that the minisystem is to be built in the b partition. Normally, the b partition of the system disk is the swap area. The minisystem occupies this space only temporarily, until the generic system is built.

Note

The CSR address for the load and system devices must be the standard VAX-11 CSR address, or the copy program will not recognize them.

Table 3-4: ULTRIX Device Names

DIGITAL Name	ULTRIX Name
MASSBUS Disks	
RM03	hp
RM05	hp
RM80	hp
RP05	hp
RP06	hp
RP07	hp
UNIBUS Disks	
RA60	ra
RA80	ra
RA81	ra
MASSBUS Tapes	
TU45	ht
TE16	ht
TU77	ht
TU78	mt
UNIBUS Tapes	
TS05	ts
TS11	ts
TU80	ts
TU81	tms

To figure out your logical unit number, you need to know the:

- adapter number. – The number of the MASSBUS or UNIBUS adapter to which the device is attached. This number ranges from 0 through 7.
- device unit number. – The physical device number on the front of the unit. This number can range from 0 through 7. For tape drives, the device unit number is the TM03 or TM78 controller unit number, not the slave unit number (which must be 0).

To identify the adapter number to which your device is attached, check the Site Management Guide (or your DIGITAL field service representative) for the:

- TR level. – The TR level is the number associated with a specific SBI address, on which the MASSBUS or UNIBUS adapter attaches to the Synchronous Backplane Interconnect (SBI). The SBI is the hardware that interconnects the processor, MASSBUS adapters, and UNIBUS adapters. The VAX 8600 supports a maximum of two SBIs (SBI 0 and SBI 1). Table 3-5 lists the TR levels, their associated SBI addresses, and their associated UNIBUS and MASSBUS adapter numbers. So, for example, if the TR level for SBI 0 in the Site Management Guide is 5, then the UNIBUS adapter number is 2.

OR

- SBI address. – The SBI address is the location of the SBI. Table 3-5 lists the SBI addresses and their associated UNIBUS and MASSBUS adapter numbers. So, for example, if the SBI address for SBI 1 in the Site Management Guide is 22006000, then the UNIBUS adapter number is 4.

Table 3-5: Adapter Numbers for VAX 8600

TR Level	SBI 0 Address	ULTRIX UNIBUS Adapter Number	TR Level	SBI 0 Address	ULTRIX MASSBUS Adapter Number
3	20006000	0	8	20010000	0
4	20008000	1	9	20012000	1
5	2000A000	2	10	20014000	2
6	2000C000	3	11	20016000	3

TR Level	SBI 1 Address	ULTRIX UNIBUS Adapter Number	TR Level	SBI 1 Address	ULTRIX MASSBUS Adapter Number
3	22006000	4	8	22010000	4
4	22008000	5	9	22012000	5
5	2200A000	6	10	22014000	6
6	2200C000	7	11	22016000	7

To obtain the logical unit numbers for the load device and the system device, follow these steps:

1. Obtain the TR number from the Site Management Guide or your DIGITAL field service representative. Then use Table 3-5 as a guide to obtaining your adapter number.
2. Obtain the device unit number.
3. Plug the adapter number and the device unit number into this formula to produce the logical unit number:

   ((adapter number) x 8) + (device unit number)

Thus, the logical unit number for RM05 disk unit 7 connected to MASSBUS adapter number 3 would be 31.

Because the example installation uses a distribution tape mounted on TU81 tape drive unit 0 (connected to UNIBUS adapter number 4), and a RA60 system disk mounted on drive unit 1 (connected to UNIBUS adapter number 0), the response is:

```
    From: tms(32,1)
    To: ra(1,1)
```

When the copy program is finished, it displays the number of blocks copied (300). Keep the system disk name and logical unit number handy. You need to specify them again in Steps 8 and 9. You also need to specify this logical unit number in Step 11.

Step 7: Load the boot Program

At this point, the copy program has copied the minisystem to the system disk and prompts for another copy. You want to stop the copy program, return to the console subsystem, and halt and initialize the system, to prepare to load the boot program:

```
From:^P                      (return to console subsystem)
>>>H                         (halt the system)
>>>B ANY
```

Step 8: Boot the Minisystem

The boot program prints this prompt:

```
    Boot
    :
```

You type the ULTRIX device name, logical unit number, and partition number for the system disk you copied to in Step 6, plus the file name vmunix:

*s(z,1)*vmunix

Again, *s* is the ULTRIX name for the system device, and *z* is the logical unit number, as discussed in Step 6.

Because the example installation uses a RA60 system disk mounted on drive unit 1 (connected to UNIBUS adapter number 0), the response is:

```
    Boot
    :ra(1,1)vmunix
```

Step 9: Specify the System Disk

The boot program prints some lines of information about the equipment on the system. You need to refer to this list in order to answer the question that follows the equipment list. In the example installation, the equipment list looks like this:

```
250304+91548+91760 start 0x1678
Ultrix X1.12 System #1: Mon Jul 1 16:29:16 EDT 1985
real mem = 33550336
avail mem = 31157248
using 135 buffers containing 1105920 bytes of memory
VAX 8600, serial no. 172, hardware level = 5
IO adapter 0 at address 0x20080000 is an SBI adapter
uba0 at address 0x20006000
uda0 at uba0 csr 172150 vec 774, ipl 15
ra0 at uda0 slave 0
ra1 at uda0 slave 1
     .
     .
     .
root device?
```

You respond to the root device? question with the ULTRIX device name and logical unit number for the system disk you copied to in Step 6, *followed by an asterisk*. This is the name and logical unit number used by the minisystem to access the system disk. To obtain the ULTRIX device name and logical unit number follow these steps (if your system disk is mounted on a MASSBUS device skip steps 1 and 2):

1. Check the equipment list for the UNIBUS adapter number you obtained in Step 6. In the example installation, this number is *0*, indicated by uba0.

2. Identify the uda controller attached to the drive your system disk is mounted on. See the Site Management Guide, or contact your DIGITAL field service representative.

   In the example installation, uda0 is the controller that attaches to the drive the system disk is mounted on.

3. Check the equipment list for the word slave for UNIBUS devices, or the word drive for MASSBUS devices. The number to the right of slave or drive is the physical unit number.

   In the example installation, there are two physical unit numbers: slave 0 and slave 1. Slave 1 is the physical unit number of interest, because the RA60 system disk is mounted on drive unit 1, which is connected to UNIBUS adapter number 0.

4. Obtain the ULTRIX device name and logical unit number associated with the physical unit number. Thus, in the example installation, the ULTRIX device name is ra and the logical unit number is 1.

5. Type the ULTRIX device name and the logical unit number, followed by an asterisk. Thus, in the example installation, the response is:

   ```
   root device? ra1*
   ```

   The asterisk is very important, as it tells the minisystem not to swap. The normal swap area, partition b of the system disk, is where the minisystem was loaded. Keep this logical unit number handy. You need to specify it again in Step 10.

The following line then appears:

```
erase ^?, kill ^U, intr ^C
```

At this point, a shell procedure takes control to present you with some questions, and to perform the installation of the generic system.

Step 10: Load the Generic System

During this step, you answer questions to define the load device, from which the procedure reads the generic system, and the system device, on which the procedure creates the root filesystem.

In this case, however, you specify the DIGITAL names for the devices. A list of possible load and system devices is displayed before each question.

Because the example installation uses a magnetic tape distribution media mounted on TU81 tape drive unit 0, the response is:

```
List of recognized distribution media devices:
    te16 ts11 ts05 tu45 tu77 tu78 tu80 tu81

What type of drive is the distribution media
mounted on [no default]: tu81
```

The script then displays a similar set of questions to define the system disk, where the root file system is to be built. Because the example installation uses a RA60 system disk, the response is:

```
List of recognized disk drives to contain
root file system:
    ra60 ra80 ra81 rm03 rm05 rm80 rp05 rp06
    rp07

What type of drive is going to contain the root
filesystem [no default]: ra60
```

The script displays a message indicating which system disk logical unit number is supported by the installation. The script asks whether you want to specify an alternate logical unit number. You should specify the same logical unit number you specified for the system device in Step 9. Therefore, in most cases your response to the alternate unit number question is n.

In the example installation, the logical unit number of the system disk is 1 and not 0. Therefore, the answer to the alternate logical unit number question is y, and the logical unit number specified is 1. Keep the logical unit number handy. You need to specify it again in Step 11:

```
Unit 0 is the suggested unit number
for this device.  These installation procedures
will use this unit number unless you specify an
alternate.

Do you wish to specify an alternate
unit number (y/n) [default: n]? y

What is the unit number for the drive [no default]: 1

You specified a unit number of 1
Is this correct (y/n)? y
```

The script then displays two messages and a question. The messages indicate the DIGITAL name and the corresponding ULTRIX device name and logical unit number for the distribution media and system disk. If these values are correct, answer y to the question following the messages, as in the example installation. Otherwise, answer n, and the script will prompt you again for the appropriate values:

```
Your distribution media is mounted on
drive type tu81, whose corresponding
ULTRIX device name and logical unit
number are tms32.

Your system disk (the root device) is mounted on
drive type ra60, whose corresponding ULTRIX device
name and logical unit number are ra1.

Are these correct (y/n) [default: n]? y
```

After you respond with a y, the script displays some informational messages:

```
- - - Build root filesystem on ra1 - - -
      .
      .
      .
- - - Check the filesystem - - -
      .
      .
      .
- - - Restoring the dump image of the root - - -
      .
      .
      .
-*-*-* Root filesystem is now extracted onto -*-*-*
-*-*-* the destination disk filesystem -*-*-*
```

Step 11: Update the RL02 Disk

The automated shell script displays the following messages:

```
    Preparing to update the RL02 console disk.
    Press RETURN, or type 'skip' to skip this step:
```

You may want to use the skip option if you have already updated the RL02 disk. For example, you may have completed this step, and are backtracking because you made an error later on in the installation procedure.

Otherwise, press the RETURN key and the shell script displays:

```
    Enter the logical unit number of the
    root device. This is the same number
    you enter to boot the system [no default]: 1
```

Type the logical unit number of the device on which your system disk is mounted. This logical unit number should be the same one you entered in Step 6. In the example installation, the response is 1. The script now reads the RL02 disk.

Output from the make program is displayed, indicating the update is in process.

Step 12: Boot the Generic System

You are now ready to boot the generic system. The shell script displays a notice and asks you to type:

```
^P                          (access console subsystem)
>>>H                        (halt the processor)
>>>B ANY                    (load the boot program)
```

At this point, the boot program prints its prompt, and you respond with the name of the system disk that you placed the software on in step 10 and the file name vmunix. This is the name and logical unit number you specified in Step 6. The second prompt is for the device where the root filesystem is to be placed (again, the system disk). In this case, however, this is the name and logical unit number you specified in Step 9. The shell script displays the correct names and unit numbers before the sequence starts.

Because the example installation uses a RA60 system disk mounted on drive unit 1 (connected to UNIBUS adapter number 0), the responses are:

```
Boot
: ra(1,0)vmunix
          .
          .
          .
(information display)
          .
          .
          .
Root device ra1
```

This completes the steps for Phase 1 of the installation procedure.

Go to Phase 2.

3.5 Bring Up The Preliminary Systems for VAX 8200

This section describes the steps you follow to bring up the two preliminary systems (the minisystem and the generic system) needed to generate your fully configured system.

Some steps require you to load and unload tapes and disks. Other steps require you to answer questions and make decisions. The responses in color next to the question are those from an example installation. Use the example installation as a guide. The example installation has these characteristics:

- Processor: VAX 8200 at VAXBI node 3
- Memory: 8 Mbytes at VAXBI nodes 4, 6, 7, and 8
- System disk: RA81 mounted on drive unit 0, connected to KDB50 at BI node 5
- Distribution media: Magnetic tape mounted on TU81 tape drive unit 0, connected to DWBUA VAXBI node 0

Your responses may differ from the ones in the example installation.

Step 1: Display the VAXBI Configuration Matrix

The VAXBI Configuration Matrix is a table that supplies you with information about your VAX 8200 configuration. Before you start the installation, use the EEPROM Utility to display the VAXBI Configuration Matrix for your system. See the Owner's Manual supplied with the VAX 8200 processor for instructions on how to use the EEPROM Utility to display the VAXBI Configuration Matrix.

The following shows the VAXBI Configuration Matrix for the example installation:

```
NODE  REVISION       DEVICE  &   TYPE
      CODE           CODE
---------------------------------------------
0     0000           0102    DWBUA
3     1002           0105    KA820  CPU
4     0001           0001    MS820  MEMORY
5     0000           010E    KDB50
6     0001           0001    MS820  MEMORY
7     0001           0001    MS820  MEMORY
8     0001           0001    MS820  MEMORY
9     0001           0001    KDB50
```

Keep the VAXBI Configuration Matrix handy; subsequent steps require you to refer to it.

Step 2: Prepare the Load Device

Select the first volume of the distribution media, which for a VAX 8200 is magnetic tape. The label for the first volume of the magnetic tape distribution media is: ULTRIX-32 V1.2 BIN 16MT9 1/2.

Mount the tape on the load device, and prepare it for operation. (Remove the write-enable ring before mounting the tape.) If you do not know how to mount a tape, see the User's Guide that was supplied with the device. The tape should be at beginning-of-tape (BOT) and on line.

Step 3: Prepare the System Disk

You must now load your system disk onto the disk drive. Use the VAXBI Configuration Matrix you displayed in Step 1 to determine which disk drive to use. Check the column in the VAXBI Configuration Matrix labelled "TYPE." Follow these guidelines:

- If this column contains only a DWBUA with a UDA50 controller, then load the system disk on the first disk drive connected to the UDA50 controller.
- If the column contains one or more KDB50 controllers, then load the system disk on the first disk drive connected to the KDB50 with the lowest VAXBI node number.

In the example installation, the TYPE column shows:

- One DWBUA with a UDA50 controller at BI node 0
- Two KDB50 controllers: one at BI node 5, and the other at BI node 9

Following the previous guidelines, the system disk in the example installation is loaded on the disk drive whose physical unit number is 0. The reason for this is that the disk drive whose physical unit number is 0 is the first drive connected to the KDB50 controller at BI node 5.

Make sure that your system disk pack is spinning and that the Ready light is lit.

Step 4: Insert the Distribution Diskette

Insert the distribution diskette into diskette drive 0.

Step 5: Prepare the Processor

Set the key switch 1 to ENABLE; set the key switch 2 to REBOOT.

Step 6: Load the copy Program

Halt the system and load the copy program from the distribution media into memory:

```
^P                              (CTRL/P to alert the console subsystem)
>>>I                            (initialize the system)
>>>B CSA1
Boot
:cs(1,0)copy
```

Step 7: Copy the Minisystem to the System Disk

At this point, the copy program is running from memory, and it asks for the necessary information to do the copy. The general form of the query and your response is:

From: *d(x,1)*
To: *ra(z,1)*

where:

d Is a two- or three-character ULTRIX device name for the distribution media. See Table 3-6 for a list of ULTRIX device names.

x Is the logical unit number used by the console subsystem to access the distribution media.

1 Indicates the second file on the tape.

z Is the physical unit number of the disk drive on which the system disk is mounted.

1 Indicates that the minisystem is to be built in the b partition. Normally, the b partition of the system disk is the swap area. The minisystem occupies this space only temporarily, until the generic system is built.

You calculate the logical unit number for the distribution device as follows:

(adapter number) x 8 + (device unit number)

The *adapter number* is the number of the DWBUA or KDB50 to which the device is attached. To determine the adapter number follow these steps:

1. Locate the controller at the lowest VAXBI node number and assign to it adapter number 0. In the example installation, the DWBUA controller is at the lowest VAXBI node number; therefore, its adapter number is 0.
2. Locate subsequent controllers at the next higher VAXBI node numbers, and assign to them adapter numbers 1, 2, and so forth. (Do not count VAX 8200 processor or MS820 memory.) In the example installation, there are two additional controllers: a KDB50 at VAXBI node 5, and a KDB50 at VAXBI node 9. The KDB50 at VAXBI node 5 is assigned an adapter number of 1; the KDB50 at VAXBI node 9 is assigned an adapter number of 2.
3. Use the adapter number for the controller that connects your distribution device. In the example installation, the distribution device is connected to the DWBUA at VAXBI node 0; therefore, the adapter number is 0.

The *device unit number* is the physical device number on the front of the unit.

Because the example installation uses a magnetic tape distribution media mounted on drive unit 0 (connected to DWBUA adapter number 0 at BI node 0), the logical unit number is 0:

$$0 \times 8 + 0 = 0$$

To calculate the logical unit number of the system device, simply use the physical unit number on which the system disk resides. Thus, in the example installation the logical unit number of the system device is 0, because the system disk is mounted on physical drive 0.

Note

> The CSR address for the load and system devices must be the standard VAX-11 CSR addresses, or the console subsystem will not recognize them.

Table 3-6: ULTRIX Device Names

DIGITAL Name	ULTRIX Name
UNIBUS or BI Disks	
RA60	ra
RA80	ra
RA81	ra
UNIBUS Tapes	
TS05	ts
TS11	ts
TU80	ts
TU81	tms

Thus, because the example installation uses a magnetic tape distribution media mounted on drive unit 0 (connected to DWBUA adapter number 0 at BI node 0), and a RA81 system disk mounted on drive unit 0 (connected to a KDB50 controller at BI node 5), the response is:

```
    From: tms(0,1)
    To: ra(0,1)
```

When the copy program is finished, it displays the number of blocks copied (300). Keep the system disk name and physical unit number handy. You need to specify them again in Steps 9 and 10.

Step 8: Load the Boot Program

At this point, the copy program has copied the minisystem to the system disk and prompts for another copy. You want to stop the copy program, return to the console subsystem, and initialize the system, to prepare to load the boot program:

```
From:^P                      (return to console subsystem)
>>>I                         (initialize the system)
>>>B CSA1
```

Step 9: Boot the Minisystem

The boot program prints this prompt:

```
Boot
:
```

You type the ULTRIX device name, physical unit number, and partition number for the system disk you copied to in Step 7, plus the file name vmunix:

*ra(z,1)*vmunix

Again, *z* is the physical unit number for the system device, as discussed in Step 7.

Because the example installation uses a RA81 system disk mounted on drive unit 0 (connected to a KDB50 controller at BI node 5), the response is:

```
Boot
:ra(0,1)vmunix
```

Step 10: Specify the System Disk

The boot program prints several lines of information about the equipment on the system, and then prompts:

```
root device?
```

You type the ULTRIX device name and physical unit number for the system disk you copied to in Step 7, *followed by an asterisk.* This is the name and physical unit number used by the minisystem to access the system disk. Because the example installation uses a RA81 system disk mounted on drive unit 0 (connected to a KDB50 controller at BI node 5) the response is:

```
root device? ra0*
```

The asterisk is very important: it tells the minisystem not to swap. The minisystem was loaded in partition b, which is the usual swap area on the system disk.

The following line then appears:

```
    erase ^?, kill ^U, intr ^C
```

At this point, a shell procedure takes control to present you with some questions, and to perform the installation of the generic system.

Step 11: Load the Generic System

During this step, you answer questions to define both the load device, from which the procedure reads the generic system, and the system device, on which the procedure creates the root file system.

In this case, however, you specify the DIGITAL names for the devices. A list of possible load and system devices is displayed before each question.

Because the example installation uses a magnetic tape distribution media mounted on a TU81 tape drive, the response is:

```
    List of recognized distribution media devices:
        ts11 ts05 tu80 tu81

    What type of drive is the distribution media
    mounted on [no default]: tu81
```

The script then displays a similar set of questions to define the system disk, which is where the root file system is to be built. Because the example installation uses a RA81 system disk, the response is:

```
    List of recognized disk drives to contain
    root file system:
        ra60 ra80 ra81

    What type of drive is going to contain the root
    filesystem [no default]: ra81
```

The script displays a message indicating which system disk physical unit number is supported by the installation. However, the script asks whether you want to specify an alternate physical unit number. You should specify the same physical unit number you specified for the system device in Step 7. Therefore, in most cases your response to the alternate unit number question is n, as in the example installation.

```
    Unit 0 is the only supported unit number
    for this device.  These installation procedures
    will use this unit number unless you specify an
    alternate.

    Do you wish to specify an alternate
    unit number (y/n) [default: n]? n
```

The script then displays two messages and a question. The messages indicate the DIGITAL name and the corresponding ULTRIX device name and logical unit number for the distribution media, and the DIGITAL name and the corresponding ULTRIX device name and physical unit number for the system disk. If these values are correct, answer y to the question following the messages, as in the example installation. Otherwise, answer n, and the script will you prompt you again for the appropriate values.

```
    Your distribution media is mounted on
    drive type tu81, whose
    corresponding ULTRIX device name and
    logical unit number are
    tms0.

    Your system disk (the root device) is mounted on
    drive type ra81, whose corresponding
    ULTRIX device name and physical unit number are
    ra0.

    Are these correct (y/n) [default: n]? y
```

After you respond with a y to the previous question, the script displays some informational messages:

```
- - - Build root filesystem on ra0 - - -
      .
      .
      .
- - - Check the filesystem - - -
      .
      .
      .
- - - Restoring the dump image of the root - - -
      .
      .
      .
```

(continued on next page)

```
-*-*-* Root filesystem is now extracted onto -*-*-*
-*-*-* the destination disk filesystem -*-*-*
```

Step 12: Boot the Generic System

You are now ready to boot the generic system. The shell script displays a notice and asks you to type:

```
^P                      (access console subsystem)
>>>I                    (initialize the processor)
>>>B duxy/R5:3          (load the boot program)
```

where:

x Is the BI node associated with the controller that connects to the drive on which the system disk is loaded. You obtain this BI node number from the the VAXBI Configuration Matrix displayed in Step 1.

In the example installation, *x* is 5, because the KDB50 controller at BI node 5 connects to the drive on which the RA81 system disk is loaded.

y Is the physical unit number of the drive on which the system disk is mounted.

In the example installation, *y* is 0 (because the physical unit number of the drive is 0).

Thus, in the example installation, the response is:

```
^P                      (access console subsystem)
>>>I                    (initialize the processor)
>>>B du50/R5:3          (load the boot program)
```

At this point, the boot program prints its prompt, and you respond with the name of the load device (in this case, the system disk that you placed the software on in Step 11) and the file name vmunix. The second prompt is for the device where the root file system is to be placed (again, the system disk). In general, the sequence is:

Boot
:*ra(z,0)*vmunix

.
.
.

(information display)

.
.
.
Root device? *raz*

Again, *z* is the physical unit number for the system device, as discussed in Step 7. The shell script displays the correct names and unit numbers before the sequence starts.

Because the example installation uses a RA81 system disk mounted on drive unit 0 (connected to KDB50 adapter number 1 at BI node 5) the response is:

```
Boot
: ra(0,0)vmunix
              .
              .
              .
(information display)
              .
              .
              .
Root device? ra0
```

This completes the steps for Phase 1 of the installation procedure.

Go to Phase 2.

Phase 2: Bring Up the Full System 4

During this phase of the installation process, a shell script takes control to lead you through the process of defining and generating a system tailored specifically to your hardware configuration.

Step 1: Setting the System Date and Time

Here you set the correct date and time. The system maintains a date and time for various displays; the time you set here is used by the system to maintain the current date and time. The shell script displays a prompt for you to enter the date in the format:

```
yymmddhhmm
```

Two digits are accepted for year (yy), month (mm), day (dd), hour using a 24-hour clock (hh), and minute (mm). For example, for 11:30 p.m. on April 1, 1985, the setting would be:

```
8504012330
```

If you use alphabetic characters, or some other inappropriate response, the script will keep prompting for an acceptable date and time.

Step 2: Setting the Superuser Password

Up to this point, no superuser password has been established. You must set the superuser password, using whatever characters you choose. After this point, anyone who logs in with the superuser name (root) must use the password you chose.

The superuser has special privileges, so choose your superuser password carefully (do not use your birthday, for example) and keep it secure.

The shell script prompts for the new password and asks you to retype it, to make sure you typed what you thought you did. If you make an error, the script starts over again.

The password you choose must be at least four characters long if you use both uppercase and lowercase letters. It must be at least six characters long if you use either all uppercase or all lowercase letters.

The following example illustrates a possible sequence. As a security measure, the passwords typed are not actually displayed on the terminal:

```
*** Changing the Super User password ***

New password: Puffnerdle
Retype new password: Puffnurdle
Mismatch - password unchanged

Trying again to set Super User password. . .
New password: Puffnerdle
Retype new password: Puffnerdle
```

Step 3: Setting the Field Service Password

Up to this point, there is no password for the field service account. You must set the field service password, using whatever characters you choose. After this point, anyone who logs in with the field service name (field) must use the password you choose here.

The field service account has special privileges (the same as the superuser), so choose the field service password carefully and keep it secure.

The shell script prompts for the new password and asks you to retype it, to make sure you typed what you thought you did. If you make an error, the script starts over again.

The password you choose must be at least four characters long if you use both uppercase and lowercase letters. It must be at least six characters long if you use either all uppercase or all lowercase letters.

The following example illustrates a possible sequence. As a security measure, the passwords typed are not actually displayed on the terminal:

```
*** Changing the Field Service password ***

New password: pilictony
Retype new password: pilictony
```

Note

To ensure the security of your system, do not use the passwords given in the previous examples.

Step 4: Continue Loading the ULTRIX-32 Software

At this point, the shell script continues loading software from the distribution media, displaying informative messages as it goes. In the display, <sysdisk> is used to show you where the script displays the disk name, unit, and disk partition where it is putting the /usr directory. The first message of special interest to you is shown below:

```
*** Making and Extracting the /usr filesystem ***

Creating the /usr filesystem in <sysdisk>.
Save the super-block numbers generated here.  Tape
them to the outside of the disk pack cover.  If
your pack is nonremovable, tape them to the
outside of the drive.  They can be very useful in
certain emergencies.

- - - - - - - - - - - - - - - - - - - - - - - -

*** SUPER BLOCK NUMBERS FOR /USR IN FS <sysdisk> ***
    .
    .
    .
(Superblock numbers useful to Digital Software Service
 in reconstructing disks in case of a head crash,
 for example.)
    .
    .
    .
- - - - - - - - - - - - - - - - - - - - - - - - -
```

The shell script then starts loading the software for the /usr directory (such as /usr/bin, /usr/lib, and so forth). The software for /usr/sys is loaded in a separate step. When all /usr software is loaded, the shell script checks the consistency of the /usr file system, using the fsck(8) program. It then uses mount(8) to mount the directory on the disk partition selected for it, creating the /usr file system.

```
Extracting /usr (except /usr/sys).  This
takes about 5-10 minutes
Finished
Extracting system source.  This
takes about 5-10 minutes.
Finished
Checking the consistency of the /usr filesystem
on <sysdisk>.
```

This finishes step 4, unless your system disk is an RK07 disk. The shell script will go on to Step 5 unless your system disk is an RK07. There is not enough room on the RK07 disk for all of the system software. You must copy the /usr/sys software from the system disk to another disk that will also contain the /usr/users directory.

Note

> Before you copy /usr/sys to another disk, see the ULTRIX-32 System Manager's Guide for instructions on how to prepare the disk to accept filesystems.

For an RK07 system disk (unit 0), the /usr/sys software is copied to the RK07 disk assigned to unit 1. The shell script will display a paragraph telling you to load a disk pack on unit 1 and spin it up. The script will wait until you press the RETURN key to continue.

The shell script then copies /usr/sys to the second disk, displaying informative messages as it executes.

Step 5: Creating the /usr/users File System

At this point, the shell script displays a message telling you to dismount the current volume of the distribution media and store it in a safe place. The shell script automatically proceeds with creating a file system called /usr/users. In most cases, you will add user accounts to this directory after the installation is completed.

The shell script assigns this directory to the h partition of the system disk if it has one. If there is no h partition, /usr/users is assigned to the g partition (where the rest of the /usr directory resides), unless the system disk is an RK07 disk.

For an RK07 system disk, the script assigns the /usr/users directory to the g partition of the RK07 disk assigned to physical unit 1.

The shell script then displays a prompt asking if you want to change the name of the /usr/users file system. To accept the default response of /usr/users simply press the RETURN key. Otherwise enter another name to identify the /usr/users file system. For example:

```
What would you like the name of the user
accounts directory to be [/usr/users]: /usr/staff
```

The script responds with a message showing the response you just specified, and a question asking if what you entered was correct:

```
    You specified /usr/staff as the directory name.
    Is this correct (y/n) [n]? y
```

The shell script now displays a series of messages on what it is doing. The final message is:

```
    * * * Listing Free Space on all Filesystems
```

This message is followed by a listing of the disk partitions on your system. For each partition, the total space available, the space used, and the space free is given in kilobytes. This is followed by a percentage telling how much space has been used on the disk partition, and the names of the directories that the disk partition has been mounted on.

Step 6: Enabling or Disabling Crash Dumps

The shell script next asks you if you want to enable crash dumps:

```
    * * * Automatic Crash Dump Query * * *
    Would you like to enable automatic crash
    dumps (y/n) [y]?
```

If you enter y, the contents of memory will be written to disk, in the partitions designated as the paging and swapping area, whenever the system crashes. This memory dump can be helpful in figuring out what caused the crash.

Step 7: Defining the Hardware Configuration

You now define the hardware configuration for the system. The shell script asks you for the name you are giving to the system. If you are going to use the uucp network, then the name you enter must be unique up to the first seven characters.

```
* * * System Configuration Dialogue * * *
Enter the name of your system (all capitals) [no default]:
You entered <sysname>.  Is this
correct (y/n) [y]?

Should the running system have this name?
(i.e., should rc.local be changed to
make <sysname> the hostname? (y/n) [y]:
```

You should enter y to this question, since your running system has no name until the first time this is done.

The shell script then asks you for information on the local time zone:

```
Enter the Time Zone for your area.  Choices for
the time zone are:
    Eastern              (e)
    Central              (c)
    Mountain             (m)
    Pacific              (p)
    Greenwich            (g)

 or number of hours (in time) west of Greenwich.

Enter the Time Zone (e, c, m, p, g, # of hours):
```

The script then asks:

```
Does your area alternate between Daylight Savings
and Standard time [yes] ?
```

If you enter y, the script displays:

```
Enter Geographic area for Daylight Savings Time.
Choices are:
     USA                 (u)
     Australia           (a)
     Eastern Europe      (e)
     Central Europe      (c)
     Western Europe      (w)

Enter the Geographic area (u, a, e, c, w) [u]:
```

The shell script polls the system to find out what hardware has already been installed, and creates a configuration file reflecting the hardware it has found. This configuration file is used to determine which drivers are needed to run the hardware on your system.

When it has finished creating the configuration file, the shell script displays it for you:

```
A configuration file has been created for the
devices this program has determined are present.
The configuration file is as follows:
```

(configuration file)

You should be able to tell if your hardware configuration is correctly presented. For details on the format of a configuration file, see Appendix B.

The shell script does not prepare any information in the configuration file for communications devices (such as terminal controllers, for example). So, the script prompts you for additional information, beginning as follows:

```
You need to add a communication device to the
configuration file.  You may also add other
devices that were not found by this program.


You will be asked to enter information about the
device(s) you wish to add to the configuration file.
Entering 'exit' in response to the first prompt will
terminate this portion of the configuration process.
Typing 'erase' will delete previously entered data for
the current device and return to the first prompt.
If the device(s) you choose are supported by Digital,
the name of interrupt routines will be supplied by
the program.
```

The first time through the questions, a few lines of explanatory text are displayed before each question, to help you supply the correct answer. After the first time, you can see this explanatory text by typing a question mark (?) in response to a question.

The questions, with the help text, are straightforward. An example that omits the explanatory text follows.

```
    Enter the device name, ?, or 'exit' -->uba1
                                .
                                .
                                .
    Enter the device that the uba1
    is connected to, ?, or erase -->nexus

    Adding data to config file


    Enter the device name, ? or 'erase' -->de1

    Enter the device that the de1
    is connected to, ? or erase -->uba0

    Enter the csr, ?, or 'erase' --->0174510

    Enter flags, ? 'erase', or 'none' --->none

    Adding data to config file
```

(continued on next page)

```
Enter the device name, ?, or 'exit' --->dz0

Enter the device that the dz0 is
connected to, ?, or erase --->uba0

Enter the csr, ?, or 'erase' ---> 0160100

Enter flags, ?, 'erase', or 'none' --->0xff

Adding data to config file


Enter the device name, ? or 'exit' --->exit

Configuration file complete
```

The script first asks for the ULTRIX name and logical unit number for the device and the device it is connected to.

The control status register (CSR) address is an octal address for the I/O device. For the proper CSR address, check your Site Management Guide, a notebook filled out when the hardware is installed. Note that the address given there is an 18-bit address, and only a 16-bit address is used in defining the configuration file. Thus, if the address given in the Site Management Guide is 760100, you would supply here the address 0160100. (The leading 0 is necessary.)

Some examples of CSR addresses are given in the *ULTRIX-32 Programmer's Manual Section 4.* However, these may or may not be correct at your site. Check the Site Management Guide to be certain.

Some device drivers recognize a hexadecimal value called a flag to further define what type of input/output is done for a device. For example, the DZ11 has a flag denoting that a particular line should be treated as hard-wired, with carrier always present. Section 4 describes any flag settings for a particular device. Hexadecimal numbers are specified as 0x*nnnn*, where *nnnn* are hexadecimal digits in the range 0 to 9 and 9 to f.

If the shell script does not have information about interrupt routines for the device, you must supply that information. You first request that an interrupt vector be defined for the device and then give the name of the interrupt-handling routine for the device. An example follows:

```
Are there any interrupt vectors for this device
Enter 'yes', 'no', or ? ---> yes
Enter the name of an interrupt routine or 'end' ---> vidint
```

For tape and disk devices, you are also asked for the drive number of the device, which is the physical unit number of the device. (For tapes, this is called the slave number.)

The shell script repeats this series of questions until you type end to the first question, to indicate you have finished specifying additional devices. If you type end in the middle of a series of questions about a device, that device is not added to the configuration file, and the questions are restarted.

The shell script then displays the message:

```
    Configuration file complete.

    Do you wish to edit the configuration
    file (y/n) [y]?
```

This question allows you to use a text editor to modify the configuration file just created. If you enter y to the above question, you are asked which text editor you want to use. Generally, you want to specify an editor name. However, you can type exit here if you want to stop the shell script for any reason.

If you specify an editor, the editor is invoked, with the configuration file as the file being edited. (See Appendix B for a description of the configuration file format.)

When you finish editing the file, the shell script resumes, displaying the following message:

```
    * * * Performing the System Configuration * * *
```

At this point, the script compiles and links the drivers indicated by the configuration file. If an error occurs, the shell script restarts with the editing step. For example:

```
    * * * Performing the System Configuration * * *
    config: line 5: syntax error
    Make:  Don't know how to make depend.  Stop.

    ---- Restarting the System Configuration
         Modification Query
```

(continued on next page)

```
    Do you wish to edit the configuration
    file (y/n) [y]?
```

Add one to the line number given for errors to determine the line where the error actually occurs. In the preceding example, the syntax error is actually at line 6. You can then edit the configuration file to correct the error.

If the build proceeds without error, you will see the MAKEDEV command that makes the special files for your system. The script then displays the message:

```
    * * * Configured kernel is in /newvmunix * * *

    What will follow is the sequence of steps that you
    must take to boot the newly configured ULTRIX
    kernel.

    If, after booting this kernel, you decide that you
    need to reconfigure the kernel, execute the
    command:

            doconfig

    and perform the booting steps, shown below, again.

    If the kernel is properly configured, then be sure
    to save the old kernel and move the new kernel to
    /vmunix, as follows:

           mv /vmunix /oldvmunix
           mv /newvmunix /vmunix
```

The script then displays the procedure for booting the new operating system. Follow this procedure and then go to Step 8 if you used an editor to change the configuration file in this step. If you did not use an editor, skip Step 8, and go on to Step 9.

Step 8: Configuring Devices You Added or Changed with an Editor

If you used a text editor to change the configuration file in Step 7, you need to create special files for the devices you added or changed with the editor. First, change your working directory to /dev:

```
#cd /dev
```

Then, run a shell script called MAKEDEV, naming each device you added with the text editor, or for which you used the text editor to change a unit number. For example, if you added dz0 with a text editor, type:

```
#MAKEDEV dz0
```

Step 9: Defining the Terminals

This, the last step in Phase 2 of the installation, is not automated. It involves using the ed text editor to modify the /etc/ttys file to define the terminals on your system.

The distribution kit contains a sample /etc/ttys file that you must edit in order to define the terminals that are part of your system.

Each line in the /etc/ttys file represents one terminal definition. Each line consists of these fields:

- terminal name
- command
- terminal type
- terminal status
- comments

Keep the following in mind when editing the /etc/ttys file:

- Use tabs and/or spaces to separate fields
- Enclose the command field within quotes, if the command field consists of two or more words
- Use as many blank lines as needed to separate terminal definitions
- Use this sign (#) at the beginning of a line to indicate the start of a comment.

Example 4-1 illustrates a sample /etc/ttys file. Note that the comments do not line up with their associated columns.

Example 4-1: Sample /etc/ttys File

```
#
# name command type status comments
#
console   "/etc/getty  std.9600"  vt100  on secure
ttyh0  "/etc/getty  std.9600"  vt100  off secure
ttyh1  "/etc/getty  std.9600"  vt100  off secure
ttyh2  "/etc/getty  std.9600"  vt100  off secure
ttyh3  "/etc/getty  std.9600"  vt100  on secure
ttyh4  "/etc/getty  std.9600"  vt100  off secure
ttyh5  "/etc/getty  std.9600"  vt100  off secure
ttyh6  "/etc/getty  std.9600"  vt100  off secure
ttyh7  "/etc/getty  std.9600"  vt100  off secure
ttyp0  none       network off secure
ttyp1  none       network off secure
ttyp2  none       network off secure
ttyp3  none       network off secure
ttyp4  none       network off secure
ttyp5  none       network off secure
ttyp6  none       network off secure
ttyp7  none       network off secure
ttyp8  none       network off secure
ttyp9  none       network off secure
ttypa  none       network off secure
ttypb  none       network off secure
ttypc  none       network off secure
ttypd  none       network off secure
ttype  none       network off secure
ttyq0  none       network off secure
ttyq1  none       network off secure
ttyq2  none       network off secure
ttyq3  none       network off secure
ttyq4  none       network off secure
ttyq5  none       network off secure
ttyq6  none       network off secure
ttyq7  none       network off secure
ttyq8  none       network off secure
ttyq9  none       network off secure
ttyqa  none       network off secure
ttyqb  none       network off secure
```

Modifying the Terminal Name Field

The terminal name field defines a name for each terminal special file in the /dev directory.

By convention, terminals connected through dz controllers are named ttyDD, where DD is a decimal number ranging from 00 through 99. Because only eight terminals can be connected to any particular dz, they are ordered according to the unit number of the dz. For example, the eight terminals connected to dz0 would be named tty00 through tty07. The eight terminals connected to dz1 would be named tty08 through tty15, and so forth.

Terminals connected through dmf controllers are named ttyHX, where H is a letter ranging from h for the first dmf, i for the second dmf, j for the third, and so forth. The X is a digit ranging from 0 through 7 for the eight terminals that can be connected to any particular dmf. For example, the eight terminals connected to dmf0 would be named ttyh0 through ttyh7. The eight terminals connected to dmf1 would be named ttyi0 through ttyi7, and so forth.

Note the terminal name entries in the sample /etc/ttys file in Example 4-1.

Modifying the Command Field

The command field defines the command used by the init program for this terminal line. In most cases the command you specify is /etc/getty. In the command field, you can also specify terminal characteristics such as the speed (baud rate) at which the terminal sends and receives characters.

The first line in the sample /etc/ttys file in Example 4-1 shows that:

- init executes the command /etc/getty for the terminal named console
- The baud rate for this terminal is 9600

The getty(8) program examines part of the command (std. 9600), and uses it as an index to the system table gettytab(5), which defines characteristics for many kinds of terminals. See the file /etc/gettytab on your system for a complete list of terminal characteristics.

You need not specify a command here, particularly if you are defining pseudoterminals. If this is the case, use the value none to indicate to init that no command should be executed. If you specify none, make sure you also specify off in the terminal status field.

Modifying the Terminal Type Field

The terminal type field contains a list of terminal types for each terminal port on the system. For example, the terminal type vt100 identifies a VT100 terminal. You can use any terminal type listed in the file

/etc/termcap. Some DIGITAL terminals listed there are vt100, vt125, and dw1, dw2, dw3, dw4 (for DECWRITER I through IV).

You also use this field to set up dial-up lines. You can connect up to 16 dialup lines to your ULTRIX system. Once the proper hardware is installed, indicate that the terminal can accept dial-up lines by using the value dialup in the type field.

Modifying the Terminal Status Field

The terminal status field defines a variety of actions allowed at a terminal. You can specify one or more of these values:

- on – This value enables logins at the terminal
- off – This value disables logins at the terminal
- secure – This value (when the value "on" is specified) enables root logins at the terminal
- nomodem – This value forces the terminal to ignore modem signals
- modem – The terminal is connected to a modem
- shared – This value accepts incoming (logins) and outgoing (using tip to log in) connections

This completes the steps for Phase 2 of the installation process.

Phase 3: Selecting Software Subsets 5

During this phase of the installation, you load some needed ULTRIX-32 utilities that have not been installed yet, and choose which optional software you want to install as part of your system.

Step 1: Run the Software Subset Script

To start the script to load the software, type:

```
    # /etc/instmedia
```

Step 2: Mount the Distribution Volume

If you need to mount a new volume of the distribution media, the script displays the following message:

```
Please mount the first <media> volume that contains
the ULTRIX software subsets.

Press RETURN when the unit is online and ready:
```

The appropriate volumes will be labeled OPTIONAL SOFTWARE. Pick the first in the series, load it onto the device, and prepare the device for read/write operation.

Step 3: Select from the Menu

The script displays a menu of the possibilities you can select. The space required by the optional software is listed below:

- FORTRAN-77. Approximately 1 Kbyte.
- PASCAL. Approximately 1 Kbyte.
- LISP. Approximately 3.4 Kbytes.
- MODULA2. Approximately 2.3 Kbytes.
- Games. Approximately 2.2 Kbytes.
- User Contributed Software. Approximately 5.2 Kbytes.
- Self-help programs (LEARN). Approximately 1.1 Kbytes.

- Documentation (/usr/man directories). Approximately 2.5 Kbytes.
- Auxiliary documentation (/usr/doc directories). Approximately 3.4 Kbytes.
- Fonts and Varian plotter programs. Approximately 2.8 Kbytes.

You make your choice, and the script loads the software. For example:

```
Here are the software subsets that you may now load into your
ULTRIX-32 system.  Note that those sets marked with a '*' are
required subsets and will be loaded automatically.

1)      BASUTIL*        Basic ULTRIX-32 utilities package
2)      LISP            The LISP and FP languages
3)      F77             The FORTRAN-77 language
4)      PASCAL          The PASCAL language
5)      MODULA2         The Modula-2 language
6)      GAMES           The usual array of fun and recreation
7)      SELFHLP         Self Help programs (LEARN)
8)      MANDOC          Documentation: /usr/man directories
9)      CONTRIB         User Contributed Software
10)     AUXDOC          Documentation: /usr/doc directories
11)     VARIAN          Fonts and Varian Plotter programs

Please specify ALL of the subsets that you wish to load by num
If you would like more detailed descriptions, type 'help'

For example, the selections: 1 2 4
would cause subsets BASUTIL LISP and PASCAL to be loaded.
and the selection: help
would cause a more detailed help message to be displayed
for each entry.

Please enter your selections: 2 3 9 11

You chose to add the following:  LISP F77 CONTRIB VARIAN

Is this correct (y/n) [n]: y

In addition, the following file set will also
be loaded:  BASUTIL
```

(continued on next page)

```
Skipping 1 subsets...

Loading BASUTIL...Finished
     .
     .
     .
(series of "loading" and "skipping" messages)
     .
     .
     .
Finished
```

If you need to load further volumes of the installation media, you will be instructed at the proper points during loading.

This completes Phase 3 of the installation procedure.

At this point, you can go to multiuser mode by pressing <CTRL/d>. However, you should remain in single-user mode if you plan to complete Phases 4, 5, and 6 immediately. If this is the case, you can go to multiuser mode after you complete Phase 6.

Phase 4: Install Local Area Network and/or uucp Connection 6

During Phase 4 of the installation, you run shell scripts to:

- Define whether or not your system is part of a local area network.
- Define whether or not your system will make connections with remote systems using uucp.

Local area networks are defined as computer systems connected by direct lines in one local area, such as a building or college campus. The ULTRIX commands rlogin, rcp, rsh, ruptime, and rwho are used for communication between systems in a local area network.

Connections using uucp can handle data communication over a wider geographic area, usually transmitting through telephone connections. The ULTRIX commands uucp, uux, uulog, uuname, and uustat are used for uucp communication.

Step 1: Run the Local Area Network Script

To start the local area network shell script, type:

```
#/etc/netsetup install
```

The word install tells the script that this is a first-time installation, so that it will ask questions about the local system before starting on the remote systems. (If you leave off the word install, the script simply updates existing network-related files.) It begins by asking what remote systems you want to add to the defined network.

Step 2: Give System Abbreviation

The script then asks for an abbreviation for the system name. Users can use this abbreviation when running rlogin(1c) or rcp(1c), for example, and not have to type the longer system name.

Usually, the abbreviation is simply the first letter of the system name. However, if you are joining the system to a computer network, and the one-letter abbreviation already exists on the network, use as many letters as are required to make a unique system abbreviation.

Step 3: Accept or Reject Local Area Network Configuration

The script next asks:

```
    Are you configuring your system on a network [no] ?
```

Enter yes if your system is to be part of a local area computer network. Otherwise, answer no or press the RETURN key to accept the default. If your answer is no, skip to Step 9, run the uucp script.

Step 4: Define Network Number

The script displays the question:

```
    Enter the network number for your network:
```

You respond with a number between 1 and 126. If yours is the first system on a network, other systems installed as part of this network will have to use this network number.

If you are installing your system as part of an existing network, this number should be the same as the network number of the other systems on the network.

Step 5: Define System Number

Each system in a network has an identifying number which serves as an address for remote connection programs such as rlogin(1c) and rcp(1c). The script displays the prompt:

```
Enter the system number on the network for <system-name>:
```

You respond with a number between 1 and 999. This number must be unique on the network. (Check with another system manager on the network to determine what numbers are free.) System numbers need not be assigned in numerical order.

Step 6: Define Network Name and Alias

During this step, you define the network name and network alias. The name and alias are used by utilities to identify your network should it be connected to other local area networks. The defaults are ethernet and deunanetworks.

The script displays the prompts:

```
Enter the network name [ethernet]:
Enter the network alias [deunanetworks]:
```

You respond with the appropriate name and alias, as determined when setting up your network, or simply accept the defaults.

Step 7: Name Other Systems on the Network

During this step, you name the other systems on the local network (those systems with the same network number as your system). You also answer questions defining the systems' abbreviated names and numbers. An example of the series of questions is shown below:

```
Enter the system name, abbreviation, and system number,
for each computer on the network.  Enter these on
separate lines when prompted.  This information is
put into the 'etc/hosts' file.

Enter the name of the 1st system to put on your network
(<RETURN> only if no more to enter): othernode

Enter the abbreviation for othernode: o

Enter the system number on the network for othernode: 7
```

You terminate this series of questions by pressing the RETURN key in response to the question about the name of the system.

Step 8: Define Trusted Systems on the Network

In this step, you define what are called trusted systems. Users on the systems you define as trusted can run rlogin to enter your system without being prompted for log-in name or password, as long as they have a valid account on your system, with the same name and password as they have on the remote system.

An example of the prompt follows:

```
   Enter the full name of the first 'trusted' system
   (<RETURN> only if none):
```

The name you type must have been entered as a valid system on the network in Step 7. If it has not, you are notified with an error message.

You terminate the series of questions about trusted systems by pressing the RETURN key in response to a question. You are then notified that the script is finished with the message:

```
Network setup is finished.
```

The information you supply in this step is stored in the file /etc/hosts.equiv.

Step 9: Run the uucp Script

To start the uucp shell script, type:

```
    #/etc/uucpsetup install
```

The word install tells the script that this is a first-time installation, so that it asks questions about the local system before asking for information on incoming and outgoing connections.

If you leave off the word install, the script will use existing uucp-related files and update them with information about new connections.

Step 10: Accept or Reject uucp Connections

The script asks:

```
Do you want your system set up for uucp connections [no] ?
```

Enter yes if you want users to be able to use uucp for data communications. If you enter yes, the script assumes that your modem is a DF02, DF03, or DF112 modem, connected for incoming calls on tty line 1 (ttyd1). Outgoing connections should be made through tty line 2 (ttyd0).

Otherwise, answer no or press the RETURN key to accept the default. Skip the remaining questions in this section and proceed to Phase 5.

Step 11: Define Outgoing Connections

During this step of the installation, you answer questions to define the calls your system will make to remote systems (outgoing connections). For each outgoing connection, you specify:

1. System name. This is the full name of the remote system, as defined in its installation.
2. Time to Call. You specify that your system will automatically call the remote system:
   - Any time. The system checks hourly to see if files have been queued for transfer to the remote system. If so, a connection is made to the remote system, and the files are transferred.
   - Evenings. Between the hours of 5:05 p.m. and 7:55 a.m. on weekdays, and any time on Saturday and Sunday, the system checks hourly to see if files have been queued for transfer to the remote system. If so, a connection is made to the remote system, and the files are transferred.
   - Nights. Between the hours of 11:05 p.m. and 7:55 a.m. on weekdays, any time on Saturday, and between midnight and 4:55 p.m. on Sundays, the system checks hourly to see if files

have been queued for transfer to the remote system. If so, a connection is made to the remote system, and the files are transferred.

- Never. Use this option for systems that can call your system, although your system never calls them.

3. Line speed of the remote system. You enter the baud rate expected by the remote system. The default rate is 1200 bits/sec.
4. Phone number of the remote system. The script accepts most typical formats for telephone number. A seven-digit number is accepted as a number within the local area code. A ten-digit number is accepted as area code and number. Parentheses and dashes are also accepted as punctuation (for example, (777)555-4444).
5. Log-in name and password. You specify the log-in name and password to be passed to the remote system for this system's login. You must, of course, have arranged with the system manager of the remote system to have an account set up for your system with this log-in name and password.
6. Modem type. The shell script allows you to specify DF02, DF03 or DF112. These are the only modems supported by DIGITAL. If you want to use another modem, you must edit the file /usr/lib/uucp/L.sys to reflect the modem type.

In addition to these questions, the script displays lines of information describing the entries it is making in the appropriate system files for these network connections. These will be of no particular interest to you, unless you have made a mistake in typing. In this case, note the file and entry, so you can later edit the files to correct the information.

This series of questions repeats until you press the RETURN key in response to the request for system name.

Step 12: Define Incoming Connections

Here, the script displays questions to allow you to define the remote systems that can access your system.

The system manager at the remote site should have provided you with the system name and password when the connection was requested.

Before the script asks questions, it creates two standard entries called remote and local, defining access for these general categories.

The remote entry defines the privileges extended to all remote systems you do not otherwise define in the following shell script dialogue. Such remote systems are not allowed to execute commands on your system. Therefore, the execution access level is defined as 0. Such remote systems, however, can transfer files to your system. These files are stored in the directory /usr/spool/uucppublic.

The local entry defines the privileges extended to all users with an account on the local system. These users are given the highest execution access level (9) and can access any directory allowed by the file protection scheme inherent in the ULTRIX-32 system.

After creating these standard entries, the script displays a message to prepare you for the questions to come. Execution access levels are explained as ranging from 0 through 9, with each level defining a set of commands that can be executed by systems with that access level. The commands for each level can vary from system to system and you are asked if you want to see the current definitions for your system, as kept in the file called L.cmds.

Next, for each incoming connection, you specify:

1. System name. This is the full name of the remote system, as defined when it was installed.
2. Short comment for the password file. The text you enter here is entered into the password file entry for the system.
3. Password for the remote system. This is the password expected from the remote system when it attempts to log in to the local system.
4. Execution access level for the remote system. A number from 0 through 9, defining the commands that can be executed by the remote system (see preceding discussion).
5. Call back option. To optimize your system's security, you can elect to call back this remote system when it attempts to call. In that way, you can ensure that you are making the connection with the remote system only. Otherwise, someone who knew the remote system's name and password could log in to your system without authorization. If you select the call back option, of course, you always pay the telephone bill for the remote connection.
6. Directory path for the remote system. This defines the directory to contain files transferred from the remote system to this system. The default is the directory /usr/spool/uucppublic.

This completes Phase 4 of the installation procedure. At this point, you can go to multiuser mode by pressing <CTRL/d>. However, you should remain in single-user mode if you plan to complete Phases 5 and 6 immediately. If this is the case, you can go to multiuser mode after you complete Phase 6.

Phase 5: Define Printer Capabilities 7

You define printer capabilities to establish how the ULTRIX-32 software controls access to the printers attached to your system. If your system is in a network and can use printers attached to remote systems, you also control access to those remote printers by defining their capabilities at your site.

The term printer capabilities can mean many things, from the hardware characteristics of the printers to the way print jobs are identified and managed. Until you define printer capabilities, your system's users cannot produce printed output.

7.1 How You Define Printer Capabilities

Your primary task is to build a data base of information about the capabilities of the printers that the people at your system can use. To build the data base, you determine what information you need to collect and then define that information to the printer capabilities data base.

The printer capabilities data base is the file named /etc/printcap, which is created during the automated phase of system installation. Most of the work you do to define printer capabilities involves tailoring the /etc/printcap file to reflect the intended uses for printers on your system.

There are other tasks associated with building the printer capabilities data base:

- You must collect information from other people, such as your site manager and users, to determine how individual printers are used.
- You have to define a spooling directory for every printer you define in /etc/printcap, so the system knows where to direct print jobs.
- You must define the directory pathnames for programs called printer output filters, when the default printer output filters provided with the system do not fit your needs.
- You have to determine whether your local printers are attached to your system through parallel or serial lines, and then identify and use their names in various data bases.

This chapter describes how you perform these tasks, and how you extract information from them to put into /etc/printcap. The line printer control program, lpc(8), provides commands for the day-to-day control of line printer jobs.

7.2 When to Start

You can start to define printer capabilities any time after the automated phase of installation is complete through Phase 4.

You should also know how to do the following:

- Use a text editor to build entries in data base files, such as /etc/printcap and /etc/hosts.
- Use elementary commands, such as mkdir(,) cp(1), ls(1), and cat(1).
- Know the physical configuration of your system, particularly the ports to which printers are attached. This information should be in the Site Management Guide, which is prepared by the DIGITAL field service personnel who install the printer hardware. You use this information to relate port numbers to logical device names in the /dev directory.
- Locate and manipulate files within the system, such as the sample /etc/printcap entries.
- Build some files, principally the printcap data base, to construct or tailor data base items.
- Obtain and organize information, such as the identification needed for any remote printer your system users can access, and the intended features and uses of each printer.

There are two steps to the process: define the printers and build the printcap data base.

Step 1: Define the Printers

The operating system supports several different printers. As an operator, all you need to know about supported printers is that they are installed properly and that you need to make some definitions based on how they are installed.

To define printers, you must relate the physical connection (the port) of each local printer to a logical device name for that printer.

Use the form in Figure 7-1 to record information as you complete the work you do to define the printers. During the next step, Build the printcap Data Base, you will use these entries.

Printer	Interface	Port No.

Figure 7-1: Logical Device Names for Printers

For example, an LQP02 printer attached through a DZ11 serial line to port number 74-4 and assigned a logical device name of tty42 would have an entry like this:

```
LQP02    DZ11    74-4    tty42
```

Naming Printers and Interfaces

Using the form in Figure 7-1, write down the name of each printer in column 1 and the name of the interface in column 2. For DMF32 interfaces, you might add serial in column 2, to distinguish between DMF32 parallel and serial interfaces.

The supported printers are:

- Printers attached through an LP11 parallel interface:

  LP25, LP26, LP05, LP14, LN01
- Printers attached through DZ11 serial lines:

  LQP02, LN03 (acting as LPQ02)
- Printers attached through DMF32 parallel interface:

  LP25, LP26, LP05, LP14, LN01
- Printers attached through DMF32 serial lines:

  LQP02, LN03 (acting as LPQ02)

Locating Printer Ports

Each printer is plugged into a port on the port panel. These ports are identified when the printer hardware is installed. Your Site Management Guide has a port definition for each printer (as it does for all install peripheral devices).

Record the port number for each printer in the Port No. column of the form in Figure 7-1.

Choosing Logical Device Names

Next, you choose logical device names for printers. These names will end up as the names of files in the /dev directory after you complete Step 1. You also use the logical device names when working through Step 2, Build the printcap Data Base.

In one case, the work is done for you during the automated phases of installation. That case is an LP11 interface to either an LP25, LP26, LP05, or LP14 printer. The result is an automatic entry in the /dev directory for lp0. The entry is the file named /dev/lp0.

For all other printers, you must relate the port for the printer to a logical device name in the /dev directory. Use the following naming conventions:

- lpn (for printers attached by LP11 parallel interfaces)
- lphn (for printers attached by DMF32 parallel interfaces)
- ttynn (for printers attached by all serial interfaces)

Record the logical device names in the fourth column of the form in Figure 7-1.

Running the MAKEDEV Script

Next, you create a device special file in the /dev directory for each line printer. To do so, you run the shell script MAKEDEV for each logical device you have defined in column 4 of the form in Figure 7-1. You use the logical device name as the parameter passed to MAKEDEV. For example:

```
% cd /dev
% MAKEDEV tty42
```

In this case, MAKEDEV creates the device special file for tty42 and names the file tty42.

You must run MAKEDEV for each logical device name. The resulting files become named special files for known devices.

Checking the Results

When you have finished running the MAKEDEV script, you can check the results by examining the /dev directory. Use the ls(1) command to determine that each entry you created is named.

Step 2: Build the printcap Data Base

Once you have defined your printers, you can build the printcap data base for those printers. To build entries in the printcap data base, you use these resources:

- A text editor, to tailor the database
- The system file named /etc/printcap.

  This file contains representative printcap entries for supported printers. You should edit these templates if your printers correspond to them. Otherwise, construct your own entries, using the templates and the descriptions of printcap parameters as a guide.
- The printer logical device names (from the /dev directory and the form in Figure 7-1.

  You recorded these names when you defined the location of printers.
- The printcap(5) file format definition

  This definition includes the name of the entry and parameters (and their arguments) for each supported printer.
- The information here, including forms you can use to record the data you might need.

Defining printcap Entries and Parameters

To build printcap entries, you have to collect different categories of information from different sources. Then, you use the definition of printcap(5) to prepare an entry for each printer. The following sections describe the major parameters in printcap(5). For other parameters you may need, consult the printcap(5) documentation.

Name of a Printcap Entry – The first required item in any printcap entry is its name. By convention, this is the logical device name for the printer. Thus, the first item in the entry for the printer located by /dev/lp0 is lp0; the first item in the entry for the printer located by /dev/tty42 is tty42.

Use the logical device names you have recorded on the form in Figure 7-1 to begin each printcap entry.

Device to Open for Output – When the system starts a print job, it needs to open a device for the output it produces. You define that device with the lp parameter. The argument for lp is the name of the file you created when you ran the MAKEDEV script. For example, you would define the device for tty42 as:

```
:lp=/dev/tty42.
```

Print Job – Many printcap(5) parameters pertain to how individual print jobs are managed. Some of the major parameters are sb, for separating print jobs; sc, for allowing multiple copies of a print job; and other parameters, for defining the paging characteristics of individual printers.

These parameters are site- and device-specific. To construct these parameters for any printer, discuss your needs with your site manager.

Serial Lines – Some printcap(5) parameters pertain only to printers attached through a serial line. The major ones are br, for setting the printer's baud rate; and the transmit/receive mode parameters fc, fs, xc, and xs. Again, these parameters are device-specific.

To construct these parameters for any serial printer, discuss what is needed with your site manager.

You can use the form in Figure 7-2 to record the information you need.

Name of the printcap entry (from /dev/''entry''): ____________

Baud rate used for printer (from site manager): ____________

Terminal mode bit settings (from site manager):

fc Parameter: ____________

fs Parameter: ____________

xc Parameter: ____________

xs Parameter: ____________

Figure 7-2: Serial Line Printcap Parameters

Spooling Directory – Each line printer must have its own spooling directory. Use the parameter sd to define this spooling directory to the printcap data base. Remember, you must create the spooling directory itself, unless you use the default directory /usr/spool/lpd.

The spooling directory for a printer must contain a lock file. The default spooling directory /usr/spool/lpd contains a lock file. Whenever you define a spooling directory, create a lock file in it. Then, define that file to the printcap data base by using the lo parameter.

Output Filters – Every printer requires an output filter program. DIGITAL supplies a set of default output filter programs for the supported printers. The default output filter programs reside in /usr/lib . If these default output filter programs are not acceptable, you must either write your own or locate one from a remote system in the network.

If you write an output filter program, you must:

1. Complete the task of adding users to the system (see the *System Manager's Guide*).
2. Copy the output filter program to the directory /usr/lib/lpf.
3. Define the directory path to the printcap data base.

If you locate an output filter program on a remote system in the network, you must:

1. Collect the information about the output filter program at the remote location. Use the form in Figure 7-3 to record the location of an output filter.
2. Ensure you are working in a network.
3. Copy the output filter program to the directory /usr/lib/lpr.
4. Define the directory path to the printcap data base.

If neither of the above options is possible now, leave the parameter null and edit the printcap entry when the filter is available.

Location of Filter (from its owner):

____

Copy the file to this location:

/usr/lib/lpf/*name*

(*name* is the filter file)

Figure 7-3: Local Printer Output Filter

Remote Printer – When your users can access a remote printer, you must define two parameters, rm and rp.

The argument for rm is the name of the remote host machine; the argument for rp is the name–at that host–of the remote printer. You must obtain these names from the remote site manager.

You also should check the contents of your system's /etc/hosts file, which contains the names of the remote machines in your network. If the remote system name is not in /etc/hosts, you must edit the name into this data base.

The form in Figure 7-4 lists the information you must get for the rm and rp parameters.

Note that you do not have to define a spooling directory for any remote printer, as the remote site has defined it as part of that system's printcap data base. For remote printers, you make the sd parameter null.

Obtain these parameters from the remote site manager:

rm Parameter (remote machine name): ________________

rp Parameter (remote printer name): ________________

Figure 7-4: Remote Host and Printer Names

Using Template printcap Entries

The /etc/printcap file contains comments which represent data base entries for supported printers used in typical situations.

To use the templates, compare them to what you need, then edit them. Should you make an error while editing, you can recover by copying the file /etc/printcap.examples, a replicate of the templates, to /etc/printcap.

The entries, in order, are:

- Draft printer on an LP11 interface

  Use this template for any LA25, LA26, LP05, or LP14 printer on an LP11 controller.

- Letter-quality printer on a DZ11 serial interface

  Use this template for an LQP02 printer on a DZ11 serial line.

- Laser printer on a DMF32 parallel line

  Use this template for an LN01 on the parallel line of a DMF32 controller.

- A serial printer at a remote site on a DZ11 serial interface

  Use this template for an LN03 at a remote site.

Testing Results

When you have finished building the data bases, you should test the definitions by exercising the spooling system before releasing it to the system's users.

Do this is by submitting print jobs to all the defined printers and observe whether they work as they should.

Phase 6: Install sendmail for Networks 8

If your system is part of a network, you will probably want to install sendmail so that your users can send mail to people at other sites. This chapter tells you how to install sendmail so that mail can be sent to other systems over one of the following types of network:

- DECnet networks and/or the ARPA (Department of Defense Advanced Researched Projects Agency) network.
- uucp networks.

It is possible to create your own sendmail configuration file so that mail can be exchanged between sites connected by a network using different protocols than those listed above. To do this work, though, you must have purchased a source code license. See the sendmail article in *ULTRIX-32 Supplementary Documents.* This article also gives information on other files used by sendmail, the implementation of sendmail, and information on maintaining the sendmail system.

Two prototype configuration files are supplied with ULTRIX-32 for the two types of network on which sendmail can be installed. The files are named exampleether.cf (for DECnet networks and/or the ARPA network), and exampleuucp.cf (for uucp networks). These files are in the directory /usr/src/usr.lib/sendmail/cf.

During this phase of the installation, you execute a makefile that creates all the binary files needed. Next, you copy the appropriate prototype file to a file named for your system, in the form *system-name*.cf in the same directory. The *system-name* is the name for the system that you supplied in Phase 2 of the installation procedure. Last, you copy the system-name file to the /usr/lib directory as /usr/lib/sendmail.cf.

For example, if you wanted to install sendmail to work on DECnet networks, and your system name was MYVAX, you would execute the following commands:

```
    #cd /usr/src/usr.lib/sendmail/cf
    #make
    #cp exampleether.cf MYVAX.cf
    #cp MYVAX.cf /usr/lib/sendmail.cf
```

This completes the installation procedure.

Installation Shell Script Error Messages A

Here are error messages which can be displayed by the shell scripts during installation.

***option*: invalid option value. Ignored.**

> You made a selection that does not appear on the menu. It is ignored, and the script waits for you to type a valid item.

***system-name* contains invalid characters.**
Please reenter using UPPERCASE alphabetics only.

> Your system name must be constructed of uppercase alphabetic characters only.

A system with that name already exists.
Replace it?

> A system has been generated with that name (a file of that name is already in the directory /sys/**conf**). If you choose to replace it, answer yes and the script will ask questions to generate a new configuration file. If you answer no, the script will use the existing configuration file as a start, and go on from there.

***system-name* is already in /usr/lib/uucp/L.sys file.**

> The system *system-name* has already been defined for outgoing uucp connections.

Bad Geographic area.

> The geographic area you signified for daylight savings time is not one of the areas listed in the menu (u, a, e, c, or w).

Bad input

> You did not enter one of the valid values for specifying time of day to call. The valid values are:
>
> a For any time.
>
> e For weekday evenings between 5:05 PM and 8:00 PM, or anytime Saturday and Sunday.

n For weekday nights between 11:05 PM and 7:55 AM, all day Saturday, and Sunday between midnight and 4:55 PM.

x For never. This answer is given for systems from which you expect incoming calls, but which you never expect to call.

Bad value.

The time zone you typed is not one of the zones listed in the menu (e, c, m, p, or g). Nor is it a digit within the range 0-24, which would be acceptable for a time zone expressed in hours west of Greenwich.

Enter a larger number.

System numbers must be within the range 1-999.

Enter a smaller number.

System numbers must be within the range 1-999.

The device*dev* has already been defined in the config file

You cannot have two of the same devices with the same unit number configured into your system.

Invalid disk drive: *name.* **Must be one of:**
disk1 disk2 .....

Specify the DIGITAL name of the disk drive. Select the disk drive you are using from *disk1*, *disk2*, and so forth.

Invalid hex number
Hex numbers are of the form 0x[0-9a-f]. Please reenter.

All flag values are hexadecimal numbers. Enter them in the format shown.

Invalid octal number.
Octal numbers are of the form 0[0-7]. Please reenter.

CSR addresses are expected in 7-digit octal format, with a leading 0. For example, 0167322.

Invalid tape drive: *name.* **Must be one of:** *tape1 tape2* ....

Here, you specify the DIGITAL name of the tape drive. Select the tape drive you are using from *tape1*, *tape2*, and so forth.

? ? ? Mount failed. Try again. ? ? ?

The software mount command did not work for the most recent volume of the software subset media. Check to make sure the volume is properly mounted and run the /etc/instmedia script again.

Only DF02, DF03, or DF112 supported.

The DF02, DF03, and DF112 modems are the only ones supported by ULTRIX-32. To specify another type of modem, you must edit the appropriate file.

Please su to root first.

You must be logged in to the superuser account before the shell script can continue. Use the command:

```
su root
```

and answer the prompt for the superuser password. Then, start the shell script again.

***response* is not a valid directory name.**

The response you typed in is not a valid directory name. Enter a valid directory name. For example:

```
/usr/kstaff
```

Sorry, the *dev* device is not supported by DIGITAL at this time

The shell script will only accept DIGITAL-supported devices in its question-and-answer session. If you want to add nonsupported devices, you must use ed to edit the configuration file.

System *name* was not entered.

The system you have named as trusted has not yet been defined as being on the network. You must do that first, before naming it as a trusted system.

The execution access level must be between 0 and 9.

Execution access levels are defined in the file /usr/lib/uucp/L.cmds. In general, they range from 0 (no access) to 9 (most access).

The network number must be greater than 0.

Negative or zero network numbers are not allowed.

The network number must be less than ***nnnn.***

The network number you entered is greater than the allowable number.

Configuration File Format B

During Phase 2 of the installation process, you generate a configuration file with the help of a shell script named mkconfig. The configuration file defines which software drivers are to be built into the kernel. A driver is a program that handles input/output.

The shell script in Phase 2 prompts you for information and constructs the configuration file from the information you supply. This appendix describes the format of the configuration file for those who are interested, or who wish to change or add to the definitions supplied by the shell script.

B.1 Overall Format

A configuration file is organized in four major sections:

- Global definitions. The shell script supplies these definitions. They apply to all the system images (kernel files) to be generated by this configuration file. Each line begins with one of the following keywords:

  ```
  ident
  machine
  cpu
  maxusers
  maxuprc
  physmem
  dmmax
  dmmin
  timezone
  options
  ```

- System definitions. The shell script generates one line of system definitions for the system you are using for the installation. You can change the definition it supplies, or add more lines, to indicate that you want to generate more than one kernel. For each kernel you wish to generate, you specify one line that begins with the keyword config. Each line can be used to define the root device, the swap area(s), the dump area, and the argument processing area for system calls.

- Device definitions. The shell script polls the system and supplies device definitions for those devices that it finds. You need to add definitions for communications devices, such as terminal controllers. You may also wish to add device definitions for those devices you expect to have in the future. The shell script will present questions for you to answer, to define each device. Or, you can edit the configuration file. In any case, each line of this section of the configuration file defines a specific hardware device, beginning with one of the following keywords:

  ```
  controller
  device
  disk
  master
  tape
  ```

- Pseudodevice definitions. Each line defines a driver for which there is no associated hardware. Each line defines a specific pseudodevice, beginning with the keyword pseudo-device, followed by the pseudo-device name.

The following sections describe the four major parts of the configuration file in detail.

B.2 Global Definitions

Each global configuration parameter appears on a separate line:

ident *name*

> The shell script enters the *name* argument as the system name you gave in response to a question in Phase 2 of the installation process. In general, *name* must consist of all alphabetic characters. GENERIC is a name for a system that runs on any VAX CPU. Do not use GENERIC for the name unless you want to configure a generic system.

machine *type*

> The argument *type* in this case is vax:
>
> ```
> machine vax
> ```

cpu *"type"*

Legal values for *type* are: VAX780 (for VAX-11/780 and VAX-11/785 processors), VAX750, VAX730, VAX8600, and VAX8200. For example:

```
cpu "VAX750"
```

You can specify more than one processor type, using more than one line beginning with cpu. For example:

```
cpu "VAX780"
cpu "VAX750"
```

maxusers *number*

The *number* argument gives the maximum number of simultaneously active users allowed on your system, as allowed by your license agreement. Do not alter this argument in your configuration file.

maxuprc *number*

The *number* argument gives the maximum number of processes one user can run at any given time. The shell script sets this number at 25.

physmem *number*

The *number* argument gives the number of megabytes of memory on your system.

dmmin *num1*
dmmax *num2*

The system satisfies requests for additional virtual memory using these values. A process is initially granted *num1* 512-byte blocks of virtual memory. The next time the process requests memory, the system grants twice as much (*num1* x 2). This allocation continues until the amount of memory granted is equal to *num2* blocks. After that, additional requests are satisfied with *num2* blocks of memory.

The default value for *num1* is 32. The default value for *num2* is 1024.

The *num1* and *num2* values should be multiples of 2, and *num2* should be a multiple of *num1.* Otherwise, the system's behavior is unpredictable.

timezone *number* [dst [*x*]]

The shell script supplies this value according to information you supplied earlier. The *number* argument identifies your time zone, measured by the number of hours west of Greenwich Mean Time (GMT). For example, Eastern Standard Time is five hours west of

GMT, and Pacific Standard Time is eight hours west. Negative numbers indicate hours east of GMT.

The argument dst indicates daylight savings time. You can include a number (*x*) to request a particular daylight savings time correction algorithm. The values are:

1	United States (the default value)
2	Australia
3	Western Europe
4	Central Europe
5	Eastern Europe

options *optionlist*

Indicates optional code to be compiled into the system. The shell script provides the INET and QUOTA options. The INET option provides Internet communication protocols. The QUOTA option allows disk quotas to be set. You should leave the options as they appear.

B.3 System Definitions

The shell script provides one line for the kernel it expects to build. You may want to change this line or add other lines for other kernels to be generated. The general format of a line for any one kernel is as follows:

config *filename configuration-clauses*

The *filename* argument is the name to be assigned to the file constituting the compiled kernel, or system image. The shell script assigns the name vmunix.

The *configuration-clauses* define the devices for the root filesystem, the paging and swapping area, crash dumps, and system call arguments:

root [on] *device*

The shell script assigns the a partition of the system disk to the root filesystem.

swap [on] *device* [and *device*] [size *x*]

The shell script assigns the b partition of the system disk for the paging and swapping area. If you want to add another partition, so that the kernel interleaves paging and swapping between the two partitions, use the and clause with a device, logical unit, and partition name.

The size clause can be used to specify a nonstandard partition size for one or more swap areas. The value of *x* is given in 512-byte sectors. A size larger than the associated disk partition is trimmed to the partition size.

Use the clause swap generic for a generic system.

dumps [on] *device*

If you requested crash dumps in Phase 2, the shell script assigns the b partition of the system disk as the area in which to write them.

args [on] *device*

The shell script assigns the b partition of the system disk as the area where the system should process arguments during system calls.

B.4 Device Definitions

This section of the configuration file contains descriptions of each current or planned device on the system. The shell script polls the system and constructs the proper descriptions for the devices it recognizes. You need to supply descriptions for communications devices such as terminal multiplexers, or devices you may want to add at a later date.

Each line of this section of the file begins with one of the following keywords:

controller A disk controller, a UNIBUS tape controller, a MASSBUS adapter, or a UNIBUS adapter

device An autonomous device which connects directly to a UNIBUS adapter (as opposed to a disk, for example, which connects through a disk controller)

disk A disk drive connected to a controller or master

master A MASSBUS tape controller

tape A tape drive connected to a controller or master

The format of the information required for each of these types of devices varies, as described in the following subsections.

B.4.1 Controllers — MASSBUS and UNIBUS Adapters

The format for MASSBUS adapters and UNIBUS adapters is:

```
mban at nexus ?
uban at nexus ?
```

The *n* is the unit number of the adapter. A NEXUS connects the UNIBUS and MASSBUS adapters on a VAX-11/785, VAX-11/780, or VAX 8600 to the internal system bus. A NEXUS also connects the UNIBUS adapter on a VAX 8200 to the internal system bus. You can specify a NEXUS for these machines by replacing the question mark (?) with the number of the NEXUS. Otherwise, accept the question mark (?), and the system will probe all NEXUSES present for the specified controller.

For consistency, you specify a NEXUS connection for UNIBUS and MASSBUS adapters on VAX-11/750 and VAX-11/730 processors even though NEXUSES do not exist on these machines.

For example:

```
controller mba0 at nexus ?
controller mba1 at nexus ?
controller uba0 at nexus ?
```

B.4.2 VAX BI Controllers

The VAX BI (VAX backplane interconnect) can have two types of controllers: the DWBUA, and the KDB50.

The DWBUA is the UNIBUS adapter on a VAX 8200 processor. Only one DWBUA is allowed in the system. This example shows a sample entry for the first DWBUA adapter on a VAX BI:

```
controller uba0 at nexus ?
```

The KDB50 is a DSA adapter for the BI. Each KDB50 on a VAX 8200 processor is treated as a separate uba with a uda at csr address 0362 (octal). This example shows sample entries for a KDB50 on the second adapter on the BI:

```
controller uba1   at nexus?
controller uda1   at uba1 csr 0000362
disk       ra0 at uda1 drive 0
disk       ra1 at uda1 drive 1
disk       ra2 at uda1 drive 2
disk       ra3 at uda1 drive 3
```

B.4.3 Other Controllers

The format for the specification of the other controllers is as follows:

controller *dev* at *condev* csr *n* vector *vec*

where:

dev is the device name and logical unit number of the controller.

condev is the name and logical unit number of the device to which the controller is connected.

n is the octal address of control status register for the device. Samples are listed below. but you should check the Site Management Guide at your site to be sure of the correct CSR address. Note that the address given in the Site Management Guide is an 18-bit address, while the address needed here is a 16-bit address. So, for example, if the address given in the guide is 772520, here you would use the address 0172520.

vec is the address of any interrupt vector for the controller.

The following list shows sample device specifications for controllers supported by ULTRIX-32.

RK07 moving head disk

```
controller hk0 at uba0 csr 0177440 vector rkintr
```

RX02 diskette interface

```
controller fx0 at uba0 csr 0177170 vector rxintr
```

TM-11/TE-10 magnetic tape interface

```
controller tm0 at uba1 csr 0172520 vector tmintr
```

TS - TS-11 magtape interface

```
controller zs0 at uba1 csr 0172520 vector tsintr
```

UDA - UDA-50 disk controller interface

```
controller uda0 at uba0 csr 0172150 vector udintr
```

UP - UNIBUS storage module controller

```
controller sc0 at uba? csr 0176700 vector upintr
```

B.4.4 Device Specifications

The format for the hardware classified as devices is:

device *dev* at *condev* csr *n* [flags *f*] vector *v1* ...

TAB characters are used to indicate continuation lines, if needed. The arguments you supply are:

dev is the device name and logical unit number of the device.

condev is the name and logical unit number of the adapter or controller to which the device is connected.

n is the octal address of the control status register for the device. Samples are listed below, but you should check the Site Management Guide at your site to be sure of the correct CSR address. Note that the address given in the Site Management Guide is an 18-bit address, while the address needed here is a 16-bit address. So, for example, if the address given in the guide is 772520, here you would use the address 0172520.

f is a number used to convey information about the device to the device driver. The only flags for DIGITAL-supported devices are for:

Line Printers

The default page width for all DIGITAL line printers is 132 columns. If you wish to change this, use flags *n*, where *n* is a decimal number giving the desired width in columns. For example:

```
    flags 80
```

Communications Multiplexers

The DH11/DM-11, DMF-32, and DZ11 multiplexers accept a hexadecimal flag value to specify any lines that should be treated as hard-wired with carrier always present. The format of the hexadecimal number is 0x*nn*, where *nn* is a hexadecimal number consisting of digits ranging from 0-9a-f.

Setting bit 0 of the flag indicates that tty00 is hardwired, setting bit 1 of the flag indicates that tty01 is hardwired, and so forth. The following example shows that tty02 is hardwired with carrier always present:

```
    flags 0x04
```

v1... are the names of interrupt vector routines for the device driver. The examples below show the needed names for DIGITAL-supported devices.

The following list shows sample device specifications for devices supported by DIGITAL.

DEUNA 10Mb/s Ethernet Interface

```
    device de0 at uba0 csr 0174510 vector deintr
```

DMC11/DMR11 point-to-point communications device

```
device dmc0 at uba0 csr 0167600 vector dmcrint
dmcxint
```

DMF32 terminal multiplexer

```
    device dmf0 at uba0 csr 0160340
        vector dmfsrint dmfsxint dmfdaint dmfdbint
        dmfrint dmfxint dmflint
```

DZ11 communications multiplexer

```
    device dz0 at uba0 csr 0160100 vector dzrint dzxint
```

Line Printer (any standard DIGITAL line printer on an LP11 parallel interface

```
    device lp0 at uba0 csr 0177514 vector lpintr
```

B.4.5 Disk Specifications

The format for disks is:

Rdisk *dev* at *condev* drive *n*

dev is the device name and logical unit number of the disk.

condev is the name and logical unit number of the adapter or controller to which the disk is connected.

n is the physical unit number of the disk.

The following list shows sample device specifications for disks supported by DIGITAL.

RK07 moving head disk

```
    disk rk0 at hk0 drive 0
```

RM03, RM05, RM80, RP05, RP06, RP07 disks

```
    disk hp0 at mba0 drive 0
```

RA60, RA80, RA81 disks

```
    disk ra0 at uda0 drive 0
```

B.4.6 Master and Slave Specifications

The format for hardware classified as master and slave is:

master *dev* at *condev* drive *n*
tape *dev* at *condev* slave *n*

dev is the device name and logical unit number of the device.

condev is the name and logical unit number of the adapter or controller to which the device is connected.

n is the physical unit number of the device.

The following list shows sample device specifications for master and slave devices supported by DIGITAL. Section 4 of the *ULTRIX-32 Programmer's Manual* shows sample specifications for these and other devices for which drivers are supplied with the ULTRIX-32 system, but are not supported by DIGITAL.

TM-03/TE16, TU45, TU77 magnetic tape

```
master ht0 at mba1 drive 0
tape tu0 at ht0 slave 0
```

TM78/TU78 magnetic tape

```
master mt0 at mba1 drive 0
tape mu0 at mt0 slave 0
```

TS11 magnetic tape

The TS11 magnetic tape controller is specified with a controller, rather than a master, specification:

```
controller zs0 at uba0 csr 0172520 vector tsintr
tape ts0 at sz0 drive 0
```

B.5 Pseudodevice Definitions

Each line defines a driver for which there is no associated hardware. The format is:

pseudo-device *name*

The *name* is the name of the pseudodevice. The shell script automatically generates four lines in your configuration file, incorporating the following pseudodevices:

pty for DARPA Internet protocols.

inet for DARPA Internet protocols.

loop for software loopback interface.

ether for 10Mb/s ethernets.

Section 4 of the *ULTRIX-32 Programmer's Manual* lists other pseudodevices.

B.6 VAX 8200 Processor Pseudodevice Definitions

The following are the required pseudodevice definitions for the VAX 8200 processor:

crx	for RX50 diskette on VAX 8200
TRka	for KA820 cpu board
Rbua	for DWBUA UNIBUS adapter
bimem	for VAX BI memory
bi	for VAX BI bus

The ed Command Summary C

This appendix provides enough information on the ed text editor for you to edit the files necessary to install your ULTRIX-32 operating system.

C.1 Concepts

To use ed you must understand a few concepts:

command mode
: In command mode, ed is ready to accept a command. When you first invoke ed, you are in command mode, as indicated by the question mark (?) prompt that ed displays.

In this mode, you can type any of the commands listed in Section C.2. Follow each command by pressing the RETURN key.

To get back to command mode from insert mode, type a period in the first position of a line, and then press the RETURN key.

input mode
: In this mode, everything you enter is inserted in the text file. You enter input mode by typing an a (for append) or i (for insert) in command mode. To leave input mode, type a period in the first position of a line, and then press the RETURN key.

current line
: The current line of a file is the line at which you are working. When you first invoke ed the current line is the last line in the file. The commands you give can change the placement of the current line.

C.2 Commands

The following list shows commonly used ed commands. Note that the period (.) can be used to refer to the current line, and the dollar sign ($) can be used to refer to the last line in the file.

ed *filename*
: Opens the file named *filename* for editing, with the current line being the last line in the file.

.=
: Prints the number of the current line. This can help you tell where you are in the file.

[*n*,*m*]p — Prints lines *n* through *m*. Leaves the current line at line *m*. If you do not specify a line number, ed prints the current line. For example: 1,10p prints the first 10 lines of the file, and leaves the current line at line 10.

- — Makes the previous line the current line.

a — Appends text below the current line. Terminate the appended text with a period (.) on a line by itself.

i — Inserts text above the current line. Terminate the inserted text with a period (.) on a line by itself.

dp — Deletes the current line and prints the new current line.

/*xxx*/p — Searches for the string *xxx* and prints the line containing that entry. This line becomes the current line.

s/*xxx*/*yyy*/p — Substitutes the string *yyy* for the string *xxx* in the current line and then prints the line.

w — Writes the file and incorporates all changes. To save the file with all changes, use the w command to write the file, and then the q command to exit the editor.

q — Quits (exits) the file. To prevent mistakes, ed does not allow you to use the q command without first writing the file with the w command.

Q — Quits (exits) the file without incorporating changes. The file remains as it was before the editing session.

C.3 Sample ed Session

Here is a sample ed session with an example /etc/fstab file.

```
#ed /etc/fstab
67                              (number of characters in the file
<RETURN>
?                               (ed prompts for next command)
1,$p                            (print the entire file)
/dev/hk0a:/:rw:1:1
/dev/hk0h:/usr:xx:1:2
/dev/hk1a:/usr/sys:rw:1:3
-                               (move back current line)
/dev/hk0h:/usr:rw:1:2
s/xx/rw/p                       (substitute rw for xx)
/dev/hk0h:/usr:rw:1:2           (prints the modified line)
<RETURN>                        (make next line the current line)
```

(continued on next page)

```
/dev/hk1a:/usr/sys:rw:1:3
a                                (add text after the current line)
/usr/dev/hk1h:/usr/users:rw:1:4  (appended text)
.                                (terminate append mode)
<RETURN>
?                                (ed prompts for next command)
1,$p                             (print the entire file)
/dev/hk0a:/:rw:1:1
/dev/hk0g:/usr:rw:1:2
/dev/hk1a:/usr/sys:rw:1:3
/usr/dev/hk1g:/usr/users:rw:1:4
i                                (insert text above the current line)
/usr/dev/hk1g:/usr/mnt:rw:1:4    (inserted text)
.                                (terminate insert mode)
1,$p                             (print the entire file)
/dev/hk0a:/:rw:1:1
/dev/hk0h:usr:rw:1:2
/dev/hk1a:/usr/sys:rw1:3
/usr/dev/hk1g:/usr/mnt:rw:1:4
/usr/dev/hk1g:/usr/users:rw:1:4
-                                (move back the current line)
/usr/dev/hk1g:/usr/mnt/rw:1:4
dp                               (delete the current line)
/usr/dev/hk1g:/usr/users:rw:1:4
1,$p                             (print the entire file)
/dev/hk0a:/:rw:1:1
/dev/hk0h:/usr:rw:1:2
/dev/hk1a:/usr/sys:rw:1:3
/usr/dev/hk1g:/usr/users:rw:1:4
w                                (write the file)
100                              (number of characters in the file)
q                                (quit)
#
```

For further information on ed, read the article in the *ULTRIX-32 Supplementary Documents*, Volume I.

Index

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA and Puerto Rico call **800–258–1710**	In Canada call **800–267–6146**	In New Hampshire, Alaska or Hawaii call **603–884–6660**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809–754–7575

ULTRIX-32
Installation Guide
AA-BG58C-TE

Reader's Comments

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. ______________________________

____

____

____

Did you find errors in this manual? If so, specify the error and the page number.

____

____

____

Please indicate the type of user/reader that you most nearly represent.

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Student programmer
- □ Other (please specify) ______________________

Name ______________________ Date ______________

Organization ______________________________________

Street ______________________________________

City ______________________ State ______ Zip Code or Country ______________

------Do Not Tear - Fold Here and Tape --

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 33 MAYNARD MASS

POSTAGE WILL BE PAID BY ADDRESSEE

Documentation Manager
ULTRIX Documentation Group
MK02-1/H 10
Continental Blvd.
Merrimack, N.H. 03054

------Do Not Tear - Fold Here and Tape --

Cut Along Dotted Line