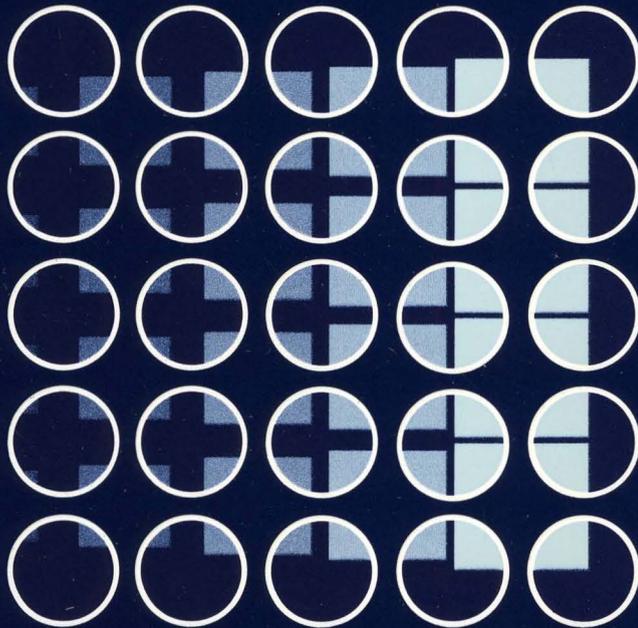


ULTRIX-32™

Programmer's Manual

ULTRIX-32™
Installation Guide

Order No. AA-BG58A-TE



digital
software

ULTRIX-32TM
Installation Guide

Order No. AA-BG58A-TE

digital equipment corporation, merrimack, new hampshire

Copyright © 1984 by 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 postage-paid READER'S COMMENTS form on the last page of this document requests your critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	ULTRIX-32
DECUS	UNIBUS
MASSBUS	VAX
PDP	VMS
ULTRIX	VT
ULTRIX-11	digital ™

UNIX is a trademark of AT&T Bell Laboratories.

Information herein is derived from copyrighted material as permitted under a license agreement with AT&T Bell Laboratories.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California. We acknowledge the Electrical Engineering and Computer Sciences Departments at the Berkeley Campus of the University of California for their role in its development.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. We acknowledge the following individuals for their role in its development:

Eric Allman, Ken Arnold, Ozalp Babaoglu, Scott B. Baden, Jerry Berkman, John Breedlove, Earl T. Cohen, Robert P. Corbett, Mike Curry, Steve Feldman, Tom Ferrin, John Foderaro, Susan L. Graham, Charles Haley, Robert R. Henry, Andy Hertzfeld, Mark Horton, S.C. Johnson, William Joy, Howard Katseff, Peter Kessler, Jim Kleckner, J.E. Kulp, James Larus, Kevin Layer, Mike Lesk, Steve Levine, Jeff Levinsky, Louise Madrid, M. Kirk McKusick, Colin L. McMaster, Mikey Olson, Geoffrey Peck, Ed Pelegri-Llopart, Rob Pike, Dave Presotto, John F. Reiser, Asa Romberger, Bill Rowan, Jeff Schreibman, Eric P. Scott, Greg Shenaut, Eric Shienbrood, Kurt Shoens, Keith Sklower, Helge Skrivervik, Al Stanberger, Ken Thompson, Michael C. Toy, Richard Tuck, Bill Tuthill, Mike Urban, Edward Wang, David Wasley, Joseph Weizenbaum, Jon L. White, Glenn Wichman, Niklaus Wirth.

The ULTRIX-32 Installation Guide

Table of Contents

Preface	ix
1. Format of the Installation Guide.....	ix
2. Additional References.....	x
3. Format Conventions	xi
4. The ed Command Summary Sheet	xiii
SECTION 1.0 Introduction.....	1
1.1 Hardware Supported.....	1
1.2 Distribution Format.....	1
1.3 Backup the Distribution Medium.....	1
1.3.1 Magnetic Tape Distribution Medium.....	1
1.3.2 RA60 and RK07 Distribution Media.....	2
1.3.3 RL02 Distribution Medium.....	2
1.4 ULTRIX-32 Devices: Block and Raw	2
1.5 newfs.....	2
SECTION 2.0 Bootstrap Procedure.....	3
2.1 Bootstrap Procedure for VAX-11/730 from Tape	5
2.1.1 Copy the Miniroot Filesystem.....	6
2.1.2 Boot from the Miniroot Filesystem	7
2.1.2.1 Explanation of System Boot Messages.....	8
2.1.2.2 Boot from the Miniroot filesystem, Continued	9
2.1.3 Restore the Root Filesystem.....	9
2.1.4 Update the Console Cassette	10
2.1.5 Reboot the Completed Root Filesystem	12
2.1.6 Set up and Install the /usr Filesystem	13
2.2 Bootstrap Procedure for VAX-11/750 from Tape	19
2.2.1 Copy the Miniroot Filesystem.....	21
2.2.2 Boot from the Miniroot Filesystem	21
2.2.2.1 Explanation of System Boot Messages.....	23
2.2.2.2 Boot from the Miniroot Filesystem, Continued.....	23
2.2.3 Restore the Root Filesystem.....	24
2.2.4 Reboot the Completed Root Filesystem	25
2.2.5 Set up and Install the /usr Filesystem	27

vi **ULTRIX-32 Installation Guide**

2.3 Bootstrap Procedure for VAX-11/780 from Tape	33
2.3.1 Copy the Miniroot Filesystem	35
2.3.2 Boot from the Miniroot Filesystem	36
2.3.2.1 Explanation of System Boot Messages.....	37
2.3.2.2 Boot from the Miniroot Filesystem, Continued.....	37
2.3.3 Restore the Root Filesystem.....	38
2.3.4 Update the Console Floppy Diskette	39
2.3.5 Reboot the Completed Root Filesystem	40
2.3.6 Set up and Install the /usr Filesystem	42
2.4 Bootstrap Procedure for All VAXes from RA60	47
2.4.1 Boot the Root Filesystem.....	48
2.4.1.1 Explanation of the System Boot Messages.....	50
2.4.2 Set the Date and the Root Password	51
2.4.3 Backup the Distribution Medium.....	52
2.4.4 Update the Console Cassette or Floppy Diskette.....	53
2.4.4.1 Update the Console Cassette	53
2.4.4.2 Update the Console Floppy Diskette.....	55
2.5 Bootstrap Procedure for All VAXes from RK07	57
2.5.1 Boot the Root Filesystem.....	58
2.5.1.1 Explanation of System Boot Messages.....	60
2.5.2 Set the Date and the Root Password	61
2.5.3 Backup the Distribution Media	61
2.5.3.1 The First Distribution Pack (ROOT).....	61
2.5.3.2 The Second Distribution Pack (USERS).....	63
2.5.3.3 The Third Distribution Pack (ADDITIONAL SOFTWARE).....	65
2.5.4 Update the Console Cassette or Floppy Diskette.....	66
2.5.4.1 Update the Console Cassette	67
2.5.4.2 Update the Floppy Diskette.....	68
2.6 Bootstrap Procedure from RL02.....	71
2.6.1 Boot the Distributed Root Filesystem	73
2.6.1.1 Explanation of System Boot Messages.....	75
2.6.2 Copy the RL02 Distributed Root to System Disk.....	75
2.6.3 Set up and Install the /usr Filesystem	77
2.6.4 Update the Console Cassette.....	80
SECTION 3.0 System Setup.....	83
3.1 Initialize /etc/fstab	83
3.2 Automatic Crash Dumps.....	85

3.3	Make Device Special Files	85
3.4	Build New System Images	88
3.5	Test the New System.....	89
3.6	Terminal Configuration	90
3.7	Site Tailoring	92
3.8	Bring the System to Multi-user Mode	93
3.9	Add New Users	93
3.9.1	Separate Partition is Available.....	94
3.9.2	Separate Partition is not Available.....	95
3.9.3	Add the User Accounts.....	95
SECTION 4.0 Network Setup.....		97
4.1	System Configuration	97
4.1.1	The Configuration File	97
4.1.2	Configuring Multiple Networks	98
4.1.3	Configuring Pseudoterminals	98
4.2	Network Data Bases.....	100
4.2.1	Editing the /etc/rc.local File.....	101
4.2.2	Editing the /etc/hosts and /etc/networks Files	103
4.2.2.1	Sites Connected to the ARPANET.....	103
4.2.2.2	Sites Not Connected to the ARPANET.....	104
4.2.3	Editing the /etc/hosts.equiv File.....	106
4.2.4	Editing the /etc/ftpusers Files	107
4.2.4.1	The FTP Account.....	107
4.2.4.2	The /etc/ftpusers File.....	108
4.2.5	The etc/services File	108
4.2.6	The etc/protocols File.....	108
4.3	Routing and Gateways.....	109
4.3.1	Maintaining Routing Tables with /etc/routed.....	109
4.3.2	Using a Smart Gateway to Create a Routing Data Base.....	109
SECTION 5.0 Additional Setup Information.....		111
5.1	Fine Tuning the Disk Configuration	111
5.1.1	Disk Filesystem Partitions.....	111

viii ULTRIX-32 Installation Guide

5.1.2 Space Available.....	114
5.1.3 Layout Considerations.....	115
5.1.4 Filesystem Parameters.....	117
5.1.5 Implement the Layout.....	119
5.2 Set up the Line Printer System	119
5.3 Set up the Mail System	120
5.4 Set up a UUCP Connection	121
5.5 Additional Software	121
5.5.1 Extracting Additional Software from Second Tape.....	122
5.5.1.1 If /usr of System Disk is Available.....	123
5.5.1.2 If /usr of System Disk is Not Available.....	124
5.5.2 Extracting Additional Software from Third RK07	125
APPENDIX A	127
APPENDIX B	129
APPENDIX C	133

Preface

The *ULTRIX-32 Installation Guide* is written for people who are responsible for installing ULTRIX-32 software on a VAX system.

To install the ULTRIX-32 system, you must have system management experience, and know how to generate, maintain, and tune a midsized system. If you plan to tailor the system, you must understand the VAX processor in general, as well as your site's specific implementation.

In addition to this general knowledge, you must be familiar with the ed(1) line editor. For a quick reference of ed, see *The ed Command Summary Sheet*, also included in this document.

1. Format of the Installation Guide

This installation guide is divided into five sections: Introduction, Bootstrap Procedure, System Setup, Network Setup, Additional Setup Information. It also contains three appendixes. Below is a brief description of each section and the appendixes.

Introduction

This section provides basic information concerning the hardware supported, distribution medium backups, and block and raw devices.

Bootstrap Procedure

This section explains how to install the ULTRIX-32 operating system and bring the system up to single-user mode.

The section is divided into six subsections:

- Bootstrap Procedure for VAX-11/730 from Tape Distribution
- Bootstrap Procedure for VAX-11/750 from Tape Distribution
- Bootstrap Procedure for VAX-11/780 from Tape Distribution
- Bootstrap Procedure for All VAXes from RA60 Distribution
- Bootstrap Procedure for All VAXes from RK07 Distribution
- Bootstrap Procedure for All VAXes from RLO2 Distribution

You need only read the subsection that pertains to your particular VAX processor. For example, if your processor is a VAX-11/780, skip subsections 2.1 and 2.2, and read subsection 2.3.

System Setup

This section explains how to set up the ULTRIX-32 operating system once the

x ULTRIX-32 Installation Guide

bootstrap procedure has been completed. Topics include generating a new kernel with selected device drivers, terminal configuration, site tailoring, and bringing the system up to multi-user mode.

Network Setup

This section explains how to configure your ULTRIX-32 operating system for networking. Network Setup contains three main subsections: The Configuration File, Network Databases, and Routing and Gateways.

Additional Setup Information

This section provides additional system setup details you will need once your ULTRIX-32 system is up and running.

The subsections in Additional Setup Information describe how to: fine tune the disk configuration, set up the line printer system, set up the mail system, set up a UUCP connection, and the installation procedure for the Additional Software found on the second distribution tape and the third distribution RK07 pack.

Appendixes

Appendix A provides information on the contents of the distribution tape and the control status register addresses which the boot software and generic system use. Appendix A is for reference only.

Appendix B provides information on troubleshooting your installation. Should you encounter problems while installing the ULTRIX-32 operating system, including errors or warning messages, consult this appendix.

Appendix C provides instructions for a final attempt to install the ULTRIX-32 software from the tape distribution. Use Appendix C in the rare event that all previous attempts to install the system from tape fail.

2. Additional References

To install the ULTRIX-32 system, you will need the following manuals. They are also included in this binder.

- *Building an ULTRIX-32 System with Config*
- *UUCP Installation and Administration*
- *Guidelines for System Management*

You will also need the *ULTRIX-32 Programmer's Manual*.

3. Format Conventions

The *ULTRIX-32 Installation Guide* observes these conventions:

Conventions

- bold** Literals are printed in **bold** type. Literals are frequently used to indicate a specific command option and, therefore, should be entered exactly as printed.
- color** Examples are printed in color. Examples represent command sequences or information that you would enter from your terminal.
- command(n)** The manual section number containing the applicable description (*n*) is enclosed in parentheses after a command or file name. For example, `cat(1)` refers to the description of the `cat` command in section one of the *ULTRIX-32 Programmer's Manual*.
- <CTRL/X>** Terminal control characters are represented by `<CTRL/X>`, where *X* is a single character. Generate terminal control characters by holding down the CTRL key and entering the character.
- <CTRL/C>** Use `<CTRL/C>` to interrupt a process.
- <CTRL/U>** Use `<CTRL/U>` to erase the current line.
- <DELETE>** The DELETE key is represented by `<DELETE>`. Use `<DELETE>` to erase the last character typed.
- italics** Substitutable parameters are printed in *italics*.
- <RETURN>** The RETURN key is printed as `<RETURN>`. Press the RETURN key after entering the command sequence to invoke the commands.
- special type** Command and file names are printed in special type.

Data Entry Conventions

- case** The ULTRIX-32 system differentiates between uppercase and lowercase characters. Therefore, enter uppercase characters only where specifically indicated by an example or the command syntax.

xii ULTRIX-32 Installation Guide

comments Any comments are enclosed in parentheses to the right of, or below commands.

A number sign represents the superuser prompt.

>>> When in console mode, the console subsystem prompt is represented by three right angle brackets, >>>.

The ed Command Summary Sheet

This command sheet provides an outline of how to use the ed(1) line editor. It provides enough information for you to edit the files necessary to install your ULTRIX-32 system.

Concepts

To use ed you must understand a few concepts:

- 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.
- .** A dot (.) represents the current line. If you type a dot on a line by itself ed prints the current line.
- \$** A dollar sign (\$) represents the last line in the file. Thus, the command `.,$p` tells ed to print the current line through the last line of the file.
- ?** A question mark (?) is the ed prompt. A question mark indicates ed is ready for your next command. ed also prompts with a question mark if you issue a command incorrectly.
- <RETURN>** Remember to press the <RETURN> key after each ed command.
- Pressing the <RETURN> key on a line by itself causes ed to make the next line the current line. If you are already at the last line of the file and press <RETURN>, ed responds with a question mark.

The following is a table of the most commonly used ed commands, and a brief description of each.

Command	Action
---------	--------

- | | |
|-----------------------|---|
| ed <i>xxxx</i> | Edits the file <i>xxxx</i> . If the file does not exist, this command creates a new file named <i>xxxx</i> and edits it. |
| . = | Prints the number of the current line. This is useful for orientation. |
| [<i>n,m</i>]p | Prints the line(s) specified. <i>n</i> is the number of the first line you want to print and <i>m</i> is the number of the last line you want to print. If you do not specify a line number(s) the current line is printed by default. For example: <code>1,10p</code> prints the first 10 lines of the file. |
| - | Makes the previous line the current line. |

xiv ULTRIX-32 Installation Guide

- a** Appends (adds) text to the file. **a** adds the text after the current line. Terminate the appended text with a dot (.) on a line by itself.
- i** Inserts text above the current line. Terminate the inserted text with a dot (.) 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 the changes, always use the **w** command. Then use 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.

Sample ed Session

The following is a sample **ed** session with an example `/etc/fstab` file. In this example, a line is deleted, a line is modified, and a line is added. Note that each command is followed by `<RETURN>`.

```
# ed /etc/fstab
67                               (number of characters in the file)
<RETURN>
?                               (ed is ready for your next command)
1,$p                             (print the entire file)
/dev/hk0a:/:rw:1:1
/dev/hk0g:/usr:xx:1:2
/dev/hk1a:/usr/sys:rw:1:3
-                               (make the previous line the new current line)
/dev/hk0g:/usr:xx:1:2
s/xx/rw/p                       (substitute rw for xx on the current line)
/dev/hk0g:/usr:rw:1:2          (this is the modified line)
<RETURN>                       (make the next line the new current line)
/dev/hk1a:/usr/sys:rw:1:3
a                               (add text after the current line)
/usr/dev/hk1g:/usr/users:rw:1:4 (appended text)
.                               (terminate the appending mode)
<RETURN>
```

```

?                (ed is ready for your next command)
1,$p            (print the entire file)
/dev/hk0a:/:rw:1:1
/dev/hk0g:/usr:rw:1:2
/dev/hk1a:/usr/sys:rw:1:3
/usr/dev/hk1g:/usr/users:rw:1:4
i                (insert text above the current line)
/usr/dev/hk1g:/usr/mnt:rw:1:4 (inserted text)
.                (terminate the inserting mode)
1,$p            (print the entire file)
/dev/hk0a:/:rw:1:1
/dev/hk1a:/usr/sys:rw:1:3
/usr/dev/hk1g:/usr/mnt:rw:1:4
/usr/dev/hk1g:/usr/users:rw:1:4
-                (make the previous line the current line)
/usr/dev/hk1g:/usr/mnt:rw:1:4
dp              (delete the current line and
                print the new current line)
/usr/dev/hk1g:/usr/users:rw:1:4
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
w                (write the file (save the modifications))
100             (number of characters in the file)
q                (quit the ed editor)
#

```

For further information on ed read *A Tutorial Introduction to the UNIX Text Editor* in *ULTRIX-32 Supplementary Documents*, Volume I.

1. Introduction

This document explains how to install ULTRIX-32 software on your VAX system.

1.1. Hardware Supported

The ULTRIX-32 distribution can be installed on a VAX-11/730, VAX-11/750, or VAX-11/780 CPU with the following disks:

MASSBUS (except 11/730): RM03, RM05, RM80, RP06, RP07 (except 11/750)
UNIBUS: RK07, RA60, RA80, RA81
IDC (11/730, only): R80

The supported tape drives are:

MASSBUS (except 11/730): TE16, TU77, TU78
UNIBUS: TS11, TU80

1.2. Distribution Format

The basic distribution kit contains these items:

- Two 1600 bits/in magnetic tapes, nine RL02s, three RK07s, or one RA60
- One TU58 console cassette, or one RX01 console floppy disk

The console medium distributed with the system is for installation only. It is not suitable for use as the standard local console medium.

NOTE: The entire distribution may not fit on some VAX configurations. If you have limited disk space, you will need to decide which files are not critical to your installation, and omit them.

1.3. Backup the Distribution Medium

If you have the facilities, make backup copies of the distribution medium.

1.3.1. Magnetic Tape Distribution Medium

The tapes are 9-track 1600 bits/in and contain 512-byte records followed by 10240-byte records. The records are interspersed with tape marks; end-of-tape is signaled by a double end-of-file.

The basic bootstrap material is in three files at the beginning of the boot tape. The first file on the tape contains preliminary bootstrapping programs and a TP image containing several stand-alone programs. The TP image is followed by the second file, which is a binary image of a 300 kilobyte mini-root filesystem. The third file is a full dump of the root filesystem (see `dump(8)`).

The remaining files on the first and second tapes contain tape archive images (see

2 ULTRIX-32 Installation Guide

tar(1)). The fourth file on the boot tape contains the kernel objects. The fifth tape file contains most of the files in the /usr filesystem.

The directories on the second tape (labeled Additional Software) are: /usr/man, /usr/games, /usr/doc, /usr/lib/vfont, /usr/lib/learn, /usr/lib/lisp, and /usr/new.

1.3.2. RA60 and RK07 Distribution Media

You should make backup copies of the RA60 and RK07 distribution disks onto scratch disks, and then run your system off the newly copied disk(s). The original distribution disk(s) should be saved for backup purposes.

Subsection 2.4.3 explains how to backup the RA60 disks, and subsection 2.5.3 explains how to backup the RK07 disks.

1.3.3. RL02 Distribution Medium

It is not necessary to make backup copies of the RL02 distribution disks because you will be running off of other, larger disks. This leaves the original distribution disks free for backup purposes.

1.4. Devices: Block and Raw

ULTRIX-32 software makes a distinction between block and raw (character) devices.

A raw device is designated with an r. For example, an fsck on a raw device called hp0a is specified:

```
# fsck /dev/rhp0a
```

It is important that you do not indiscriminately alter the command syntax given in this document with regard to block versus raw device specification.

1.5. newfs

Throughout the installation you will be using the newfs(8) command. Always save the super-block numbers that newfs generates; they may be valuable in an emergency. A good storage place is taped to the outside of the disk cover (if the disk is removable).

2. Bootstrap Procedure

Section 2 explains how to bootstrap ULTRIX-32 software.

If you are already running UNIX software, save your existing files before beginning the bootstrap. The ULTRIX-32 system and the 4.2 BSD UNIX system use a totally different file system organization than previous versions of UNIX systems; therefore, you must rebuild the file system format before restoring the data. The easiest way to save the current files on tape is to do a full dump and then restore under the new system.

The subsections in this section describe the bootstrap procedure for installing the ULTRIX-32 software on a VAX-11/730, VAX-11/750, and VAX-11/780 processor from distribution tapes, and distribution disks (RA60s, RK07s, and RL02s). Go to the subsection pertaining to your particular VAX system and/or distribution media, and follow the procedures outlined. For example, if your system is a VAX-11/780 and your distribution media is magnetic tape, then go directly to subsection 2.3.

Table 2.1 lists each subsection, the type of media, and the VAX system each subsection describes.

Table 2.1 Bootstrap Procedure Subsections

Subsection	Media	System
2.1	mag. tape	VAX-11/730
2.2	mag. tape	VAX-11/750
2.3	mag. tape	VAX-11/780
2.4	RA60	all VAX systems
2.5	RK07	all VAX systems
2.6	RL02	all VAX systems

2.1. Bootstrap Procedure for VAX-11/730 from Tape Distribution

Subsection 2.1 explains the bootstrap procedure for installing the ULTRIX-32 software on a VAX-11/730, from the distribution tape.

This procedure should take between one and one and one half hours.

Getting Started

Before starting the bootstrap procedure, you must be familiar with your system configuration. If your system is under a Digital Field Service Contract, then you can obtain detailed configuration information from the Configuration Worksheet section of the *Site Management Guide*.

You will need the following equipment:

- The two ULTRIX-32 distribution tapes
- The ULTRIX-32 distribution cassette
- One blank cassette
- Formatted disk pack(s)
- VAX-11/730 system with a tape drive and disk drive(s)

Follow these steps to prepare the system for booting from tape:

- 1) If your disk packs are removable (RK07, RA60), load a formatted disk pack into disk drive 0. (Digital Equipment Corporation disk packs are shipped preformatted). Load any additional disk packs you may need as well, but the disk pack in drive 0 will be the root (or system) disk.

Even if your pack is nonremoveable, be sure your system disk pack is spinning and that the READY light is lit, even if your disk pack is nonremovable.

- 2) Load the first distribution tape (labeled BOOT TAPE) on tape drive 0. Be sure the tape drive is ON-LINE and that the tape is rewound to load point (BOT).
- 3) Insert the distribution cassette into the console TU58 disk drive on the outside of the processor (drive 0).
- 4) On the processor itself, set the restart switch to OFF and the rotary key switch to LOCAL.

General Steps

The tape bootstrap procedure to create a working system consists of the following general steps:

- 1) Copy a miniroot filesystem from the tape onto the swap area of the disk.

6 ULTRIX-32 Installation Guide

- 2) Boot the ULTRIX-32 system on the miniroot.
- 3) Restore the full root filesystem.
- 4) Update a console cassette for bootstrapping.
- 5) Reboot the completed root filesystem.
- 6) Build and restore the /usr filesystem from tape.

The next few subsections describe these steps in detail. References to disk drives are of the form $xx(n,m)$ and references to files on tape drives are of the form $yy(n,m)$ where xx and yy are one of the names in the table below.

n is usually 0 because it is equal to the unit number of the disk or tape drive; it specifies the logical unit number.

m is always specified in the command lines of this document. For a disk, m is the partition (in the range 0 - 7). For a tape, m is the file number on the tape.

Table 2.1.1 is a quick reference table for the disk and tape drive ULTRIX-32 system naming conventions.

Table 2.1.1 ULTRIX-32 Disk and Tape Drive Naming Conventions

Disk	xx	Tape	yy
RA60	ra	TS11	ts
RA80	ra	TU80	ts
RA81	ra		
RK07	hk		
R80 (IDC)	rb		

If you have a DECwriter III (LA120), you must set up your console terminal before proceeding with the installation.

To set up the console, hold down <SETUP> and press the D key simultaneously until the LED display on the console (at the upper left-hand corner of the keys) is 0 (zero).

2.1.1. Copy the Miniroot Filesystem

The first step is to run the program *copy*, which copies the miniroot filesystem from the second tape file of the first distribution tape into the second partition of the disk. This filesystem will serve as the base for creating the actual root filesystem.

The system must be halted; if it is not, type:

```
^P
>>>H
```

The console subsystem prompts with >>>. To load the copy program, type the following commands in response to the prompt. If you make a mistake typing, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

```
>>>I                               (initialize the processor)
>>>L DD0:COPY                       (boot the cassette)
                                     (this takes about two minutes)
>>>S 2                             (start the copy program)
From: yy(n,1)                       (from the distribution tape)
To: xx(n,1)                         (to the disk pack)
                                     (this takes about one minute)

Copy completed: 300 records copied
From: ^P                             (<CTRL/P> returns you to the console
                                     subsystem)
?02 PC=xxxxxxxx                    (program counter)
>>>I                               (initialize the processor)
>>>
```

You now have the set of tools necessary to create a root filesystem and restore the filesystem contents from tape.

2.1.2. Boot from the Miniroot Filesystem

The second step is to load the bootstrap program and boot the ULTRIX-32 miniroot filesystem which you just copied to the disk:

```
>>>L DD0:BOOT                       (boot the cassette)
                                     (this takes about one minute)
>>>S 2                             (start the boot program)
Boot

: xx(n,1)vmunix                     (boot the ULTRIX-32 kernel off the
                                     miniroot)
                                     (this takes about one minute)

.
.
.
```

The standalone boot program then reads the ULTRIX-32 miniroot system (which you just created via the copy program) from the disk. The system then starts, displaying the following messages:

8 ULTRIX-32 Installation Guide

215564+64088+69764 start 0xf98 (these numbers are approximate)
ULTRIX-32 *date and time* (this takes a minute)
real mem = *xxxxxx*
avail mem = *yyy*

.
.
.

information about available devices

.
.
.

root device?

2.1.2.1. Explanation of System Boot Messages

The first three numbers printed by the bootstrap programs are the sizes of different parts of the system (text, initialized and uninitialized data).

The ULTRIX-32 system then runs for the first time, beginning with a banner identifying the release and version of the system that is in use, and the date it was compiled.

The mem messages give the amount of real (physical) memory and the memory available to user programs in bytes. For example, if your machine has 4mb of memory, then *xxxxxx* will be 4190208, 4096 bytes less than 4mb. The system reserves the last 4096 bytes of memory for use in logging system messages and does not count it as part of real memory. Available memory is the amount of physical memory left over for user processes after the ULTRIX-32 software has been loaded.

Information about the system is shown next, followed by messages showing what devices were found on the current processor. These messages are described in *autoconf(4)*. The distributed system may not find all the communications devices (dmf's and dz's), or mass storage peripherals if you have more than two of one device. This is corrected later when you create a description of your machine from which to configure the ULTRIX-32 system. These messages contain much of the information that will be used in creating the configuration. In a correctly configured system most of the information present in the configuration description is printed out at boot time as the system verifies that each device is present.

You may receive a warning message about the date. The system is telling you that it realizes the date and time currently stored in memory are wrong. Do not worry about this message; you will be changing the date and time later (subsection 2.1.6).

2.1.2.2. Boot from the Miniroot Filesystem, Continued

The “root device?” prompt asks for the name of the root filesystem to use:

```

.
.
.

root device? xx0*
erase ^?, kill ^U, intr ^C
#

```

The response *xx0** supplies two pieces of information. First, *xx0* indicates the disk it is running on is drive 0 of type *xx*. Second, the *** indicates the system is running atop the paging area. The *** is very important because it prevents the system from attempting to page on top of itself.

NOTE: If at any time the miniroot filesystem requires rebooting, be sure to copy the miniroot first (see subsection 2.1.1).

The shell executes `/.profile` when it starts, which in turn generates the erase ... message. This message reminds you that `<DELETE>` is the erase character, `<CTRL/U>` is the erase line character (kill), and `<CTRL/C>` is the interrupt character.

2.1.3. Restore the Root File System

The ULTRIX-32 system is now running. The `#` is the prompt from the shell, and lets you know that you are the superuser, whose login name is root.

The third step is to create the root filesystem. To do this, execute the `xtr` shell script by typing:

```
# disk=xx0 type=dddd tape=yy xtr
```

where *xx0* is the name of the disk drive (unit 0), *dddd* is the type of disk drive on which the root filesystem is to be restored and *yy* is the name of the tape drive from which the distribution tape is loaded (see Table 2.1.2). Table 2.1.2 is a reference of the disk and tape drive ULTRIX-32 system naming conventions.

Table 2.1.2 ULTRIX Disk and Tape Drive Naming Conventions

Disk	xx	dddd	Tape	yy
RA60	ra	ra60	TS11	ts
RA80	ra	ra80	TU80	ts
RA81	ra	ra81		
RK07	hk	rk07		
R80 (IDC)	rb	rb80		

10 ULTRIX-32 Installation Guide

For example, if you have an RK07 disk drive and a TU80 tape drive, type:

```
# disk=hk0 type=rk07 tape=ts xtr
```

The xtr script generates messages regarding the construction of the filesystem and the restoration of the tape contents. You may receive a warning giving the inode blocks/cylinder group and its implications; disregard this message. The messages will eventually terminate with the following:

```
Root filesystem extracted
```

```
If this is a 780, update floppy  
If this is a 730, update the cassette  
#
```

Check the consistency of the root filesystem by typing:

```
# fsck /dev/rxx0a (xx is from Table 2.1.2)
```

The output from fsck should look similar to:

```
** /dev/rxx0a  
** Last Mounted on /  
** Phase 1 - Check Blocks and Sizes  
** Phase 2 - Check Pathnames  
** Phase 3 - Check Connectivity  
** Phase 4 - Check Reference Counts  
** Phase 5 - Check Cyl groups  
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

2.1.4. Update the Console Cassette

The fourth step is to build an ULTRIX-32 boot cassette by copying some files from the local console cassette (that is, the cassette which came with your hardware), adding some files from the ULTRIX-32 system and writing them all to a blank cassette.

This new boot cassette will allow the ULTRIX-32 system to reboot automatically whenever necessary.

First, remove the distribution cassette from the external TU58 drive (drive 0) and insert a blank cassette. Be sure the blank cassette is NOT write protected (the record tab is in the left-most position).

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

Then, change into the directory where the console cassette information is stored:

```
# cd /sys/cassette
```

Next set up the default boot device. If you intend to use an IDC R80 as your primary root type:

```
# cp defboo.rb defboo.cmd
```

If you intend to use an RK07 as your primary root type:

```
# cp defboo.hk defboo.cmd
```

If you intend to use a drive on a UDA50 as your primary root type:

```
# cp defboo.ra defboo.cmd
```

To build the updated boot cassette, type:

```
# make update
```

```
.  
.
.
```

(this takes about 25 minutes)

lists all of the files on the cassette

```
.  
.
.
```

Console cassette update complete.
The new cassette should be installed in the internal
tu58 slot (Drive 1).

```
#  
#
```

Store the distribution cassette and the local console cassette in a safe place. The original local console will be required to boot VAX/VMS systems. Remove the boot cassette you just built from the external drive, and insert it in the internal TU58 cassette drive (drive 1).

At this point, you can make more copies of the new boot cassette by using the `fcopy` maintenance command (see `arff(8V)`).

12 ULTRIX-32 Installation Guide

2.1.5. Reboot the Completed Root File System

The fifth step is to reboot the system. To do this, type the following commands. If you make a mistake typing, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

```
# sync                (flush incore buffers to disk)
# sync
# ^P                  (<CTRL/P> returns you to the console
                      subsystem)
?02 PC=xxxxxxxx      (program counter)
>>>                  (initialize processor)
>>> B ANY            (boot program is eventually loaded)
```

```
.
.
.
```

boot program messages appear

```
.
.
.
```

```
Boot
: xx(x,0)vmunix      (boot the ULTRIX-32 kernel off root)
                      (this takes a minute)
215564+64088+69764 start 0xf98 (these numbers are approximate)
```

```
.
.
.
```

ULTRIX-32 *date and time*

```
real mem = xxxxxx
avail mem = yyy
```

```
.
.
.
```

information about available devices

```
.
```

```

.
.
root device? xx0
erase ^?, kill ^U, intr ^C
#

```

If the system does not reboot properly, see Appendix C for help with troubleshooting.

The system is running in single-user mode on the installed root filesystem. The # prompt is from the ULTRIX-32 operating system, and tells you that you are the superuser.

The next subsection tells how to complete the installation of distributed software for the /usr filesystem.

2.1.6. Set up and Install the /usr Filesystem

The last step is to build and restore the /usr filesystem.

First, set a shell variable to the name of your disk drive so the commands given will work regardless of the drive you have. To do this, find your disk drive in the following list and type the command to the right:

Disk Drive	Command
RA60	disk=ra
RA80	disk=ra
RA81	disk=ra
RK07	disk=hk
R80 (IDC)	disk=rb

For example, if you have an RK07, type:

```
# disk=hk
```

The next two commands select a partition for /usr. If you have more drives available, see subsection 5.1 (Fine Tuning the Disk Configuration) to help decide if some other partition might be more appropriate.

The partitions in the list below are those most appropriate in size. Find your disk in the list and type the commands to the right:

14 ULTRIX-32 Installation Guide

Disk	Commands
RA60	name=ra0h; type=ra60
RA80	name=ra0h; type=ra80
RA81	name=ra0h; type=ra81
RK07	name=hk0g; type=rk07
R80	name=rb0h; type=rb80

For example, if you have an RK07, type:

```
# name=hk0g; type=rk07
```

Next, find your tape drive in the following list and type the commands to the right.

Tape Drive	Commands
TS11	cd /dev; MAKEDEV ts0; sync
TU80	cd /dev; MAKEDEV ts0; sync

For example, if you have a TU80, type:

```
# cd /dev; MAKEDEV ts0; sync
```

Now set the date and time:

```
# date yymmddhhmm
```

Where:

yy equals the last two digits of the year
mm equals the number of the month of the year
dd equals the number of the day of the month
hh equals the hour of the day (based on 24 hours)
mm equals the minutes of the hour

(the system echos the new date and time)

Next, set the superuser password. The password you select must contain at least six characters:

```
# passwd root                (set password for superuser)
New password:                (password will not echo)
Retype new password:
#
```


16 ULTRIX-32 Installation Guide

If you have an RK07 as your system disk, type:

```
# umount /dev/hk1a
```

Continue with the following for all system disks, including RK07s:

```
# umount /dev/${name}          (umount /usr)
```

Check the consistency of the /usr filesystem by typing:

```
# fsck /dev/r${name}
```

The output from fsck should look similar to:

```
** /dev/rxx0h
** Last Mounted on /usr
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

If your system disks are RK07s, check the consistency of the /usr/sys file on drive 1 by typing:

```
# fsck /dev/rhk1a
```

The output from *fsck* should look similar to:

```
** /dev/rhk1a
** Last Mounted on /usr/sys
** Phase 1 - Check Blocks and sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
864 files, 3745 used, 3678 free (134 frags, 433 blocks)
```

If there are inconsistencies in the filesystems, the system may prompt you for corrective action; see *fsck - the UNIX File System Check Program* for further information. If serious problems exist, try the extractions again, starting with the newfs `${name}` `${type}` command above.

Now remount the /usr filesystem by typing:

```
# /etc/mount /dev/${name} /usr
```

If your system disks are RK07s, you also need to remount the /usr/sys filesystem. To do this type:

```
# /etc/mount /dev/hk1a /usr/sys
```

You have now extracted all the data on the first distribution tape. Unload the first tape from the tape drive, and store it in a safe place. You will be extracting the files from the second distribution tape later in subsection 4.5.

The ULTRIX-32 operating system is bootstrapped and running in single-user mode. See section 3 (System Setup) for information on how to further set up your system.

2.2. Bootstrap Procedure for VAX-11/750 from Tape Distribution

Subsection 2.2 explains the bootstrap procedure for installing the ULTRIX-32 software on a VAX-11/750, from the distribution tape.

This procedure should take between half an hour and a full hour to complete.

Getting Started

Before starting the bootstrap procedure, you must be familiar with your system configuration. You must know which hardware devices are connected to which controllers, naming conventions, unit numbers, and so forth. If your system is under a Digital Service Contract, then you can obtain detailed configuration information from the Configuration Worksheet section of the *Site Management Guide*.

You will need the following equipment:

- The two ULTRIX-32 distribution tapes
- The ULTRIX-32 distribution cassette
- Formatted disk pack(s)
- VAX-11/750 system with a tape drive and a disk drive(s)

Follow these steps to prepare your system for booting from tape:

- 1) Load a formatted disk pack into disk drive 0 if your disk packs are removable (Digital Equipment Corporation disk packs are shipped preformatted). Load any additional disk packs you may need as well. Remember, the disk pack on drive 0 will be the root (or system) disk.

Even if your disk pack is nonremovable, be sure your system disk pack is spinning and that the READY light is lit.

- 2) Load the first distribution tape (labeled BOOT TAPE) on tape drive 0. Be sure the tape drive is ON-LINE and READY, and that the tape is rewound to load point (BOT).
- 3) Insert the distribution cassette into the console TU58 disk drive on the front of the processor (drive 0).
- 4) On the processor itself, 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 the installation.

20 ULTRIX-32 Installation Guide

General Steps

The tape bootstrap procedure to create a working system consists of these steps:

- 1) Copy a miniroot filesystem from the tape onto the swap area of the disk.
- 2) Boot the ULTRIX-32 system on the miniroot.
- 3) Restore the full root filesystem.
- 4) Reboot the completed root filesystem.
- 5) Build and restore the /usr filesystem from tape.

The next few subsections describe these steps in detail. References to disk drives are of the form $xx(n,m)$ and references to files on tape drives are of the form $yy(n,m)$ where xx and yy are one of the names in Table 2.2.1.

For UNIBUS devices, n is usually 0 because it is equal to the unit number of the disk or tape drive.

For MASSBUS devices, n is computed as follows:

$$n = 8 * (\text{mba}) + (\text{device unit number})$$

For example, if drive 0 were on the first MASSBUS adapter (mba0) then n would be 0; if drive 0 were on the second MASSBUS adapter (mba1) then n would be 8.

The m is always specified in the command lines of this document. For a disk, m is the partition (in the range 0-7). For a tape, m is the file number on the tape.

Table 2.2.1 is a quick reference table for the disk and tape drive ULTRIX-32 system naming conventions.

Table 2.2.1 Disk and Tape Drive Naming Conventions

Disk	xx	Tape	yy
RA60	ra	TE16	ht
RA80	ra	TS11	ts
RA81	ra	TU77	ht
RK07	hk	TU78	mt
RL02	rl	TU80	ts
RM03	hp		
RM05	hp		
RM80	hp		
RP06	hp		

If you have a DECwriter III (LA120), you must set up your console terminal before proceeding with the installation.

To set up the console, hold down <SETUP> and press D simultaneously until the

LED display on the console (at the upper left-hand corner of the keys) is 0 (zero).

2.2.1. Copy the Miniroot Filesystem

The first step uses the *copy* program to copy the miniroot filesystem from the second tape file of the first distribution tape into the second partition of the disk. This filesystem will serve as the base for creating the actual root filesystem.

The system must be halted; if it is not, type:

```
^P
>>>H
```

The console subsystem prompts with >>>. To load the copy program, type the following in response to the prompt. If you make a typing mistake, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

```
>>>I                (initialize the processor)
>>>B DDAO          (boot the cassette)
%%                (signifies the beginning and completion
                  of diagnostics)
= copy            (start the copy program)
                  (this takes a few minutes)
From: yy(n,1)    (from the distribution tape to the disk pack)
To: xx(n,1)
                  (this takes about five minutes)

Copy completed: 300 records copied
From: ^P          (<CTRL/P> returns you to the console
                  subsystem)

xxxxxxxx 02       (program counter)
>>> I           (initialize the processor)
>>>
```

2.2.2. Boot from the Miniroot Filesystem

You now have the minimal set of tools necessary to create a root filesystem and restore the filesystem contents from tape.

The second step is to load the bootstrap program and boot the ULTRIX-32 miniroot system which you just copied to the disk. To do this, type:

22 ULTRIX-32 Installation Guide

```
>>> B DDA0                (boot the cassette)
                             (this takes about one minute)

%%

= boot                    (start the boot program)
                             (this takes about one minute)

Boot
: xx(n,1)vmunix          (boot the ULTRIX-32 kernel off the
                           miniroot)

.
.
.
```

The standalone boot program then reads the ULTRIX-32 miniroot system (which you just created with the copy program) from the disk. The system then starts, displaying the following messages:

```
.
.
.

Updating 11/750 micro code... (appears if you are running
                              revision 5 or greater)

215564+64088+69764 start 0xf98 (these numbers are approximate)
ULTRIX-32 (date and time)    (this takes a minute)
real mem = xxxxxx
avail mem = yy

.
.
.

information about available devices

.
.
.

root device?
```

2.2.2.1. Explanation of System Boot Messages

The first three numbers printed by the bootstrap programs are the sizes of different parts of the system (text, initialized and uninitialized data).

The ULTRIX-32 system then runs for the first time, beginning with a banner identifying the release and version of the system that is in use and the date it was compiled.

The mem messages give the amount of real (physical) memory and the memory available to user programs in bytes. For example, if your machine has 4mb of memory, then `xxxxxx` will be 4190208, 4096 bytes less than 4mb. The system reserves the last 4096 bytes of memory for use in logging system messages and does not count it as part of real memory. Available memory is the amount of physical memory left over for user processes after the ULTRIX-32 software has been loaded.

Information about the system is shown next, followed by messages showing what devices were found on the current processor. These messages are described in `autoconf(4)`. The distributed system may not have found all the communications devices (`dmfs` and `dzs`), or mass storage peripherals you have if you have more than two of one device. This is corrected later when you create a description of your machine from which to configure the ULTRIX-32 system. These messages contain much of the information that will be used in creating the configuration. In a correctly configured system most of the information present in the configuration description is printed out at boot time as the system verifies that each device is present.

You may receive a warning message about the date. The system is telling you that it realizes the date and time currently stored in memory are wrong. Do not worry about this message; you will be changing the date and time later (subsection 2.2.5).

2.2.2.2. Boot from the Miniroot Filesystem, Continued

The root device? prompt asks for the name of the root filesystem to use:

```

.
.
.

root device? xx0*
erase ^?, kill ^U, intr ^C
#
```

The `xx0*` response supplies two pieces of information. First, `xx0` indicates the disk is running on drive 0 of type `xx`. Second, the `*` indicates the system is running atop the paging area. The `*` is very important because it prevents the system from attempting to page on top of itself.

NOTE: If at any time the miniroot requires rebooting, be sure to copy the miniroot first (see subsection 2.2.1).

24 ULTRIX-32 Installation Guide

The shell executes `/.profile`, which in turn generates the “erase ...” message. This message informs you that `<DELETE>` is the erase character, `<CTRL/U>` is the erase line character (kill), and `<CTRL/C>` is the interrupt character.

The ULTRIX-32 system is now running. The `#` is the prompt from the shell, and lets you know that you are the superuser, whose login name is `root`.

2.2.3. Restore the Root Filesystem

The third step is to create the root filesystem. To do this, run the `xtr` shell script by typing:

```
# disk=xx0 type=dddd tape=yy xtr
```

where `xx0` is the name of the disk drive (unit 0), `dddd` is the type of disk drive on which the root filesystem is to be restored and `yy` is the name of the tape drive from which the distribution tape is mounted (see Table 2.2.2). Table 2.2.2 is a quick reference of the disk and tape drive ULTRIX-32 system naming conventions.

Table 2.2.2 ULTRIX-32 Disk and Tape Drive Naming Conventions

Disk	xx	dddd	Tape	yy
RA60	ra	ra60	TE16	ht
RA80	ra	ra80	TS11	ts
RA81	ra	ra81	TU77	ht
RK07	hk	rk07	TU58	tu
RM03	hp	rm03	TU80	ts
RM05	hp	rm05		
RM80	hp	rm80		
RP06	hp	rp06		

For example, if you have an `RK07` disk drive and a `TU80` tape drive, type:

```
# disk=hk0 type=rk07 tape=ts xtr
```

The `xtr` script generates messages regarding the construction of the filesystem and the restoration of the tape contents. You may receive a warning giving the inode blocks/cylinder and its implications; disregard this message. The messages eventually terminate with the following:

(this takes about five minutes)

Root filesystem extracted

If this is a 780, update floppy

If this is a 730, update the cassette

#

NOTE: The VAX-11/750 does not use console media for rebooting. Therefore, you do not need to build a new console boot cassette.

Check the consistency of the root filesystem by typing:

fsck /dev/rxx0a (xx is found in Table 2.2.2)

The output from fsck should look similar to:

```
** /dev/rxx0a
** Last Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

2.2.4. Reboot the Completed Root Filesystem

The fourth step is to reboot the system. To do this, type the following commands. If you make a mistake typing, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

```
# sync (flush incore buffers to disk)
# sync
# ^P (<CTRL/P> return you to console
subsystem)
xxxxxxx 02 (program counter)
>>> (initialize processor state)
>>> B/3 ddcu (see the following table for an
explanation of ddcu)
```

Table 2.2.3 is a quick reference to help you fill the variable *ddcu* in the command above. Find your kind of disk drive in the table, and use the *ddcu* to the right.

26 ULTRIX-32 Installation Guide

Table 2.2.3 Disk Drives and their Device Type Codes

Disk Drive	<i>ddcu</i>	
RA60	DU <i>cu</i>	(<i>c</i> is usually A for each
RA80	DU <i>cu</i>	of the drives listed. <i>u</i> is
RA81	DU <i>cu</i>	equal to the drive number. See
RK07	DM <i>cu</i>	below for further information.)
RM03	DB <i>cu</i>	
RM05	DB <i>cu</i>	
RM80	DB <i>cu</i>	
RP06	DB <i>cu</i>	

ddcu is the general format for device names, and is computed as follows:

dd The first two characters are the disk drive device designator.

c The third character is the controller to which the boot drive is connected. It is usually A. If the controller unit number is 0 then *c* is A, if the unit number is 1 then *c* is B, and if the unit number is 2 then *c* is C.

u The fourth character is the disk drive unit number. This number can be found on the front panel of the disk drive unit.

After supplying the correct *ddcu*, continue rebooting the system as follows:

%% (signifies beginning and successful completion of diagnostics)

(boot program is eventually loaded)

.
.
.

boot program messages appear

.
.
.

Boot

: xx(n,0)vmunix

(boot the ULTRIX-32 kernel off root)

(this takes a minute)

215564+64088+69764 start 0xf98

(these numbers are approximate)

.
.

```
real mem = xxx
avail mem = yyy
```

```
.
.
.
information about available devices
```

```
.
.
.
root device? xx0
erase ^?, kill ^U, intr ^C
#
```

If the system does not reboot properly, see Appendix C for help with troubleshooting.

The system is running in single-user mode on the installed root filesystem. The # prompt is from the ULTRIX-32 operating system, and tells you that you are the superuser.

The next subsection tells how to complete the installation of distributed software for the /usr filesystem.

2.2.5. Set up and Install the /usr Filesystem

The last step is to build and restore the /usr filesystem.

First, set a shell variable to the name of your disk drive so the commands given will work regardless of your drive. To do this, find your disk drive in the following list, and enter the command to the right:

<u>Disk Drive</u>	<u>Command</u>
RA60	disk=ra
RA80	disk=ra
RA81	disk=ra
RK07	disk=hk
RM03	disk=hp
RM05	disk=hp
RM80	disk=hp
RP06	disk=hp

28 ULTRIX-32 Installation Guide

For example, if you have an RK07, type:

```
# disk=hk
```

The next two commands select a partition for /usr. If you have more drives available, see subsection 5.1 (Fine Tuning the Disk Configuration) to help decide if some other partition might be more appropriate.

The partitions below are those most appropriate in size. Find your disk in the following list and type the commands to the right:

Disk	Commands
RA60	name=ra0h; type=ra60
RA80	name=ra0h; type=ra80
RA81	name=ra0h; type=ra81
RK07	name=hk0g; type=rk07
RM03	name=hp0g; type=rm03
RM05	name=hp0g; type=rm05
RM80	name=hp0g; type=rm80
RP06	name=hp0g; type=rp06

For example, if you have an RK07, then you would type name=hk0g; type=rk07.

Next, find your tape drive in the following list, and enter the commands to the right:

Tape Drive	Commands
TE16	cd /dev; MAKEDEV ht0; sync
TS11	cd /dev; MAKEDEV ts0; sync
TU45	cd /dev; MAKEDEV ht0; sync
TU77	cd /dev; MAKEDEV ht0; sync
TU78	cd /dev; MAKEDEV mt0; sync
TU80	cd /dev; MAKEDEV ts0; sync

For example, if you have a TU80, type:

```
# cd /dev; MAKEDEV ts0; sync
```

Now set the date and time:

```
# date yymmddhhmm
```

Where:

yy equals the last two digits of the year
mm equals the number of the month of the year
dd equals the number of the day of the month
hh equals the hour of the day (based on 24 hours)
mm equals the minutes of the hour

(the system echos the new date and time)

Next, set the superuser password. The password you select must contain at least six characters:

```
# passwd root                (set password for superuser)
New password:                (password will not echo)
Retype new password:
```

Now create an empty user filesystem by typing:

```
# newfs ${name} ${type}      (remember to type the braces ({ }))
.                             (this takes a few minutes)
.
.
```

newfs lists all the filesystem partitions for /usr. Save the super-block numbers. Tape them to the outside of the disk pack cover. If your pack is nonremovable, tape the numbers to the outside of the drive.

Then continue with these commands:

```
# mt rew                    (rewind the magnetic tape)
# mount /dev/${name} /usr   (mount the usr filesystem)
                             (remember to type the braces)
# cd /usr                   (make /usr the current directory)
# mkdir sys                 (make directory for system objects)
```

If your system disk is an RK07, be sure your second pack is loaded onto the second

30 ULTRIX-32 Installation Guide

drive (drive 1), and spinning, and the READY light is lit. Then type:

```
# newfs hk1a rk07
```

newfs prints all the filesystem partitions for /usr/sys. Save these numbers as you did above. Then enter this command:

```
# mount /dev/hk1a /usr/sys
```

Continue with the following commands for all system disks, including RK07s:

```
# cd sys                (make /usr/sys the current directory)
# mt fsf 3              (wind the magnetic tape forward three
                        files; this takes a minute)
# tar xpbf 20 /dev/rmt12 (extract the system source)
                        (this takes 5-10 minutes)
# cd ..                (make /usr the current directory)
# mt fsf
# tar xpbf 20 /dev/rmt12 (extract all of usr except /usr/src)
                        (this takes 15-20 minutes)
# cd /                  (make / (root) the current directory)
# chmod 755 /usr/sys
```

If your system disks are RK07s, enter the following:

```
# umount /dev/hk1a
```

Then continue with the following for all system disks, including RK07s:

```
# umount /dev/${name}  (umount /usr; remember to type the
                        braces)
#
```

Check the consistency of the /usr filesystem by typing:

```
# fsck /dev/r${name}
```

The output from `fsck` should look similar to:

```
** /dev/rxx0h
** Last Mounted on /usr
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

If your system disks are RK07s, check the consistency of the `/usr/sys` file on drive 1 by typing

```
# fsck /dev/rhk1a
```

The output from `fsck` should look similar to:

```
** /dev/rhk1a
** Last Mounted on /usr/sys
** Phase 1 - Check Blocks and sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
864 files, 3745 used, 3678 free (134 frags, 433 blocks)
```

If there are inconsistencies in the filesystems, the system may prompt you for corrective action. See *fsck - the UNIX File System Check Program* for further information. If serious problems exist, try the extractions again, starting with the `newfs ${name} ${type}` command above.

Now remount the `/usr` filesystem by entering:

```
# /etc/mount /dev/${name} /usr
```

If your system disks are RK07s, you also need to remount the `/usr/sys` filesystem. To do this enter:

```
# /etc/mount /dev/hk1a /usr/sys
```

You have now extracted all of the data on the first distribution tape. Unload the first tape from the tape drive, and store it in a safe place. You will be extracting the files from the second distribution tape later in subsection 5.5.

The ULTRIX-32 system is bootstrapped and running in single-user mode. See section 3 (System Setup) for information on how to further set up your system.

2.3. Bootstrap Procedure for VAX-11/780 from Tape Distribution

Subsection 2.3 explains the bootstrap procedure for installing the ULTRIX-32 software on a VAX-11/780, from the distribution tape.

This procedure should take between one hour and one and one half hours to complete.

Getting Started

Before starting the bootstrap procedure, you must be familiar with your system configuration. You must know which hardware devices are connected to which controllers, naming conventions, unit numbers, and so forth. If your system is under a Digital Field Service Contract, then you can obtain detailed configuration information from the Configuration Worksheet section of the *Site Management Guide*.

You will need the following equipment:

- The Two ULTRIX-32 distribution tapes
- The ULTRIX-32 distribution floppy diskette
- One blank floppy diskette
- Formatted disk pack(s)
- VAX-11/780 system with a tape drive and disk(s)

The following steps are required to prepare the system for booting from tape:

- 1) If your disk packs are removable, load a formatted disk pack into disk drive 0. (All Digital Equipment Corporation disk packs are shipped pre-formatted.) Load any additional packs you may need as well, but the pack on drive 0 will be the root (or system) disk.

Even if your disk pack is nonremoveable, be sure your system disk pack is spinning and that the READY light is lit, even if your pack is nonremoveable.

- 2) Load the first distribution tape (labeled BOOT TAPE) on tape drive 0. Be sure the tape drive is ON-LINE and READY, and that the tape is rewound to load point (BOT).
- 3) Insert the distribution floppy diskette into the console floppy inside the processor (drive 0). (Note that the floppy diskette label points toward the handle of the floppy diskette drive.)
- 4) On the processor itself, set the auto restart switch to OFF and the rotary key switch to LOCAL.

34 ULTRIX-32 Installation Guide

General Steps

The tape bootstrap procedure to create a working system consists of the following general steps:

- 1) Copy a miniroot filesystem from the tape onto the swap area of the disk.
- 2) Boot the ULTRIX-32 system on the miniroot.
- 3) Restore the full root filesystem.
- 4) Update a console floppy diskette for bootstrapping.
- 5) Reboot the completed root filesystem.
- 6) Build and restore the /usr filesystem from tape.

The next few subsections describe these steps in detail. References to disk drives are of the form $xx(n,m)$ and references to files on tape drives are of the form $yy(n,m)$ where xx and yy are one of the names given in the table below.

For UNIBUS devices, n is usually 0 because it is equal to the unit number of the disk or tape drive. For MASSBUS devices, n is computed as follows:

$$n = 8 * (mba) + (\text{device unit number})$$

For example, if drive 0 were on the first MASSBUS adapter (mba0) then n would be 0; if drive 0 were on the second MASSBUS adapter (mba1) then n would be 8.

m is always specified in the command lines of this document. For a disk, m is the partition (in the range 0–7). For a tape, m is the file number on the tape.

Table 2.3.1 is a quick reference table for the disk and tape drive ULTRIX-32 system naming conventions:

Table 2.3.1 Disk and Tape Drive Naming Conventions

Disk	xx	Tape	yy
RA60	ra	TE16	ht
RA80	ra	TS11	ts
RA81	ra	TU77	ht
RK07	hk	TU78	mt
RL02	rl	TU80	ts
RM03	hp		
RM05	hp		
RM80	hp		
RP06	hp		
RP07	hp		

If you have a DECwriter III (LA120), you must set up your console terminal before proceeding with the installation.

To set up the console, hold down <SETUP> and press the D key simultaneously until the LED display on the console (at the upper left-hand corner of the keys) is 0 (zero).

2.3.1. Copy the Miniroot Filesystem

The first step is to run the program *copy*, which copies the miniroot filesystem from the second tape file of the first distribution tape into the second partition of the disk. This filesystem will serve as the base for creating the actual root filesystem.

The system must be halted; if it is not, type:

```
^P
>>>H
```

The console subsystem prompts with >>>. To load the *copy* program, type the following in response to the prompt. If you make a typing mistake, type <DELETE> to erase a character, <CTRL/U> to erase a line, or <CTRL/C> to interrupt the process.

```
>>>H                                (halt the processor)
    (a VAX-11/780 can be in the console subsystem, but not be halted)
    HALTED AT xxxxxxxx                (program counter)
>>>UNJAM                             ("unjam" the SBI)
>>>I                                  (initialize the processor)
    INIT SEQ DONE
>>>LOAD COPY                          (boot the floppy diskette)
                                        (this takes about two minutes)
    LOAD DONE, xxxxxxxx BYTES LOADED
>>>START 2                            (start the copy program)
From: yy(n,1)                         (from the distribution tape)
To: xx(n,1)                            (to the disk pack)
                                        (this takes about five minutes)
Copy completed: 300 records copied
From:^P                                (<CTRL/P> returns you to the console
                                        subsystem)
H                                       (halt the processor)
    HALTED AT xxxxxxxx                (program counter)
>>>>
```

You now have the set of tools necessary to create a root filesystem and restore the filesystem contents from tape.

36 ULTRIX-32 Installation Guide

2.3.2. Boot from the Miniroot Filesystem

The second step is to load the bootstrap program and boot the ULTRIX-32 miniroot system which you just copied to the disk. To do this, type:

```
>>> B ANY                                (boot the floppy diskette)
      CPU HALTED
      INIT SEQ DONE
      LOAD DONE, xxxxxxxx BYTES LOADED
```

```
Boot
: xx(n,1)vmunix                          (boot the ULTRIX-32 kernel off the
                                          miniroot; this takes a minute)
.
.
.
```

The standalone boot program then reads the ULTRIX-32 miniroot system (which you just created via the *copy* program) from the disk. The system then starts, displaying the following messages:

```
215564+64088+69764 start 0xf98          (these numbers are approximate)
ULTRIX-32 date and time                (this takes a minute)
real mem = xxxxxx
avail mem = yyy
```

```
.
.
.
information about available devices
```

```
.
.
.
root device?
```

2.3.2.1. Explanation of System Boot Messages

The first three numbers printed by the bootstrap programs are the sizes of different parts of the system (text, initialized and uninitialized data).

The ULTRIX-32 system then runs for the first time, beginning with a banner identifying the release and version of the system that is in use and the date it was compiled.

The *mem* messages give the amount of real (physical) memory and the memory available to user programs in bytes. For example, if your machine has 4mb of memory, then *xxxxxx* will be 4190208, 4096 bytes less than 4mb. The system reserves the last 4096 bytes of memory for use in logging system messages and does not count it as part of real memory. Available memory is the amount of physical memory left over for user processes after the ULTRIX-32 software has been loaded.

Information about the system is shown next, followed by messages showing what devices were found on the current processor. These messages are described in *autoconf*(4). The distributed system may not have found all the communications devices (dmf's and dz's), or mass storage peripherals you have if you have more than two of one device. This is corrected later when you create a description of your machine from which to configure the ULTRIX-32 system. These messages contain much of the information that will be used in creating the configuration. In a correctly configured system most of the information present in the configuration description is printed out at boot time as the system verifies that each device is present.

You may receive a warning message about the date. The system is telling you that it realizes the date and time currently stored in memory are wrong. Do not worry about this message; you will be changing the date and time later (subsection 2.3.6).

2.3.2.2. Boot from the Miniroot Filesystem, Continued

The root device? prompt asks for the name of the root filesystem to use:

```
root device? xx0*
erase ^?, kill ^U, intr ^C
#
```

The response *xx0** supplies two pieces of information. First, *xx0* indicates the disk it is running on is drive 0 of type *xx*. Second, the *** indicates the system is running atop the paging area. The *** is very important because it prevents the system from attempting to page on top of itself.

NOTE: If at any time the miniroot filesystem requires rebooting, be sure to copy the miniroot first (see subsection 2.3.1).

The shell executes */.profile* when it starts, which in turn generates the erase ... message. This message informs you that <DELETE> is the erase character, <CTRL/U> is the erase line character (kill), and <CTRL/C> is the interrupt character.

38 ULTRIX-32 Installation Guide

The ULTRIX-32 system is now running. The # is the prompt from the shell, and lets you know that you are the superuser, whose login name is root.

2.3.3. Restore the Root Filesystem

The third step is to create the root filesystem. To do this, execute the xtr shell script by typing:

```
# disk=xx0 type=dddd tape=yy xtr
```

where *xx0* is the name of the disk drive (unit 0), *dddd* is the type of disk drive on which the root filesystem is to be restored and, *yy* is the name of the tape drive on which the distribution tape is mounted (see table below). The Table 2.3.2 is a quick reference of the disk and tape drive ULTRIX-32 system naming conventions:

Table 2.3.2 ULTRIX-32 System Naming Conventions

Disk	xx	dddd	Tape	yy
RA60	ra	ra60	TE16	ht
RA80	ra	ra80	TS11	ts
RA81	ra	ra81	TU77	ht
RK07	hk	rk07	TU78	mt
RM03	hp	rm03	TU80	ts
RM05	hp	rm05		
RM80	hp	rm80		
RP06	hp	rp06		
RP07	hp	rp07		

For example, if you have an RK07 disk drive and a TU80 tape drive, type:

```
# disk=hk0 type=rk07 tape=ts xtr
```

The xtr script generates messages regarding the construction of the filesystem and the restoration of the tape contents. The messages will eventually terminate with the following:

```
. (this takes about five minutes)  
. .  
. .
```

Root filesystem extracted

```
If this is a 780, update floppy  
If this is a 730, update the cassette  
#
```

Check the consistency of the root filesystem by typing:

```
# fsck /dev/rxx0a
```

(*xx* is found in Table 2.3.2)

The output from `fsck` should look similar to:

```
** /dev/rxx0a
** Last Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

2.3.4. Update the Console Floppy Diskette

The fourth step is to build an ULTRIX-32 console floppy diskette. To do this, copy some files from the console floppy diskette, add some files from the ULTRIX-32 system and write them all to a blank diskette.

The console floppy diskette will allow the ULTRIX-32 system to reboot automatically whenever necessary.

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

Then change into the directory where the console floppy diskette information is stored by typing:

```
# cd /sys/floppy
```

Then set up the default boot device. If you intend to use an RK07 as your primary root type:

```
# cp defboo.hk defboo.cmd
```

If you intend to use a drive on a UDA50 (for example, an RA81) as your primary root type:

```
# cp defboo.ra defboo.cmd
```

Otherwise type:

```
# cp defboo.hp defboo.cmd
```

NOTE: If the local configuration requires any changes in `restar.cmd` or `defboo.cmd` (for example, for interleaved memory controllers), these should be made now.

Type the following command to build the updated boot floppy diskette:

40 ULTRIX-32 Installation Guide

```
# make update
```

```
.  
. .  
.
```

lists all the files on the hardware floppy diskette

```
.  
. .  
.
```

Change Floppy, Hit return when done.

Remove the VAX-11/780 console floppy diskette, insert a blank (scratch) diskette, and press the <RETURN> key. To complete the procedure, respond with yes to the system's next question.

Are you sure you want to clobber the floppy? yes

```
.  
. .  
.
```

lists all the files on the new boot floppy diskette

```
.  
. .  
.
```

```
#
```

You can make more copies of the boot floppy diskette by using the `fcopy` maintenance command (see `arff(8V)`).

2.3.5. Reboot the Completed Root Filesystem

The fifth step is to reboot the system. To do this, type the following commands. If you make a mistake typing, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

```
# sync (flush incore buffers to disk)
# sync
# ^P (CTRL/P returns you to the console
      subsystem)
>>>H (halt the processor)
      HALTED AT xxxxxxxx
>>>U ("unjam" the SBI)
>>>I (initialize processor state)
      INIT SEQ DONE
>>>B ANY
      (boot program is eventually loaded)
```

```
.
.
.
boot program messages appear
```

```
.
.
.
Boot
: xx(n,0)vmunix (boot the ULTRIX-32 kernel off the
                  root; this take about one minute)
```

```
.
.
.
ULTRIX-32 date and time
real mem = xxx
avail mem = yyy
```

```
.
.
.
information about available devices
```

```
.
.
.
root device? xx0
erase ^?, kill ^U, intr ^C
```

42 ULTRIX-32 Installation Guide

#

If the system does not reboot properly, see Appendix C for help with troubleshooting.

The system is running in single-user mode on the installed root filesystem. The # prompt is from the ULTRIX-32 operating system, and tells you that you are the superuser.

The next subsection tells how to complete the installation of distributed software for the /usr filesystem.

2.3.6. Set up and Install the /usr Filesystem

The last step is to build and restore the /usr filesystem.

First, set a shell variable to the name of your disk drive so the commands given will work regardless of the drive you have. To do this, find your disk drive in the following list, and type the command to the right:

<u>Disk Drive</u>	<u>Command</u>
RA60	disk=ra
RA80	disk=ra
RA81	disk=ra
RK07	disk=hk
RM03	disk=hp
RM05	disk=hp
RM80	disk=hp
RP06	disk=hp
RP07	disk=hp

For example, if you have an RK07, type:

```
# disk=hk
```

The next two commands select a partition for /usr. If you have more drives available, see subsection 5.1 (Fine Tuning the Disk Configuration) to help decide if some other partition might be more appropriate.

The partitions used below are those most appropriate in size. Find your disk in the following list, and type the commands to the right:

Disk	Commands
RA60	name=ra0h; type=ra60
RA80	name=ra0h; type=ra80
RA81	name=ra0h; type=ra81
RK07	name=hk0g; type=rk07
RM03	name=hp0g; type=rm03
RM05	name=hp0g; type=rm05
RM80	name=hp0g; type=rm80
RP06	name=hp0g; type=rp06
RP07	name=hp0h; type=rp07

For example, if you have an RK07, type:

```
# name=hk0g; type=rk07
```

Next, find your tape drive in the following list, and type the commands to the right:

Tape Drive	Commands
TE16	cd /dev; MAKEDEV ht0; sync
TS11	cd /dev; MAKEDEV ts0; sync
TU77	cd /dev; MAKEDEV ht0; sync
TU78	cd /dev; MAKEDEV mt0; sync
TU80	cd /dev; MAKEDEV ts0; sync

For example, if you have a TU80, type:

```
# cd /dev; MAKEDEVts0; sync
```

Now set the date and time:

```
# date yymmddhhmm (set date)
```

Where:

yy equals the last two digits of the year
mm equals the number of the month of the year
dd equals the number of the day of the month
hh equals the hour of the day (based on 24 hours)
mm equals the minutes of the hour

(the system echos the new date and time)

Next, set the superuser password. The password you select must contain at least six


```
# cd /
# chmod 755 /usr/sys
```

(make / (root) the current directory)

If your system disks are RK07s, type:

```
# umount /dev/hk1a
```

Continue with the following for all system disks, including RK07s:

```
# umount /dev/${name}
```

Check the consistency of the /usr filesystem by typing:

```
# fsck /dev/r${name}
```

The output from fsck should look similar to:

```
** /dev/rxx0h
** Last Mounted on /usr
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

If your system disks are RK07s, check the consistency of the /usr/sys file on drive 1 by typing:

```
# fsck /dev/rhk1a
```

The output from fsck should look similar to:

```
** /dev/rhk1a
** Last Mounted on /usr/sys
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
864 files, 3745 used, 3678 free (134 frags, 433 blocks)
```

If there are inconsistencies in the filesystems, the system may prompt you for corrective action. See *fsck - the UNIX Filesystem Check Program* for further information. If serious problems exist, try the extractions again, starting with the newfs \${name} \${type} command above.

46 ULTRIX-32 Installation Guide

Now remount the /usr filesystem by typing:

```
# /etc/mount /dev/$name /usr
```

If your system disks are RK07s, you also need to remount the /usr/sys filesystem. To do this, type:

```
# /etc/mount /dev/hk1a /usr/sys
```

You have now extracted all the data on the first distribution tape. Unload the first tape from the tape drive, and store it in a safe place. You will be extracting the files from the second distribution tape later in subsection 4.5.

The ULTRIX-32 operating system is installed and running in single-user mode. See section 3 (System Setup) for information on how to further set up your system.

2.4. Bootstrap Procedure for All VAXes from RA60 Distribution

Subsection 2.4 explains the bootstrap procedure for installing the ULTRIX-32 software on the VAX-11/730, VAX-11/750, and VAX-11/780 processors from the distribution RA60 disk.

This procedure should take about half an hour to complete.

Getting Started

Before starting the bootstrap procedure, you must be familiar with your system configuration. If your system is under a Digital Service Contract, then you can obtain detailed configuration information from the Configuration Worksheet section of the *Site Management Guide*.

If your processor is a VAX-11/750 you need a UDA Boot ROM. However, this document explains how to bootstrap a system without the ROM.

You will need the following equipment:

- One ULTRIX-32 distribution RA60 disk
- One ULTRIX-32 distribution cassette (for VAX-11/730, and VAX-11/750 systems without the UDA Boot ROM)
- One ULTRIX-32 distribution floppy diskette (for VAX-11/780 systems, only)
- One blank floppy diskette (for VAX-11/780 systems)
- One blank cassette (for VAX-11/730 systems)
- One formatted RA60 disk pack (to make the backup copy of the distribution)
- A VAX system

Follow these steps to prepare your system for booting from the RA60:

1) Load the Distribution RA60 Disk Pack

Load the distribution disk pack into disk drive 0. Be sure the RA60 disk pack is spinning and that the READY light is lit.

2) Select the destination disk drive

If the destination disk drive you select is not drive UNIT 1, change the unit plug so it becomes UNIT 1.

3) Insert the Distribution Cassette or Floppy Diskette

If you have either a VAX-11/750 without the UDA ROM or a VAX-11/730 processor, insert the distribution cassette into the console TU58 disk drive on the front of the processor (drive 0). (Note that the cassette label faces up.)

If you have a VAX-11/780, insert the distribution floppy diskette into the console

48 ULTRIX-32 Installation Guide

floppy diskette drive inside the processor (drive 0). (Note that the floppy diskette label points toward the handle of the diskette drive.)

4) Prepare the CPU

If your CPU is a VAX-11/730, set the re-start switch to OFF and the rotary key switch to LOCAL.

If your CPU is a VAX-11/750, 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 the installation.

If your CPU is a VAX-11/780, set the auto re-start switch to OFF and the rotary key switch to LOCAL. It makes no difference how the boot device switch is set during the installation.

5) Prepare the DECwriter III (LA120)

If you have a DECwriter III (LA120), you must set up your console terminal before proceeding with the installation.

To set up the console, hold down <SETUP> and press D simultaneously until the LED display on the console (at the upper left-hand corner of the keys) is 0 (zero).

2.4.1. Boot the Root Filesystem

The system must be halted; if it is not, type:

```
^P
>>>H
```

The console subsystem prompts with >>>. If you make a typing mistake, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

The first step is to load the bootstrap program and boot the ULTRIX-32 root system. To do this, type the following command sequence which pertains to your system:

a) VAX-11/730

```
>>>I                (initialize the processor)
>>>L DD0:BOOT       (boot the cassette)
                    (this takes about one minute)
>>>S 2              (start the boot program)
Boot

:ra(0,0)vmunix      (boot the ULTRIX-32 kernel off the root)
                    (this takes about one minute)
```

.
.
.

b) VAX-11/750

VAX-11/750 With a UDA Boot ROM:

```
>>>I (initialize the processor)
>>>B/3 DUA0
%%%
```

VAX-11/750 Without a UDA Boot ROM:

```
>>>I (initialize the processor)
>>>B DDA0 (boot the cassette)
          (this takes about 60 seconds)
%%%
= boot (start the boot program)
        (this takes about one minute)
```

Both VAX-11/750s:

Type the following for VAX-11/750 systems with and without the UDA Boot ROM:

```
Boot
: ra(0,0)vmunix (boot the ULTRIX-32 kernel off the root)
.
.
.
```

c) VAX-11/780

```
>>>U (unjam the SBI)
>>>I (initialize the processor)
>>>B ANY (boot the floppy diskette)

CPU HALTED
INIT SEQ DONE
LOAD DONE, xxxxxxxx BYTES LOADED
```

50 ULTRIX-32 Installation Guide

```
Boot
: ra(0,0)vmunix                (boot the ULTRIX-32 kernel off the root)
                                (this takes about one minute)
```

.
.
.

All VAXes

The standalone boot program (for all VAXes) then reads the ULTRIX-32 kernel from the root filesystem. The system starts, displaying the following messages:

```
.
.
.

Updating 11/750 micro code...      (appears if you are running
                                   revision 5 or greater on
                                   VAX-11/750s only)

215564+64088+69764 start 0xf98     (these numbers are approximate)
ULTRIX-32 (date and time)         (this takes a minute)
real mem = xxxxxx
avail mem = yyy

.
.
.

information about available devices

.
.
.

root device? ra0
erase ^?, kill ^U, intr ^C
#
```

2.4.1.1. Explanation of System Boot Messages

The first three numbers printed by the bootstrap programs are the sizes of different parts of the system (text, initialized and uninitialized data).

The ULTRIX-32 system then runs for the first time, beginning with a banner identifying the release and version of the system that is in use and the date it was compiled.

The *mem* messages give the amount of real (physical) memory and the memory available to user programs in bytes. For example, if your machine has 4mb of memory, then *xxxxxx* will be 4190208, 4096 bytes less than 4mb. The system reserves the last 4096 bytes of memory for use in logging system messages and does not count it as part of real memory. Available memory is the amount of physical memory left over for user processes after the ULTRIX-32 software has been loaded.

Information about the system is shown next, followed by messages showing what devices were found on the current processor. These messages are described in `autoconf(4)`. The distributed system may not have found all the communications devices (*dmfs* and *dzs*), or mass storage peripherals you have if you have more than two of one device. This is corrected later when you create a description of your machine from which to configure the ULTRIX-32 system. These messages contain much of the information that will be used in creating the configuration. In a correctly configured system most of the information present in the configuration description is printed out at boot time as the system verifies that each device is present.

If the system does not reboot properly, see Appendix C for help with troubleshooting.

The ULTRIX-32 system is running in single-user mode. The `#` prompt is from the ULTRIX-32 operating system, and tells you that you are the superuser.

2.4.2. Set the Date and the Root Password

First, set the date and time:

```
# date yymddhhmm
```

Where:

```
yy equals the last two digits of the year
mm equals the number of the month of the year
dd equals the number of the day of the month
hh equals the hour of the day (based on 24 hours)
mm equals the minutes of the hour
```

Next, set the superuser password. The password you select must contain at least six

52 ULTRIX-32 Installation Guide

characters:

```
# passwd root                (set password for superuser)
New password:                (password will not echo)
Retype new password:
#
```

2.4.3. Backup the Distribution Medium

In this step you will copy your RA60 distribution onto another disk (RA60, RA80, or RA81). The newly copied disk will become your system disk, and the original distribution pack will be stored in a safe place for backup purposes.

To make the copy, follow the directions below.

If the destination drive you chose is an RA60, load it into drive 1 (see **Getting Started**).

Be sure the destination disk is spinning and the READY light is lit.

Find your destination drive from the list below and use the corresponding value for *zz* in the *newfs* command:

<i>zz</i>	Destination Drive
60	RA60
80	RA80
81	RA81

Issue the *newfs* command:

```
# newfs ra1a razz
```

For example, if your destination drive were an RA81 you would type:

```
# newfs ra1a ra81
```

Now copy the RA60 distribution using the *dump(8)* and *restore(8)* commands to copy the */* and */usr* partitions as follows.

First copy the */* partition from the RA60 distribution onto the corresponding partition of your destination pack on drive 1:

```
# sync
# /etc/mount /dev/ra1a /mnt          (specify the destination partition)
# dump Of - /dev/ra0a | (cd /mnt; restore rf -)
                                     (ra0a is the source partition)

# sync
# /etc/umount /dev/ra1a              (unmount the destination partition)
# fsck /dev/rra1a                    (fsck the destination raw device)
# sync
```

Now copy the /usr partition. First issue the newfs command:

```
# newfs ra1h razz                (zz has the same value as above)
```

Now repeat the commands above altering the dump and restore commands to reflect the partition:

```
# sync
# /etc/mount /dev/ra1h /mnt      (specify the destination partition)
# dump Of - /dev/ra0g | (cd /mnt; restore rf -)
                                (ra0h is the source partition)

# sync
# /etc/umount /dev/ra1h         (unmount the destination partition)
# fsck /dev/rra1h              (fsck the destination raw device)
# sync
# halt
```

Spin down and unload the original distribution pack in drive 1. Store the distribution pack in a safe place for backup purposes.

Make sure the unit number of the disk (drive) you just copied is 0. If it is not UNIT 0, swap the unit plug with that of drive UNIT 0.

Now reboot the system to single-user mode on the new disk (drive 0) using the steps in subsection 2.4.1.

Your system should be running off the newly copied pack.

2.4.4. Update the Console Cassette or Floppy Diskette

If you have a VAX-11/730 or a VAX-11/780 you need to update the console cassette or console floppy diskette.

To prepare the system type:

```
# /etc/mount /dev/ra0h /usr
```

2.4.4.1. Update the Console Cassette

If you have a VAX-11/730 system, you need to build the boot cassette. You do this by copying some files from the local console cassette (that is, the cassette that came with your hardware), adding some files from the ULTRIX-32 system, and writing them all to a blank cassette.

This new boot cassette will allow the ULTRIX-32 system to reboot automatically whenever necessary.

First, remove the distribution cassette from the external TU58 drive (drive 0) and insert a blank cassette. Be sure the blank cassette is NOT write protected (the record tab should be in the left-most position).

54 ULTRIX-32 Installation Guide

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

Then, change into the directory where the console cassette information is stored:

```
# cd /sys/cassette
```

Next set up the default boot device. If you are using an IDC R80 as your primary root type:

```
# cp defboo.rb defboo.cmd
```

If you are using a drive on a UDA50 as your primary root type:

```
# cp defboo.ra defboo.cmd
```

To build the updated boot cassette, type:

```
# make update
```

```
.  
. .  
.
```

(this takes about 25 minutes)

lists all of the files on the cassette

```
.  
. .  
.
```

Console cassette update complete.

The new cassette should be installed in the internal tu58 slot (Drive 1).

```
#  
#
```

Store the distribution cassette and the local console cassette in a safe place. The local console cassette is required to boot VAX/VMS systems. Remove the boot cassette you just built from the external drive and insert it in the internal TU58 cassette drive (drive 1).

At this point, you can make more copies of the new boot cassette by using the `fcopy` maintenance command (see `arff(8V)`).

The system is now bootstrapped, and running on your system disk in single-user mode. Be sure to store the original RL02 distribution disks in a safe place.

See section 3 (System Setup) for information on how to further set up your system.

2.4.4.2. Update the Console Floppy Diskette

If you have a VAX-11/780 you will need to build the boot floppy diskette. You do this by copying some files from the console floppy diskette, adding some files from the ULTRIX-32 system, and writing them all to a blank diskette.

The boot floppy diskette will allow the ULTRIX-32 system to reboot automatically whenever necessary.

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

Then change into the directory where the console floppy diskette information is stored by typing:

```
# cd /sys/floppy
```

Then set up the default boot device. If you are using a drive on a UDA50 (for example, an RA81) as your primary root type:

```
# cp defboo.ra defboo.cmd
```

Otherwise type:

```
# cp defboo.hp defboo.cmd
```

NOTE: If the local configuration requires any changes in `restar.cmd` or `defboo.cmd` (for example, for interleaved memory controllers), these should be made now.

Type the following command to build the updated boot floppy diskette:

```
# make update
```

```
.  
.  
.
```

lists all the files on the hardware floppy diskette

```
.  
.  
.
```

Change Floppy, Hit return when done.

Remove the VAX-11/780 console floppy diskette, insert a blank (scratch) diskette, and press the <RETURN> key. To complete the procedure, respond with yes to the system's next question.

56 ULTRIX-32 Installation Guide

Are you sure you want to clobber the floppy? yes

.
.
.

lists all the files on the new boot floppy diskette

.
.
.

#

You can make more copies of the boot floppy diskette by using the floppy maintenance command (see `arff(8V)`).

Your ULTRIX-32 system is now fully bootstrapped. See section 3 (System Setup) for information on how to further set up your system.

5. Bootstrap Procedure for All VAXes from RK07 Distribution

Section 2.5 explains the bootstrap procedure for installing the ULTRIX-32 software on the VAX-11/730, VAX-11/750, and VAX-11/780 processors from the distribution RK07 disks.

This procedure should take about five hours to complete.

Getting Started

Before starting the bootstrap procedure, you must be familiar with your system configuration. If your system is under a Digital Service Contract, then you can obtain detailed configuration information from the Configuration Worksheet section of the *Site Management Guide*.

If your processor is a VAX-11/750 you need an RK Boot ROM.

You will need the following equipment:

- Three ULTRIX-32 distribution RK07 disks
- One ULTRIX-32 distribution cassette (for VAX-11/730 and VAX-11/750 systems)
- One ULTRIX-32 distribution floppy diskette (for VAX-11/780 systems, only)
- One blank cassette (for VAX-11/730 systems)
- One blank floppy diskette (for VAX-11/780 systems)
- Three formatted RK07 disk packs (to make a backup copy of the distribution)
- A VAX system

Follow these steps to prepare your system for booting from the RK07s:

Load the First Distribution & Scratch RK07 Disk Packs

Load the distribution disk pack labeled ROOT in disk drive 0, and load a scratch disk pack in disk drive 1. Be sure the disk packs are spinning and that the READY lights are lit.

Insert the Distribution Cassette or Floppy Diskette

If you have a VAX-11/730 or VAX-11/750 processor, insert the distribution cassette into the console TU58 disk drive on the front of the processor (drive 0). (Note that the cassette label faces up.) Be sure the cassette is write protected (the record tab is in the right-most position).

If you have a VAX-11/780, insert the distribution floppy diskette into the console floppy diskette drive inside the processor (drive 0). (Note that the floppy diskette label points toward the handle of the diskette drive.)

58 ULTRIX-32 Installation Guide

3) Prepare the CPU

If your CPU is a VAX-11/730, set the restart switch to OFF and the rotary key switch to LOCAL.

If your CPU is a VAX-11/750, 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 the installation.

If your CPU is a VAX-11/780, set the auto re-start switch to OFF and the rotary key switch to LOCAL. It makes no difference how the boot device switch is set during the installation.

4) Prepare the DECwriter III (LA120)

If you have a DECwriter III (LA120), you must set up your console terminal before proceeding with the installation.

To set up the console, hold down <SETUP> and press D simultaneously until the LED display on the console (at the upper left-hand corner of the keys) is 0 (zero).

2.5.1. Boot the Root Filesystem

The system must be halted; if it is not, type:

```
^P
>>>H
```

The console subsystem prompts with >>>. If you make a typing mistake, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

The first step is to load the bootstrap program and boot the ULTRIX-32 root system. To do this, type the following command sequence that pertains to your system:

a) VAX-11/730

```
>>>I                               (initialize the processor)
>>>L DD0:BOOT                       (boot the cassette)
                                     (this takes about one minute)
>>>S 2                               (start the boot program)

Boot

: hk(0,0)vmunix                     (boot the ULTRIX-32 kernel off
                                     the root)
.
.
.
```

b) VAX-11/750

```
>>>I
>>>B/3 DMA0
%%%
```

(initialize the processor)

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

(boot the ULTRIX-32 kernel off
the root)

```
.
.
.
```

c) VAX-11/780

```
>>>U
>>>I
>>>B ANY
```

(unjam the SBI)
(initialize the processor)
(boot the floppy diskette)

```
CPU HALTED
INIT SEQ DONE
LOAD DONE, xxxxxxxx BYTES LOADED
```

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

(boot the ULTRIX-32 kernel off
the root)

```
.
.
.
```

All VAXes

The standalone boot program (for all VAXes) then reads the ULTRIX-32 kernel from

60 ULTRIX-32 Installation Guide

the root filesystem. The system starts, displaying the following messages:

```
.
.
.

Updating 11/750 micro code...                (appears if you are running on
                                              a revision 5 or greater
                                              VAX-11/750 system)

215564+64088+69764 start 0xf98              (these numbers are approximate)
ULTRIX-32 (date and time)                  (this takes about five minutes)
real mem = xxxxxx
avail mem = yyy

.
.
.

information about available devices

.
.
.

root device? hk0                            (an RK07 on drive 0)
erase ^?, kill ^U, intr ^C
#
```

2.5.1.1. Explanation of System Boot Messages

The first three numbers printed by the bootstrap programs are the sizes of different parts of the system (text, initialized and uninitialized data).

The ULTRIX-32 system then runs for the first time, beginning with a banner identifying the release and version of the system that is in use and the date it was compiled.

The *mem* messages give the amount of real (physical) memory and the memory available to user programs in bytes. For example, if your machine has 4mb of memory, then *xxxxxx* will be 4190208, 4096 bytes less than 4mb. The system reserves the last 4096 bytes of memory for use in logging system messages and does not count it as part of real memory. Available memory is the amount of physical memory left over for user processes after the ULTRIX-32 software has been loaded.

Information about the system is shown next, followed by messages showing what devices were found on the current processor. These messages are described in `autoconf(4)`. The distributed system may not have found all the communications devices (dmfs and dzs), or mass storage peripherals you have if you have more than two of one

device. This is corrected later when you create a description of your machine from which to configure the ULTRIX-32 system. These messages contain much of the information that will be used in creating the configuration. In a correctly configured system most of the information present in the configuration description is printed out at boot time as the system verifies that each device is present.

If the system does not reboot properly, see Appendix C for help with troubleshooting.

The ULTRIX-32 system is running in single-user mode. The # prompt is from the ULTRIX-32 operating system, and tells you that you are the superuser.

2.5.2. Set the Date and the Root Password

First, set the date and time:

```
# date yymmddhhmm
```

Where:

```
yy equals the last two digits of the year
mm equals the number of the month of the year
dd equals the number of the day of the month
hh equals the hour of the day (based on 24 hours)
mm equals the minutes of the hour
```

Next, set the superuser password. The password you select must contain at least six characters:

```
# passwd root                (set password for superuser)
New password:                (password will not echo)
Retype new password:
#
```

2.5.3. Backup the Distribution Media

You should make a backup copy of your RK07 distribution. To do this, you will need three scratch RK07 disk packs.

You will use the `dump(8)` and `restore(8)` commands to make the backup copy of the distribution disk packs.

Follow the instructions in the next three subsections to make a backup copy of the distribution disk packs.

2.5.3.1. The First Distribution Pack (ROOT)

Be sure the ROOT distribution pack is in drive 0, the first scratch pack is in drive 1, and both disks are spinning and the READY lights are on.

62 ULTRIX-32 Installation Guide

Type the following to copy root and /usr (the a and g partitions):

```
# update
# newfs rhk1a rk07                (specify the destination partition. Save
                                   the numbers newfs generates)
Warning: inode blocks/cyl group (12) >= data blocks (2) in last
.
.
.
10656, 11744, 12704, 13792, 14880,
# sync
# /etc/mount /dev/hk1a /mnt        (specify the destination partition)
# dump 0f - /dev/hk0a | (cd /mnt; restore rf -)
                                   (hk0a is the source partition)
                                   (the entire dump takes about 5 minutes)
DUMP: Date of this level 0 dump: xxx xxx xx xx:xx:xx xxxx
.
.
.
Warning: ./lost+found: File exists
[NIL] ./tmp/rstmdir450741748: inconsistent state
abort? [yn] n                      (type n)
[NIL] ./tmp/rstmode450741748: inconsistent state
abort? [yn] n                      (type n)
DUMP: 50.78% done, finished in 0:04
DUMP: DUMP: 3958 tape blocks on 1 tape(s)
DUMP: DUMP IS DONE

# sync
# /etc/umount /dev/hk1a            (unmount the destination partition)
# fsck /dev/rhk1a                  (fsck the destination raw device)
** /dev/rhk1a
** last Mounted on /mnt
.
.
.
350 files, 3631 used, 3792 free (56 frags, 467 blocks)
# sync
# newfs rhk1g rk07                (specify the destination partition. Save
                                   the numbers newfs generates)
/dev/rhk1g: 27786 sectors in 421 cylinders of 3 tracks, 22 sectors
.
.
.
21152, 22232, 23312, 24392, 25376, 26456, 27536,
```

```
# sync
# /etc/mount /dev/hk1g /mnt          (specify the destination partition)
# dump 0f - /dev/hk0g | (cd /mnt; restore rf -)
                                   (hk0g is the source partition. The
                                   entire dump takes about 15 minutes)
```

DUMP: Date of this level 0 dump: xxx xxx xx xx:xx:xx xxxx

.
.
.

Warning: ./lost+found: File exits

DUMP: 53.37% done, finished in 0:04

DUMP: 99.68% done, finished on 0:00

DUMP: DUMP: 10286 tape blocks on 1 tape(s)

DUMP: DUMP IS DONE

```
# sync
# /etc/umount /dev/hk1g             (unmount the destination partition)
# fsck /dev/rhk1g                   (fsck the destination raw device)
** /dev/rhk1g
```

.
.

803 files, 18878 used, 7363 free (107 frags, 907 blocks)

```
# sync
```

```
# halt
```

```
syncing disks... done
```

```
halting (in tight loop): hit
```

```
^P
```

(type <CTRL/P>)

```
HALT ^P
```

(type H)

```
xx xxxxx
```

(program counter)

```
>>> H
```

NOTE: Do not be concerned with the warnings cited above. Do remember to save the numbers that newfs generates; they may prove helpful in an emergency.

Now spin down and unload the distribution disk pack in drive 0 and the scratch disk pack in drive 1. Label the scratch pack ROOT.

2.5.3.2. The Second Distribution Pack (USERS)

Load the distribution pack labeled USERS into drive 0. Load the second scratch pack into drive 1.

When both drives are spinning and the READY lights are lit, reboot the system off of drive 0 using the directions described in subsection 2.5.1.

64 ULTRIX-32 Installation Guide

After the system has been rebooted, type the following to copy root (the a partition):

```
# update
# newfs rhk1a rk07                (specify the destination partition. Save
                                   the numbers newfs generates)

Warning: inode blocks/cyl group (12) >= data blocks (2) in last
.
.
.
10656, 11744, 12704, 13792, 14880,
# sync
# /etc/mount /dev/hk1a /mnt        (specify the destination partition)
# dump 0f - /dev/hk0a | (cd /mnt; restore rf -)
                                   (hk0a is the source partition. The
                                   entire dump takes about 15 minutes)

DUMP: Date of this level 0 dump: xxx xxx xx xx:xx:xx xxxxx
.
.
.
Warning: ./lost+found: File exists
[NIL] ./tmp/rstmdir450741748: inconsistent state
abort? [yn] n                      (type n)
[NIL] ./tmp/rstmode450741748: inconsistent state
abort? [yn] n                      (type n)
DUMP: 50.78% done, finished in 0:04
DUMP: DUMP: 3958 tape blocks on 1 tape(s)
DUMP: DUMP IS DONE

# sync
# /etc/umount /dev/hk1a            (unmount the destination partition)
# fsck /dev/rhk1a                 (fsck the destination raw device)
** /dev/rhk1a
** last Mounted on /mnt
.
.
.
966 files, 5529 used, 1894 free (102 frags, 224 blocks)
# sync
# halt
# halt
syncing disks... done
halting (in tight loop); hit
    ^P                               (type <CTRL/P>)
    HALT ^P                          (type H)
xx xxxxx                             (program counter)
```

>>> H

NOTE: Do not be concerned with the warnings sited above. Do remember to save the numbers that newfs generates; they may prove invaluable in an emergency.

Now spin down and unload the distribution disk pack in drive 0 and the scratch disk pack in drive 1. Label the scratch pack USERS.

2.5.3.3. The Third Distribution Pack (ADDITIONAL SOFTWARE)

Load the distribution pack labeled ADDITIONAL SOFTWARE into drive 0, and load the third scratch pack into drive 1.

When both disks are spinning and the READY lights are on, reboot the system off of drive 0 using the directions described in subsection 2.5.1.

After the system has been rebooted, type the following to copy root and /usr (the a and g partitions):

```
# update
# newfs rhk1a rk07                (specify the destination partition. Save
                                   the numbers newfs generates)
Warning: inode blocks/cyl group (12) >= data blocks (2) in last
.
.
.
10656, 11744, 12704, 13792, 14880,
# sync
# /etc/mount /dev/hk1a /mnt        (specify the destination partition)
# dump 0f - /dev/hk0a | (cd /mnt; restore rf -)
                                   (hk0a is the source partition. The
                                   entire dump takes about 5 minutes)
DUMP: Date of this level 0 dump: xxx xxx xx xx:xx:xx xxxx
.
.
.
Warning: ./lost+found: File exists
[NIL] ./tmp/rstmdir450741748: inconsistent state
abort? [yn] n                      (type n)
[NIL] ./tmp/rstmode450741748: inconsistent state
abort? [yn] n                      (type n)
DUMP: 50.78% done, finished in 0:04
DUMP: DUMP: 2028 tape blocks on 1 tape(s)
DUMP: DUMP IS DONE

# sync
# /etc/umount /dev/hk1a            (unmount the destination partition)
# fsck /dev/rhk1a                 (fsck the destination raw device)
```

66 ULTRIX-32 Installation Guide

```
** /dev/rhk1a
** last Mounted on /mnt
.
.
.
350 files, 3631 used, 3792 free (56 frags, 467 blocks)
# sync
# dd if=/dev/rhk0g of=/dev/rhk1g bs=10b
# halt
syncing disks... done
halting (in tight loop): hit
    ^P                               (type <CTRL/P>)
    HALT ^P                          (type H)
xx xxxxx                            (program counter)
>>> H
```

NOTE: Do not be concerned with the warnings and diagnostics sited above. Do remember to save the numbers that newfs generates; they may prove invaluable in an emergency.

Now spin down and unload the distribution disk pack in drive 0 and the scratch disk pack in drive 1. Label the scratch pack ADDITIONAL SOFTWARE.

Load the backup copy labeled ROOT into drive 0, and load the backup copy labeled USERS into drive 1.

If you have a third RK07 drive, load the backup copy labeled ADDITIONAL SOFTWARE into drive 2.

When all the drives are spinning and the READY lights are on, reboot the system off of drive 0 using the directions described in subsection 2.5.1.

The backup is now complete. From now on you should run your system off of the backup copies. Store the distribution disk packs in a safe place.

If your system is a VAX-11/750, proceed to section 3 (System Setup) for information on how to further set up your system. Otherwise continue with the subsection below.

2.5.4. Update the Console Cassette or Floppy Diskette

If you have a VAX-11/730 or a VAX-11/780 you need to update the console cassette or console floppy diskette.

Be sure the backup pack labeled ROOT is in drive 0, and the backup pack labeled USERS is in drive 1. Both packs should be spinning, with the READY lights lit.

Type the following:

```
# /etc/mount /dev/hk0g /usr
# /etc/mount /dev/hk1a/usr/sys
# rm /sys
# ln -s /usr/sys/sys /sys
```

You are now ready to build the boot cassette or floppy diskette.

2.5.4.1. Update the Console Cassette

If you have a VAX-11/730 system, you need to build the boot cassette. You do this by copying some files from the local console cassette (that is, the cassette that came with your hardware), adding some files from the ULTRIX-32 system, and writing them all to a blank cassette.

This new boot cassette will allow the ULTRIX-32 system to reboot automatically whenever necessary.

First, remove the distribution cassette from the external TU58 drive (drive 0) and insert a blank cassette. Be sure the blank cassette is NOT write protected (the record tab should be in the left-most position).

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

Then, change into the directory where the console cassette information is stored and set up the default boot device:

```
# cd /sys/cassette
# cp defboo.hk defboo.cmd
```

To build the updated boot cassette, type:

```
# make update
```

```
.
.
.
```

(this takes about 25 minutes)

lists all of the files on the cassette

```
.
.
.
```

Console cassette update complete.

68 ULTRIX-32 Installation Guide

The new cassette should be installed in the internal tu58 slot (Drive 1).

```
# sync
# newfs rhk1g rk07
# sync
#
```

Store the distribution cassette and the local console cassette in a safe place. The original local console cassette is required to boot VAX/VMS systems. Remove the boot cassette you just built from the external drive and insert it in the internal TU58 cassette drive (drive 1).

At this point, you can make more copies of the new boot cassette by using the `fcopy` maintenance command (see `arff(8V)`).

The system is now bootstrapped, and running on your system disk in single-user mode. Be sure to store the original RL02 distribution disks in a safe place.

See section 3 (System Setup) for information on how to further set up your system.

2.5.4.2. Update the Console Floppy Diskette

If you have a VAX-11/780 you need to build the boot floppy diskette. You do this by copying some files from the console floppy diskette, adding some files from the ULTRIX-32 system, and writing them all to a blank diskette.

The boot floppy diskette will allow the ULTRIX-32 system to reboot automatically whenever necessary.

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

Then change into the directory where the console floppy diskette information is stored and set up the default boot device:

```
# cd /sys/floppy
# cp defboo.hk defboo.cmd
```

NOTE: If the local configuration requires any changes in `restar.cmd` or `defboo.cmd` (for example, for interleaved memory controllers), these should be made now.

Type the following command to build the updated boot floppy diskette:

```
# make update
```

```
.  
.
.
```

lists all the files on the hardware floppy diskette

```
.  
.
.
```

Change Floppy, Hit return when done.

Remove the VAX-11/780 console floppy diskette, insert a blank (scratch) diskette, and press the <RETURN> key. To complete the procedure, respond with yes to the system's next question.

Are you sure you want to clobber the floppy? yes

```
.  
.
.
```

lists all the files on the new boot floppy diskette

```
.  
.
.
```

```
# sync
```

```
# newfs rhk1g rk07
```

```
# sync
```

You can make more copies of the boot floppy diskette by using the `fcopy` maintenance command (see `arff(8V)`).

Your system is now bootstrapped and running in single-user mode. You will be booting the ADDITIONAL SOFTWARE backup disk later in subsection 5.5.

See section 3 (System Setup) for information on how to further set up your system.

2.6. Bootstrap Procedure for VAX-11/730 & VAX-11/750 from RL02

Subsection 2.6 explains the bootstrap procedure for installing the ULTRIX-32 software on the VAX-11/730 and VAX-11/750 processors from the distribution RL02 disk.

This procedure should take about one and one half hours to complete.

Getting Started

Before starting the bootstrap procedure, you must be familiar with your system configuration. If your system is under a Digital Service Contract, then you can obtain detailed configuration information from the Configuration Worksheet section of the *Site Management Guide*.

If your processor is a VAX-11/750 you should have an RL Boot ROM. However, this document explains how to bootstrap a system without the ROM.

You will need the following equipment:

- Four ULTRIX-32 distribution RL02 disks
- One ULTRIX-32 distribution cassette (for VAX-11/730's, and VAX-11/750 systems without the RL ROM)
- One blank cassette (for VAX-11/730 systems)
- A VAX-11/730 or VAX-11/750 system

Follow these steps to prepare your system for booting from the RL02:

1) Load the RL02 Miniroot Disk Pack

On a VAX-11/730, load the distribution disk pack labeled ROOT into disk drive 1.

On a VAX-11/750, load the distribution disk pack labeled ROOT into disk drive 0.

Be sure the RL02 disk pack is spinning and that the READY light is lit.

2) Insert the Distribution Cassette

If you have either a VAX-11/750 without the RL ROM or a VAX-11/730 processor, insert the distribution cassette into the console TU58 disk drive on the front of the processor (drive 0). (Note that the cassette label faces up.)

3) Prepare the CPU

If your CPU is a VAX-11/730, set the re-start switch to OFF and the rotary key switch to LOCAL.

If your CPU is a VAX-11/750, 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

72 ULTRIX-32 Installation Guide

set during the installation.

4) Prepare the DECwriter III (LA120)

If you have a DECwriter III (LA120), you must set up your console terminal before proceeding with the installation.

To set up the console, hold down <SETUP> and press D simultaneously until the LED display on the console (at the upper left-hand corner of the keys) is 0 (zero).

General Steps

The RL02 bootstrap procedure to create a working system consists of the following general steps:

- 1) Boot the Distributed Root Filesystem
- 2) Copy the RL02 Distributed Root to the System Disk
- 3) Set up and Install the /usr Filesystem
- 4) Update the Console Cassette (VAX-11/730 systems, only)

The next few subsections describe these steps in detail. References to disk drives are of the form $xx(n,m)$, where xx is one of the names in Table 2.6.1

Table 2.6.1 ULTRIX Disk Names and their Systems

Disk	xx	System
RA80	ra	VAX-11/730 & VAX-11/750
RA81	ra	VAX-11/730 & VAX-11/750
RL02 (IDC)	rb	VAX-11/730
RL02	rl	VAX-11/750
RM03	hp	VAX-11/750
RM05	hp	VAX-11/750
RM80	hp	VAX-11/750
RP06	hp	VAX-11/750
R80 (IDC)	rb	VAX-11/730

n is equal to the unit number of the disk drive you are booting off of. If your system is a VAX-11/730, and you are booting off of the distribution RL02, n is 1. If you are booting off any other disk, n is always 0. If your system is a VAX-11/750, n is always 0.

m is always specified in the command lines of this document. It is the partition (in the range 0 - 7).

2.6.1. Boot the Distributed Root Filesystem

The system must be halted; if it is not, type:

```
^P
>>>H
```

The console subsystem prompts with >>>. If you make a typing mistake, type <DELETE> to erase a character, <CTRL/U> to erase the line, or <CTRL/C> to interrupt the process.

The first step is to load the bootstrap program and boot the ULTRIX-32 root system. To do this, type the following command sequence which pertains to your system:

a) VAX-11/730

```
>>>I                (initialize the processor)
>>>L DD0:BOOT       (boot the cassette)
                    (this takes about one minute)
>>>S 2              (start the boot program)
Boot

: xx(n,0)vmunix     (boot the ULTRIX-32 kernel off the
                    root; this takes about five minutes)
                    (for the first boot, xx is rb and n is 1)

.
.
.
```

b) VAX-11/750

VAX-11/750 With an RL Boot ROM:

```
>>>I                (initialize the processor)
>>>B/3 DLA0
%%%
```

VAX-11/750 Without an RL Boot ROM:

```
>>>I                (initialize the processor)
>>>B DDA0           (boot the cassette)
                    (this takes about one minute)
%%%
= boot              (start the boot program)
                    (this takes about one minute)
```


2.6.1.1. Explanation of System Boot Messages

The first three numbers printed by the bootstrap programs are the sizes of different parts of the system (text, initialized and uninitialized data).

The ULTRIX-32 system then runs for the first time, beginning with a banner identifying the release and version of the system that is in use and the date it was compiled.

The *mem* messages give the amount of real (physical) memory and the memory available to user programs in bytes. For example, if your machine has 4mb of memory, then *xxxxxx* will be 4190208, 4096 bytes less than 4mb. The system reserves the last 4096 bytes of memory for use in logging system messages and does not count it as part of real memory. Available memory is the amount of physical memory left over for user processes after the ULTRIX-32 software has been loaded.

Information about the system is shown next, followed by messages showing what devices were found on the current processor. These messages are described in `autoconf(4)`. The distributed system may not have found all the communications devices (`dmfs` and `dzs`), or mass storage peripherals you have if you have more than two of one device. This is corrected later when you create a description of your machine from which to configure the ULTRIX-32 system. These messages contain much of the information that will be used in creating the configuration. In a correctly configured system most of the information present in the configuration description is printed out at boot time as the system verifies that each device is present.

The ULTRIX-32 system is running in single-user mode. The `#` prompt is from the ULTRIX-32 operating system, and tells you that you are the superuser.

2.6.2. Copy the RL02 Distributed Root to System Disk

In this step you will copy the entire RL02 distribution onto your system disk. On a VAX-11/730 system, the system disk can be an RA80, RA81 or R80. On a VAX-11/750 system, the system disk can be an RA80, RA81, RM03, RM05, RM80 or RP06.

If the system disk is removable, load it into drive 1. Whether the system disk is removable or not, make sure it is spinning and the READY light is lit. Copy the RL02 distribution root to the new system disk by typing:

```
# dest=xxxa devtype=dddd source=yyy cproot
```

where *xxxa* is the name of the destination (system) disk drive, unit number, and partition *a*; *dddd* is the type of the destination disk (system), and *yyy* is the name of the source (distribution) disk drive (see Table 2.6.2). Table 2.6.2 is a reference of the disk drive ULTRIX-32 system naming conventions.

76 ULTRIX-32 Installation Guide

Table 2.6.2 ULTRIX Disk Drive Naming Conventions

Disk	xxxx	dddd	yyy	System
RA80	ra0a	ra80		VAX-11/730 and 11/750
RA81	ra0a	ra81		VAX-11/730 and 11/750
R80 (IDC)	rb0a	rb80		VAX-11/730
RL02 (IDC)			rb1	VAX-11/730
RL02			rl0	VAX-11/750
RM03	hp0a	ra80		VAX-11/750
RM05	hp0a	ra80		VAX-11/750
RM80	ra0a	ra81		VAX-11/750
RP06	hp0a	rp06		VAX-11/750

For example, if you have a VAX-11/730 with an RL02 (drive 1) and an R80 (drive 0) disk, type:

```
# dest=rb0a devtype=rb80 source=rb1 cproot
```

The system responds with something similar to:

```
Warning: 36 sector(s) in last cylinder unallocated
/dev/rrb0a: 15884 sectors in 199 cylinders of 2 tracks, 40 sectors
8.1Mb in 13 cyl groups (16 c/g, 0.66Mb/g, 192 1/g)
super-block backups (for fsck -b#) at:
32, 1352, 2592, 3912, 5152, 6472, 7712, 9032, 10272, 11592,
12832, 14152, 15392,
```

```
DUMP: Date of this level 0 dump: Thu Apr 5 09:17:24 1984
```

```
.
.
DUMP: dumping (Pass IV) [regular files]
Warning: ./lost+found: File exists
DUMP: DUMP: 2042 tape blocks on 1 tape(s)
DUMP: DUMP IS DONE
set owner/mode for '.'? [yn]
```

Do not be concerned with the warnings.

NOTE: Save the super-block numbers. They may prove helpful in an emergency.

Type *y* in response to the prompt:

```
set owner/mode for '.'? [yn] y
Sync, halt, and reboot the destination rb0a rb02.
#
```

Sync and halt the system:

```
# sync
# halt

                    syncing disks... done

    halting (in tight loop); hit
      ^P                    (type <CTRL/P>)
    HALT ^P                (type H)
?02 PC=xxxxxxxxx         (system counter)
>>> H
```

Now reboot the system off the new system disk (see subsection 2.6.1). Remember to substitute the system disk for *xx*, and the system drive for *n*. The system drive (*n*) should be 0. For example, if your system disk is an RM05, substitute *hp(0,0)vmunix* for *xx(n,0)vmunix*.

The System is Rebooted

After the system is rebooted, use *fsck* to be sure the filesystem is correct:

```
# fsck /dev/xxxx
```

xxxx is the destination (system) disk, unit number, and partition you used in the *cproot* command above. For example, if the destination drive is an R80 in drive 0 and on partition *a*, type:

```
# fsck /dev/rb0a
```

In our example, the output should be similar to:

```
** /dev/rb0a
** Last Mounted on /
** Phase 1 Check Blocks and Sizes
** Phase 2 Check Pathnames
** Phase 3 Check Connectivity
** Phase 4 Check Reference Counts
** Phase 5 Check Cyl Groups
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

2.6.3. Set up and Install the /usr Filesystem

In this step you will set up and restore the */usr* filesystem.

78 ULTRIX-32 Installation Guide

First, set the date and time:

```
# date yymmddhhmm
```

Where:

yy equals the last two digits of the year
mm equals the number of the month of the year
dd equals the number of the day of the month
hh equals the hour of the day (based on 24 hours)
mm equals the minutes of the hour

The system will echo the new date and time.

Now set a shell variable to the name of your system disk, its device type, and your source (distribution) type (r102 or rb80). First, find your system disk drive in the following list and type the commands to the right:

<u>Disk Drive</u>	<u>Command</u>	<u>Command</u>
RA80	dest=ra0h	devtype=ra80
RA81	dest=ra0h	devtype=ra81
R80 (IDC)	dest=rb0h	devtype=rb80
RM03	dest=hp0g	devtype=rm03
RM05	dest=hp0g	devtype=rm05
RM80	dest=hp0g	devtype=rm80
RP06	dest=hp0g	devtype=rp06

For example, if your destination (system) disk is an R80, type:

```
# dest=rb0h  
# devtype=rb80
```

Now set up a shell variable for the source disk. It is usually r10, but if you have a VAX-11/730 with an RL02 and an R80 drive, it is rb1. For example, if you have a VAX-11/730 system with an RL02 and an R80, type:

```
# source=rb1
```

Otherwise, type:

```
# source=r10
```

Now export the three new variables:

```
# export dest devtype source
```

Now restore the root filesystem by executing the xrl script:

```
# xrl
Warning: 94 sector(s) in last cylinder unallocated
/dev/rrb0h: 110142 sectors in 254 cylinders of 14 tracks, 31 sectors
          56.4Mb in 16 cyl groups (16 c/g, 3.56Mb/g, 1600 i/g)
super-block backups (for fsck -b#) at:
 32, 7008, 13984, 20960, 27936, 34912, 41888, 48864, 55840, 62816,
 69792, 76768, 83744, 90720, 97696, 104672,
Mount RL02 Distribution Pack number 1.
Hit carriage return when pack is spinning and ready light comes on.
```

NOTE: Save the super-block numbers along with the other super-block numbers. Do not worry about the warning.

At this point, spin down the ROOT system disk and remove it from the drive. Load the distribution pack labeled 1 of 4 into the empty drive. When the disk is spinning and the READY light is lit, press <RETURN>. The system responds:

```
rb1c: hard error sn0                (Once is fine, twice means problems)
                                     (wait about 20 minutes)

Done with pack number 1.
```

```
Mount RL02 Distribution Pack number 2.
Hit carriage return when pack is spinning and ready light comes on.
```

Repeat changing RL02 distribution disks until the system tells you that you have finished (signified with a #).

Next, set the superuser password. The password you select must contain at least six characters:

```
# passwd root                        (set password for superuser)
New password:                        (password will not echo)
Retype new password:
#
```

Now unmount the destination (system) disk drive:

```
# /etc/umount /dev/${dest}
```

Remember to type the braces ({}).

80 ULTRIX-32 Installation Guide

Check the consistency of the filesystem by typing:

```
# fsck /dev/r${dest}
```

The output from `fsck` should look similar to:

```
** /dev/rxxxx                (xxxx is your dest assignment)
** Last Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
671 files, 3497 used, 137067 free (75 frags, 34248 blocks)
```

If there are inconsistencies in the filesystem, the system may prompt you for corrective action; see *fsck – the UNIX File System Check Program* for further information. If serious problems exist, try the extractions again, starting with the `xrl` script above. Be sure your shell variables are defined correctly.

Now remount the filesystem by typing:

```
# /etc/mount /dev/${dest} /usr
```

2.6.4. Update the Console Cassette

If you have a VAX-11/730 system, you need to build the boot cassette. You do this by copying some files from the local console cassette (that is, the cassette that came with your hardware), adding some files from the ULTRIX-32 system, and writing them all to a blank cassette.

This new boot cassette will allow the ULTRIX-32 system to reboot automatically whenever necessary.

First, remove the distribution cassette from the external TU58 drive (drive 0) and insert a blank cassette. Be sure the blank cassette is NOT write protected (the record tab should be in the left-most position).

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

Then, change into the directory where the console cassette information is stored:

```
# cd /sys/cassette
```

Next set up the default boot device. If you are using an IDC R80 as your primary root type:

```
# cp defboo.rb defboo.cmd
```

If you are using a drive on a UDA50 as your primary root type:

```
# cp defboo.ra defboo.cmd
```

To build the updated boot cassette, type:

```
# make update
```

(this takes about 25 minutes)

```
lists all of the files on the cassette
```

```
Console cassette update complete.  
The new cassette should be installed in the internal  
tu58 slot (Drive 1).
```

```
#  
#
```

Store the distribution cassette and the local console cassette in a safe place. The original local console cassette is required to boot VAX/VMS systems. Remove the boot cassette you just built from the external drive and insert it in the internal TU58 cassette drive (drive 1).

At this point, you can make more copies of the new boot cassette by using the `fcopy` maintenance command (see `arff(8V)`).

The system is now bootstrapped, and running on your system disk in single-user mode. Be sure to store the original RL02 distribution disks in a safe place.

See section 3 (System Setup) for information on how to further set up your system.

3. System Setup

Section 3 describes the system setup procedures for an ULTRIX-32 operating system. Before using the system setup procedures, your ULTRIX-32 system must first be bootstrapped (section 2). The only time you will need to use the system setup procedures is the first time you install the system, or whenever you change the system configuration.

The steps for setting up your system are as follows. Each step is described in the accompanying subsection.

- Initialize `/etc/fstab` (subsection 3.1)
- Automatic crash dumps (subsection 3.2)
- Make device special files (subsection 3.3)
- Build the new system images (subsection 3.4)
- Test the new system (subsection 3.5)
- Terminal configuration (subsection 3.6)
- Add site-specific information (subsection 3.7)
- Bring system to multi-user mode (subsection 3.8)
- Add new users (subsection 3.9)

3.1. Initialize `/etc/fstab`

In this step you will initialize information about the disk partition usage in the `/etc/fstab` file.

To initialize the information, first change directory to `/etc`. Then copy the `fstab.` file that corresponds with your system disk to `fstab`. Copy the appropriate file from the following list:

- `fstab.ra60`
- `fstab.ra80`
- `fstab.ra81`
- `fstab.rb80`
- `fstab.rk07`
- `fstab.rm03`
- `fstab.rm05`
- `fstab.rm80`
- `fstab.rp06`
- `fstab.rp07`

84 ULTRIX-32 Installation Guide

For example, if you have RK07 disk drives, then you should enter these commands:

```
# cd /etc
# cp fstab.rk07 fstab
```

The `fstab` file needs a correct entry for each filesystem that you will mount and use. As distributed, the `fstab` file has entries for the most frequently used filesystems.

To see what your particular `fstab` file contains use the `cat(1)` command. For example, after copying `stab.rk07` to `fstab`, the `fstab` file contains:

```
# cat fstab
/dev/hk0a:/:rw:1:1
/dev/hk0g:/usr:rw:1:2
/dev/hk1a:/usr/sys:rw:1:3
```

If your system disks are RK07s then you need to edit the `fstab` file and add an entry for the `g` partition.

But first you need to be sure `newfs(8)` has been executed on the `g` partition once.

CAUTION: Doing a `newfs` destroys all data on the partition; therefore, use caution when using `newfs`.

Then mount the partition onto the `/usr` directory and make the `/usr/users` directory:

```
# cd /usr
# newfs rhk1g rk07
# /etc/mount hk0g /usr
# mkdir /usr/users
# cd /etc
```

Be sure to save the numbers generated by `newfs`. A good place to keep them is taped to the outside of the disk cover (if the disk is removable).

Now use the `ed(1)` editor to add an entry like this to `fstab`:

```
/dev/hk1g:/usr/users:rw:1:4
```

See *The ed Command Summary Sheet*, included in this document, for a brief introduction to the `ed` editor.

For other types of system disks you should add an entry for `/usr/users` as well. See `fstab(5)` and *Guidelines for System Management* for further information on the `fstab` file.

Now change directory to `/`, clear the mount table, and check all the file systems. To

do this, enter these commands:

```
# cd /
# sync
# umount /dev/usr
# cat /dev/null > /etc/mtab
# sync
# fsck
# mount -a
```

You should use the `df(1)` command to be sure all the filesystems are correct:

```
# df
```

Filesystem node	total kbytes	kbytes used	kbytes free	percent used	Mounted on
/dev/hk0a	xxxx	xxxx	xxxx	xx%	/
/dev/hk0g	xxxx	xxxx	xxxx	xx%	/usr
/dev/hk1a	xxxx	xxxx	xxxx	xx%	/usr/sys
/dev/hk1g	xxxx	xxxx	xxxx	xx%	/usr/users

3.2. Automatic Crash Dumps

If a crash occurs, two files are created: one is the kernel image and one is a copy of the physical memory. These files require a substantial amount of disk space.

To prevent a crash from creating the files on the disk, the ULTRIX-32 system, as distributed, masks the following line in the `/etc/rc.local` file:

```
# /etc/savecore /usr/adm/crash > /dev/console
```

If you have enough disk space on your system to accommodate the crash files, in the event of a crash, then edit the `/etc/rc.local` file and remove the `#` masking the above line.

See *Guidelines for System Management* for information on `savecore`.

3.3. Make Device Special Files

Special files kept in the `/dev` directory access the ULTRIX-32 system devices. The `MAKEDEV` shell script creates these special files.

In this step you will install devices in your system by first creating a new device directory, and then copying the `MAKEDEV` script into it. To do this, enter these com-

86 ULTRIX-32 Installation Guide

mands:

```
# cd /  
# mkdir newdev  
# cp dev/MAKEDEV newdev  
# cd newdev
```

Next, determine the ULTRIX-32 naming conventions for the devices you need to install. Use Table 3.1 as a guide. For example, an RP06 disk is referred to as hp, and a TE16 tape drive is referred to as ht.

Table 3.1 ULTRIX-32 Device Naming Conventions

<i>MASSBUS Disks</i>		<i>MASSBUS Tape</i>	
Disk name	ULTRIX name	Tape name	ULTRIX name
RM03	hp	TE16	ht
RM05	hp	TU77	ht
RM80	hp	TU78	mt
RP06	hp		
RP07	hp		

<i>UNIBUS Disks</i>		<i>UNIBUS Tapes</i>	
Disk Name	ULTRIX Name	Tape Name	ULTRIX Name
RA60	ra	TS11	ts
RA80	ra	TU80	ts
RA81	ra		
RC25	ra		
RK07	hk		
RL02	rl		

<i>Communication Devices</i>		<i>Other Devices</i>	
Device Name	ULTRIX Name	Device Name	ULTRIX Name
DMF32	dmf	LP11	lp
DZ11	dz		
DZ32	dz		

<i>IDC Disks</i>	
Disk Name	ULTRIX Name
RL02	rb
R80	rb

Now that you have determined the device names, run the MAKEDEV script. To run

88 ULTRIX-32 Installation Guide

MAKEDEV, enter the following:

```
# MAKEDEV dev1 dev2 ... std
```

where *dev1 dev2 ...* are the device name and unit number of each device you need to install. For example, the first DZ-11 multiplexor would be called **dz0**. You must include **std** in the argument list. **std** causes MAKEDEV to create special files for standard devices such as the machine console (`/dev/console`), the console floppy disk interface for the VAX-11/780 (`/dev/floppy`), and the console cassette interfaces for the VAX-11/750 and VAX-11/730 (`/dev/tu0` and `/dev/tu1`).

For example, suppose your system has these devices:

```
two DZ-11 multiplexors
one DMF-32 multiplexor
an RM03 disk as drive 0
an RP06 disk as drive 1
one TE16 tape drive
```

In this case you would use this MAKEDEV command line:

```
# MAKEDEV dz0 dz1 dmf0 hp0 hp1 ht0 std
```

Finally, replace the old special files with the new special files you just created. To do this, enter the following commands:

```
# cd /
# mv dev olddev ; mv newdev dev
# chmod 755 dev
# sync
```

3.4. Build New System Images

In this step you will build a new system image (kernel) of your system using the `config` program.

These are the basic steps:

1. Create a configuration file for your system. You can either modify an existing file distributed with the ULTRIX-32 system, or create a configuration file from scratch.
2. Make a directory in which the system will be constructed.
3. Run `config` on the configuration file. `config` generates the files required to compile and load the system image with `make(1)`.
4. Construct source code interdependency rules for the configured system.
5. Compile and load the system with `make`.

This procedure results in a configured system image called `vmunix`.

Read the document *Building an ULTRIX-32 System with Config*, included in this binder, for a detailed description of how to perform these steps.

3.5. Test the New System

In this step you should test the new system and make sure it boots properly.

Your current working directory should now be /sys, which is where your system was configured. For example, if your configuration file is called MYVAX, the configuration procedure should have left you in the /usr/sys/MYVAX directory.

First, copy the system image to the root directory. Name the new copy /newvmunix to avoid destroying the working system until you are sure it works:

```
# cp vmunix /newvmunix
# sync
# cd /
# umount -a
```

Now halt the processor by typing:

```
# halt
```

The system responds with messages, and eventually tells you to press ^P. Press <CTRL/P> to return to the console subsystem, which has a >>> prompt:

```
^P
```

```
>>>
```

If you have a VAX-11/780, type the following:

```
>>>H                (halt the processor)
    HALTED AT xxxxxxx (program counter)
>>>U                (unjam the SBI)
>>>I                (initialize the processor)
    INIT SEQ DONE
```

Now boot the system. If you have a VAX-11/780 or VAX-11/730, type:

```
>>> B ANY
```

90 ULTRIX-32 Installation Guide

If you have a VAX-11/750, type:

```
>>> B/3
```

After printing some boot information, the system responds with this prompt:

```
Boot  
:
```

Now type `xx(n,0)newvmunix`, where `xx(n,0)` is the device specification you used to boot the completed root file system (see section 2).

```
Boot  
: xx(n,0)newvmunix
```

After you have typed the device specifications, the system displays several lines of information. When you receive a `#` prompt, you know the system has booted properly, and you are now running in single-user mode.

Every time you make a new kernel you should copy the old version of the system to another file before installing the new system. You should make this backup copy once you feel comfortable with the new system — you can run `fsck` to check the file system.

To make a backup copy and install the new system image, enter these commands:

```
# mv vmunix genvmunix  
# mv newvmunix vmunix  
# sync  
# /etc/mount /usr
```

NOTE: Keep a permanent copy of the old system under a different name. In particular, retain the generic distribution version of the system permanently as `/genvmunix` for use in emergencies.

3.6. Terminal Configuration

In this step you will configure terminals to allow simultaneous access to your system by many users. The configuration procedure assumes:

- The devices have been configured in to the system with an appropriate line in the configuration file (see *Building an ULTRIX-32 System with Config*).
- A special file exists in the `/dev` directory for each device (see subsection 3.1 for MAKEDEV information).

To configure your terminals do the following:

1. **Edit the `/etc/ttys` file.** (see `ttys(5)`) The `/etc/ttys` file contains a list of terminal names. By convention, terminals connected via dz interfaces are named `ttyDD` where `DD` is a decimal number, also known as the minor device number. The lines

on dz0 are named /dev/tty00, /dev/tty01, ... /dev/tty07. Lines on dmf interfaces are named **ttyhX**, where *X* is a hexadecimal digit. If more than one dmf interface is present in a configuration, successive terminals are named **ttyiX**, **ttyjX**, etc.

NOTE: Do not use arbitrary terminal names, because accounting programs and the **ps(1)** command use the last two characters of the tty naming conventions.

The first character of each line is either a 0 or a 1. If the first character is a 1, the **init(8)** program creates a login process for that line. To add a new terminal to your system, change the first character of the appropriate line of /etc/ttys to 1, or add a new line to the file if necessary.

NOTE: Do not remove the 1 from the line describing the console terminal.

The second character of each line lists the speed and initial parameter settings for the terminal. (See **gettytab(5)** for a complete explanation of the parameter settings, and look at the /etc/gettytab file.) The most common choices are:

0	300-1200-150-110
2	9600
3	1200-300
4	300 (console)
5	300-1200
8	9600-300

For example, the line 12tty00 in the /etc/ttys file describes a new terminal (/dev/tty00) set to run at 9600 baud. The line 10tty62 describes a new terminal (/dev/tty62) set to run at 300, 1200, 150, 110 baud. In this case, the terminal would start up at the first speed (300). The user could then switch to the appropriate speed for his particular terminal by pressing the <BREAK> key.

2. **Edit the /etc/ttytype file.** The /etc/ttytype file contains a list of terminal types for each tty port on the system. For example, the line **vt100 tty00** identifies tty00 as a VT100 terminal. Edit the /etc/ttytype file, adding the type of each terminal on your system (see **ttytype(5)** for details). Use the /etc/ttytype file distributed with the system as a model.
3. **Set dialup terminals.** For dialup lines, carrier is not assumed, therefore you must be sure to set the flag in /sys/conf/MACHINENAME. For hardwired terminals, the system configuration file (/sys/conf/MACHINENAME) must show that the carrier is assumed to be present. See **dz(4)**, and **dmf(4)** for further information.

NOTE: The ULTRIX-32 operating system currently supports no more than 100 terminals.

Next, change the names of any dialup lines to ttydx where *x* is in the range [0-9]

92 ULTRIX-32 Installation Guide

or [a-f]. Some programs use this naming convention to determine if a terminal is a dialup.

For example, suppose you want to set lines tty06 and tty07 to make dialup terminals. First change the name of the devices in the /dev directory with commands similar to these:

```
# mv tty06 ttyd0
# mv tty07 ttyd1
```

Now change the names of the new dialup lines in the /etc/ttys file. An example of a dialup line entry in this file is:

```
13ttyd0
```

Finally, change the names of the dialup lines in the /etc/ttytype file. A example entry of an entry in the /etc/ttytype file for a dialup terminal is:

```
dialup ttyd0
```

You may also wish to set the flags field. This flag is explained in section 1.2.3 of *Building an ULTRIX-32 System with Config*.

You can automate this process by putting the commands needed to change line names in the /dev/MAKEDEV.local file.

3.7. Site Tailoring

In this step you will set your system specifications. All programs that require the site's name, or site specific information obtain the information through system calls, or from files located in /etc.

To tailor your system site, edit the files /etc/rc.local and /etc/hosts. Choose a site name, and add it to the first line of both files.

In the /etc/rc.local file, substitute the name of your site for **myname** in the following line:

```
/bin/hostname myname
```

This line defines the value returned by the gethostname(2) system call. Programs such as getty(8), mail(1), wall(1), uucp(1), and who(1) use this system call.

In the /etc/hosts file, substitute the name of your site for **myname** in the following line:

```
127.1 myname localhost
```

3.8. Bring the System to Multi-user Mode

You are now ready to bring the system to multi-user mode. To do this, type <CTRL/D>. The system responds with system related information, followed by a login prompt:

```
# ^D
.
.
.
(system related information)
.
.
.
login:
```

The login prompt should appear on every terminal you configured in to the system.

NOTE: If the login prompt does not appear on the terminals, check the flag settings on the dz and dh entries in the configuration file for your system, /sys/conf/MACHINENAME. If the flags are set, you can wire terminals in to the system with no modem control. If the flags are not set, the line requires carrier. If necessary, set the flags for hardwired lines that are not properly wired. See dz(4) and dh(4) for details.

Once you receive the login prompt, log in to the system as root and add new users to the system:

```
login: root
Password:

#
```

NOTE: Your system prompt will actually be *sitename#*, which indicates you are logged in as root and the system is in multi-user mode. *sitename* is the name of your particular site which you entered in the /etc/rc.local and /etc/hosts files.

For the rest of this document, # indicates the superuser prompt, regardless of whether the system is in multi-user mode or single-user mode.

3.9. Add New Users

In this step you will add login accounts for the initial user community. Each login account needs an entry in /etc/passwd and a login, or home, directory.

First, select a location (disk and partition) for the directory called /usr/users that will hold the home directories. Second, create /usr/users, and if the directory has its own partition, mount the partition. Once these steps are completed, you are ready to add

94 ULTRIX-32 Installation Guide

accounts to the system. Section 3.1 of *Guidelines for System Management* explains in detail how to add accounts.

If at all possible, you should place `/usr/users` on its own separate partition. For better performance and ease of maintenance, you should try to locate this partition on a disk other than the system disk. If you do not have a separate disk available, however, a partition on the system disk will work.

To learn which partitions are available, you need to determine which partitions overlap, and which partitions are in use. Use the `mount(8)` command to see a table of all the file systems and the partition each is using:

```
# /etc/mount
```

Table 5.1 lists the partitions for most major disk drives, their sizes, and their typical uses.

3.9.1. Separate Partition Is Available

A partition of a disk separate from the system disk is the best location for the home directories. If your system can accommodate this situation, then choose a partition on the separate disk to contain `/usr/users`.

If no partition is available on a disk other than the system disk, then the second best choice is an unused partition on the system disk. For example, suppose you have only one disk, an RA81, and `/usr` is on partition `h`; partition `g` is available. In this situation, the best location for `/usr/users` is partition `g` (`ra0g`).

Now add an entry for the partition to `/etc/fstab`. The entry ensures that the file system will be checked automatically and mounted on reboot. For example, in our case above with the RA81 disk, the entry in `/etc/fstab` should be:

```
/dev/ra0g:/usr/users:rw:1:3
```

NOTE: The order of the entries in `/etc/fstab` is important. For example, the `fstab` entry for `/usr` must appear before any other entries that are mounted on `/usr`. See `fstab(8)` for more information on the format and meaning of entries in this file.

The new partition (`ra0g`, in our example) must contain a file system. Use the `newfs(8)` command to create the file system. For example:

```
# /etc/newfs ra0g ra81
```

CAUTION: If you create a file system on a partition that already contains files, all the files will be destroyed.

The next command creates the empty directory `/usr/users`:

```
# mkdir /usr/users
```

Use the `mount(8)` command:

```
# /etc/mount /usr/users
```

Proceed with subsection 3.9.3.

3.9.2. Separate Partition is not Available

If you do not have any separate partitions available (either on the system disk or any other disk), simply create `/usr/users`. The user files will then occupy space on the `/usr` partition of the system disk.

```
# mkdir /usr/users
```

3.9.3. Add the User Accounts

You are now ready to add the new user accounts. The document *Guidelines for System Management* (also included in this binder) describes in detail how to create the new user accounts. These are the basic steps:

- Create a user entry in `/etc/passwd`.
- Create a user home directory.
- Copy the shell startup files.
- Change the UIDs and GIDs assigned to the newly created files.

This completes your ULTRIX-32 system installation for a standard system configuration. The remainder of this document supplies information for network configuration, fine-tuning your system, dealing with nonstandard configurations, and additional setup information such as `sendmail` and `uucp`.

4. Network Setup

The ULTRIX-32 system provides support for the TCP, IP, and UDP protocols. Section 4 describes how to configure your system to use the networking support.

If your machine is not connected to another machine, read subsections 4.1 and 4.2. If you plan networking links to other machines, read the entire section.

Programs such as `mail(1)`, `talk(1)`, `script(1)`, and the line printer daemon `lpd(1)` rely on information outlined in this section to function properly. This section contains three subsections:

- The Configuration File

Subsection 4.1 outlines the steps to configure network hardware into your machine. It also tells how to configure pseudodevices that the system needs even if your machine is not connected to other machines.

- Network Databases

Subsection 4.2 discusses the database files that the network library routines and server programs use. You need to add site-specific information to some of these files; others can remain unaltered.

- Routing and Gateways

Subsection 4.3 explains how to set up routing information so that packets can be routed correctly. It is relevant for ARPANET sites, or sites running large, complex networks.

4.1. System Configuration

Section 4.1 outlines the steps required to configure networking hardware into your system. It discusses the needed information in your machine's configuration file, and how to configure pseudodevices.

4.1.1. The Configuration File

To configure network hardware in your system, place an appropriate entry in the configuration file for your machine, `/sys/conf/MACHINE` (`MACHINE`, written in capital letters, is the name of your system). The appropriate entry depends on the kind of hardware you plan to configure. Refer to section 4 of the *ULTRIX-32 Programmer's Manual* (for example `de(4)`) for the description of the intended device, then check the SYNTAX section to get the exact wording for the entry. Here are the supported devices:

Device Name	Description
de	DEUNA Ethernet
dmc	DMR-11 (in DMC mode)

For example, to configure a DEUNA Ethernet interface, refer to `de(4)`, then edit your configuration file, adding this line:

```
device de0 at uba? csr 0174510 vector deintr
```

4.1.2. Configuring Multiple Networks

The order in which devices are listed in the configuration file is important. If you are configuring more than one network in your system, list the most widely known network first. For example, if your machine is connected to both the ARPANET and a local network, list the device connected to the ARPANET first.

4.1.3. Configuring Pseudoterminals

Whether or not your machine is connected to a network, programs such as mail, talk, and `lpd` depend on information defined in your machine's configuration file to operate properly. The ULTRIX-32 system is distributed with this information already included.

Specifically, a properly installed ULTRIX-32 system defines the `INET` option in the configuration file for your machine. The configuration file also includes definitions for these pseudodevices:

Pseudodevice Name	Function
<code>inet</code>	Includes IPC and network code
<code>pty</code>	Forces pseudoterminal device driver to be configured into the system
<code>loop</code>	Used by mail system and in network testing
<code>ether</code>	Maps Internet and Ethernet addresses

Do not remove these pseudodevice specifications from your configuration file.

To use the pseudoterminals, create device entries for them in the `/dev` directory. The

following creates entries for 16 pseudoterminals:

```
# cd /dev
# MAKEDEV pty0
```

Sixteen entries is adequate for most networking needs. If you have a heavy network load, you can create additional pseudoterminal entries by adding `pty1` and `pty2` to the command line.

This procedure actually creates two files in `/dev` for each pseudoterminal: a master file, and a slave file. The master pseudoterminal files are called `/dev/pty[p-r][0-9a-f]`; the slave pseudoterminal files are called `/dev/tty[p-r][0-9a-f]`. The `MAKEDEV` command above creates a total of 16 master pseudoterminals and 16 slave pseudoterminals.

The master pseudoterminals are:

```
ptyp0  ptypa
ptyp1  ptypb
ptyp2  ptypc
ptyp3  ptypd
ptyp4  ptype
ptyp5  ptypf
ptyp6
ptyp7
ptyp8
ptyp9
```

The slave pseudoterminals are:

```
ttyp0  ttypa
ttyp1  ttypb
ttyp2  ttypc
ttyp3  ttypd
ttyp4  ttype
ttyp5  ttypf
ttyp6
ttyp7
ttyp8
ttyp9
```

Now install the pseudoterminals by making sure there is an entry for each slave pseudoterminal in the `/etc/ttys` and the `/etc/ttytype` files. The `/etc/ttys` file lists the terminals connected to your machine, and the `/etc/ttytype` file identifies each terminal type.

If some of the entries are missing from the `/etc/ttys` file, edit the file, and add the necessary lines. The `/etc/ttys` file should contain one entry per slave pseudoterminal

100 ULTRIX-32 Installation Guide

created by the MAKEDEV command:

```
02ttyp0
02ttyp1
02ttyp2
.
.
.
02ttyp9
02ttya
02ttyb
.
.
.
02ttyf
```

The zero (0) in the first column prevents a `getty(8)` process from starting on these terminals when the system goes into multi-user mode. The two (2) in the second column specifies the baud rate; however, because `getty(8)` will ignore the entry, the two is significant only as a place-holder.

Now be sure each pseudoterminal is identified as type `network` in the `/etc/ttytype` file. If `/etc/ttytype` is missing any entries, edit the file and add any necessary lines. In our example, the `/etc/ttytype` file should contain the following lines – one per slave pseudoterminal listed in `/etc/ttys`:

```
network  ttyp0
network  ttyp1
.
.
.
network  ttyp3
network  ttya
network  ttyb
.
.
.
network  ttyf
```

4.2. Network Data Bases

Network library routines and server programs rely on information contained in several data base files. A few of these files must be altered to reflect site-specific information; others remain untouched. Because some files need altering, even if you do not plan networking at your site, you should be aware of these data base files. Table 4.2.1 lists these files and provides a brief description of each.

Table 4.2.1 Network Data Base Files

Table 4.2.1 Network Data Base Files

File	Manual Reference	Function	Action Required
/etc/rc.local	rc(8)	Script for starting servers	Requires modification, even if your machine is not on a network.
/etc/hosts	hosts(5)	Host names	Requires modification, even if your machine is not on a network.
/etc/networks	networks(5)	Network names	Requires modification, even if your machine is not on a network.
/etc/hosts.equiv	rshd(8C)	Trusted hosts	Requires site-specific network information. Be careful when you set up this file.
/etc/ftpdusers	ftpd(8C)	Unwelcome ftp users	Requires site-specific network information.
/etc/services	services(5)	Network services	No modification necessary.
/etc/protocols	protocols(5)	Protocol names	No modification necessary.

The following subsections provide details on these network databases.

4.2.1. Editing the /etc/rc.local File

The /etc/rc.local file contains site-dependent information needed at boot time. Edit this file using the `ed(1)` editor so it contains local information, even if your machine is not on a network. For a brief description of how to use `ed` see *The ed Command Summary Sheet* included in this document.

Identify your machine name in the first line of the /etc/rc.local file. To do this, edit /etc/rc.local, and modify this line by substituting *myname* with the name you choose for your machine:

```
/bin/hostname myname
```

For example, if you decide to call your machine *myvax*, substitute *myvax* for

102 ULTRIX-32 Installation Guide

myname. The line should now read:

```
/bin/hostname myvax
```

The `/etc/rc.local` file should also include a call to the `ifconfig(8C)` program that identifies your machine's name to the network interface at boot time. If your interface is something other than a DEUNA, edit the `/etc/rc.local` file substituting your interface for `de0` in the following line:

```
/etc/ifconfig de0 `bin/hostname`
```

Your substitution for `de0` should be comprised of the interface device name and unit number. For example, the first device you configure to a DMR-11 interface is called `dmr0`; the second is `dmr1`, and so forth. The first device is the one listed first in your machine's configuration file. The appropriate line for the first device on a DMR-11 interface is:

```
/etc/ifconfig dmr0 `bin/hostname`
```

NOTE: Enclose the word `/bin/hostname` with back quotation marks (```), not single quotation marks. Refer to the device descriptions in section 4 of *The ULTRIX-32 Programmer's Manual* for device name abbreviations.

The `/etc/rc.local` file is used in conjunction with the `/etc/rc` file at boot time. The `/etc/rc` file automatically starts these network servers:

Server Name	Function	Manual Reference
<code>/etc/rshd</code>	Remote shell server	<code>rshd(8C)</code>
<code>/etc/rexecd</code>	Remote execution server	<code>rexecd(8C)</code>
<code>/etc/rlogind</code>	Remote login server	<code>rlogind(8C)</code>
<code>/etc/rwhod</code>	System status daemon	<code>rwhod(8C)</code>

The site-dependent `/etc/rc.local` file includes by default commands to start other network servers. If you want these other network servers started at boot time, be sure to retain commands such as the ones shown here in `/etc/rc.local`:

```
if [ -f /etc/telnetd ]; then
    /etc/telnetd & echo -n 'telnetd'    >/dev/console
fi
```

NOTE: These particular commands start the TELNET server. Commands for other servers take a similar form.

The `/etc/rc.local` file for the ULTRIX-32 system is distributed with commands to start the following servers:

Server Name	Description	Manual Reference
* /etc/telnetd	TELNET server	telnetd (8C)
* /etc/ftpd	FTP server	ftpd (8C)
/etc/syslog	Error logging server	syslog (8)
/etc/sendmail	SMTP server	sendmail (8C)
* /etc/routed	Network routing daemon	routed (8C)

If you are on a network, you need all the servers listed; if you are not on a network, you do not need the servers marked with an asterisk (*).

If you do not need a server, either mask the appropriate lines in the /etc/rc.local file, or remove the server from the /etc directory. To mask a line, place a number sign (#) at the beginning of it. For example, masked lines for the TELNET server look like this:

```
#if [ -f /etc/telnetd ]; then
#   /etc/telnetd & echo -n ' telnetd'      >/dev/console
#fi
```

Consult *The ULTRIX-32 Programmer's Manual* for information concerning a particular server. For example, telnetd(8) describes the TELNET server. Also consult the *Sendmail Installation and Operation Guide*, for information on sendmail, and its relationship to the servers.

4.2.2. Editing the /etc/hosts and /etc/networks Files

The /etc/hosts file contains information identifying host names. The /etc/networks file lists network names and numbers. You use different procedures to generate both the /etc/hosts file and the /etc/networks file, depending on whether your site is connected to the ARPANET. The following two sections provide the information to generate these files.

4.2.2.1. Sites Connected to the ARPANET

If you are an ARPANET site, you generate the /etc/hosts and /etc/networks files by using information retrieved from a network data base at the Internet Network Information Center (NIC) at Stanford Research Institute. Follow the steps in the next paragraph to generate these files.

In the procedure listed in the following example, you use the /etc/gettable program to retrieve the NIC host data base. Then you use the /etc/htable program to convert

104 ULTRIX-32 Installation Guide

this data-base information to the format required by network library routines.

```
# cd /usr/src/ucb/netser/htable
# /etc/gettable sri-nic
Connection to sri-nic opened.
Host table received.
Connection to sri-nic closed.
# /etc/htable hosts.tst
Warning, no localgateways file.
#
```

The `htable` program generates `hosts` and `networks` files and places them in the current directory (`/usr/src/ucb/netser/htable`). If you want to make local modifications to these files (to define local hosts or networks, for example), create files in the current directory called `localhosts` and `localnetworks`. The `htable` program incorporates the information in the `localhosts` and `localnetworks` files when it creates `/etc/hosts` and `/etc/networks`. Refer to `hosts(5)` and `networks(5)`, for the format to use for these locally-generated files.

Before installing the newly-modified `hosts` and `networks` files in the `/etc` directory, you should compare the old files with the new ones using `diff(1)`.

To install the `hosts` and `networks` files in the system, type:

```
# mv hosts /etc/hosts
# mv networks /etc/networks
```

4.2.2.2. Sites Not Connected to the ARPANET

If your machine is not connected to the ARPANET, you must modify the `/etc/hosts` and `/etc/networks` files by hand. The contents of these files varies slightly depending on whether your machine is connected to other machines.

If your machine is not connected to other machines:

Your ULTRIX-32 system must have the `/etc/hosts` and `/etc/networks` files, even if your machine is not connected to other machines. Programs such as `mail(1)`, `talk(1)`, and the line printer daemon rely on these files to function correctly.

For machines not connected to other machines, edit the `/etc/hosts` file. In the following line in `/etc/hosts`, substitute the official name you chose for your system for

myname, and substitute any short abbreviation for *my*:

```
127.1 myname localhost my
```

For example, if the official name of your machine is *myvax* (defined in the `/bin/hostname` command in the `/etc/rc.local` file), and you choose *m* to be the abbreviation for *myvax* (the abbreviation is a shorthand version of the machine name; it should not contain punctuation characters), the edited line in `/etc/hosts` should look like this:

```
127.1 myvax localhost m
```

NOTE: Type 127.1 and localhost literally.

Do not edit the `/etc/networks` file if your machine is not connected to other machines. The contents of `/etc/networks` should be:

```
loop 127 loopback
```

If your machine is connected to other machines:

If your machine is connected to others machines, `/etc/hosts` should contain information identifying the machines with which your machine is connected. For example, if your machine is connected in a local network, you need to identify the connected machines in `/etc/hosts`.

As distributed, the `/etc/hosts` file reads:

```
127.1 myname localhost my
```

Edit this file so that the line reads:

```
127.1 localhost
```

Keep this as the first line. Then add a line containing your netaddress, the official name of your machine (as defined in your machine's `/etc/rc.local` file), and your official abbreviation. The netaddress is a number describing the address of your machine on a network, and takes the form:

```
netnumber.machinenumber
```

netnumber is a number describing the network in question. All sites on the same network have the same number. *netnumber* is a number of your own choosing, less than 127. It is best to choose a low number. *machinenumber* identifies the machine on the network. If you are using Ethernet hardware with Address Resolution Protocol, *machinenumber* is a number of your own choice. It is best to choose a low number. If you are not using Ethernet hardware, *machinenumber* is often the low byte number of the physical device address.

Add similar lines for each additional site on your network.

106 ULTRIX-32 Installation Guide

For example, suppose your machine is called *myvax*, and is connected through a local network identified as 1 to two other machines called *hisvax* and *hervax*. The edited */etc/hosts* file for *myvax* should be similar to:

```
127.1 localhost
1.1 myvax m
1.5 hisvax h
1.17 hervax hr
```

NOTE: These guidelines tell you how to configure class A networks with no sub-netrouting.

The */etc/networks* file contains information describing every network to which your machine is connected. For each network to which your machine is connected, add one line in the */etc/networks* file containing information in the following format:

networkname networknumber alias

In this format, *networkname* is the name of the network; *networknumber* is an identifying number for the network; and *alias* is an alternate name for the network, used by *netstat(1)* and other networking programs.

For example, if your machine is connected to the local DMR-11 network, and 1 is the identifying number for the network, edit the */etc/networks* file, adding a line like this:

```
dmrnet 1 dmrnetwork
```

If your machine is connected to another network as well, add another similar line describing that network.

For more information concerning the format of */etc/hosts*, and */etc/networks*, consult *hosts(5)* and *networks(5)*.

4.2.3. Editing the */etc/hosts.equiv* File

The remote login program *rlogind(8C)* and the remote shell server *rshd(8C)* authenticate user requests based on a list of trusted host names. Trusted hosts are machines considered secure, or under similar administrative control. Thus, if a user has an account on two equivalent machines (that is, both machines trust each other), the user can execute commands and log in to a remote machine from his own, without password verification on the remote machine.

The */etc/hosts.equiv* file contains the names of trusted hosts. Edit this file to contain the names of your system's trusted hosts. List the official name of each machine, one machine name per line. Official machine names are defined in each machine's */etc/rc.local* file.

The `/etc/hosts.equiv` file as distributed contains:

```
myname
```

Edit `/etc/hosts.equiv`, substituting the name of your system's trusted hosts for *myname*. Then add the official names of your system's other trusted hosts.

For example, if the official names of your system's trusted hosts are *hisvax* and *her-
vax*, and your system's name is *myvax*, the edited `/etc/hosts.equiv` should look like:

```
myvax  
hisvax  
hervax
```

4.2.4. Editing the `/etc/ftpusers` Files

4.2.4.1. The FTP Account

The ULTRIX-32 system provides support for an anonymous ftp (File Transfer Protocol) account. The account offers a location for users logging in through ftp to deposit and retrieve files from local users. You may want to create an anonymous account on your machine for guest users for any of these reasons:

- You are running ftp.
- You are on the the ARPANET or are talking to many other machines.
- Your machine communicates with nontrusted sites.

Security problems are inherent to such an account. To reduce the risk of tampering, there are restrictions imposed on the ftp account. When a remote user accesses this account, the ftp server `ftpd(8C)` executes a `chroot(2)` system call. This call confines the user to a restricted part of the file system that contains the ftp home directory.

Here are the steps to set up a guest ftp account. This procedure creates a subdirectory in the ftp home directory called `pub`. Local users who wish to place files in the anonymous area should put them in `~ftp/pub`. The procedure also provides the ftp server process with the programs and files it needs to execute properly, and ensures that all directories (other than `~ftp/pub`) and executable images are unwritable. The steps are:

108 ULTRIX-32 Installation Guide

```
# cd ~ftp
# chmod 555 .;chown ftp .; chgrp ftp .
# mkdir bin etc pub
# chown root bin etc
# chmod 555 bin etc
# chown ftp pub
# chmod 777 pub
# cd bin
# cp /bin/sh /bin/ls .
# chmod 111 sh ls
# cd ../etc
# cp /etc/passwd /etc/group .
# chmod 444 passwd group
```

This procedure limits the commands available to the ftp account to ls(1) and the shell sh(1). You can make more commands available, but it is wise to be conservative.

4.2.4.2. The /etc/ftpusers File

The /etc/ftpusers file contains names of unwanted users. The ftp server process checks this file on each connection. If the user's name appears in the unwanted list, the ftp server denies the unwanted user access to the machine. This prevents potentially destructive users from logging in to the ftp account on your machine. As distributed, the /etc/ftpusers file contains:

```
uucp
ingres
root
```

Edit this file if necessary, adding any unwanted user login names.

You can delete the /etc/ftpusers file if:

- Your machine is not on a network.
- You have removed the ftp server /etc/ftpd.

4.2.5. The /etc/services File

The /etc/services file maps network service names (for example systat, netstat, and smtp) to port numbers, and the protocol each service requires. Do not alter this file.

4.2.6. The /etc/protocols File

The /etc/protocols file maps protocol names (for example TCP, IP, and ICMP) to protocol numbers. Do not alter this file.

4.3. Routing and Gateways

Sites on the ARPANET and sites with large complex networks need to set up routing information so that packets can be properly routed. This subsection describes two methods of setting up the routing information:

1. Using the routing table daemon `/etc/routd` to maintain system routing tables.
2. Obtaining information from a smart gateway to dynamically create a routing database. This method is appropriate for ARPANET sites only.

Read this subsection if you access networks that are not directly attached to your host and need to set up routing information.

4.3.1. Maintaining Routing Tables with `/etc/routed`

The routing table management daemon `/etc/routed` uses a variation of the Xerox Routing Information Protocol to maintain up-to-date routing tables. The method `/etc/routed` uses to maintain a routing table is: 1) start `/etc/routed` from the `/etc/rc.local` file 2) read the `/etc/gateways` file 3) install the routes defined there. The `routed` daemon then broadcasts on each local network to locate other routing daemons, that is, other machines that are running the same type of `routed` daemon. If the `routed` daemon receives responses, all the routing daemons cooperate to maintain globally consistent routes for the local environment.

It is also possible for `/etc/routed` to cooperate with remote sites running routing daemons to maintain a table with routes for remote sites. Refer to `routed(8C)` for more information.

4.3.2. Using a Smart Gateway to Create a Routing Data Base

Sites on the ARPANET can choose to rely on information provided by a smart gateway to create a routing data base. By defining a wildcard route to a smart gateway, you can rely on the smart gateway to provide ICMP routing redirect information. To define a wildcard route, edit the `/etc/rc.local` file, adding an entry of this form:

```
/etc/route add 0 smartgateway 1
```

The wildcard route is indicated by a destination of 0 (zero). Refer to `route(8C)` for more information.

The system uses the wildcard route as a last resort to route packets to their destination. If the destination receiving the packets is able to generate the proper routing redirect information, the system adds routing table entries based on that information.

There are disadvantages to this approach. It is unsuitable in environments with pseudogateways that cannot generate routing redirect information. Also, if you rely on the smart gateway to provide up-to-date routing information and the smart gateway goes down, there is no practical way to maintain service.

ARPANET sites can choose to combine these two approaches. For example, you can use the routing table daemon to maintain routing information for local networks, and

110 ULTRIX-32 Installation Guide

use smart gateways for routing information for more distant networks. The `netstat(1)` command displays routing table contents and other routing information. The command `netstat -r` displays the contents of routing tables. Adding the option `-s` to the command indicates the number of routing table entries dynamically created from routing redirect information (`netstat -r -s`).

5. Additional Setup Information

Section 5 provides additional system setup details that you will need once your system is up and running and you are familiar with the needs of your users.

The additional information covered in this section is as follows:

- Subsection 5.1 provides information for fine tuning the system's disk configuration.
- Subsection 5.2 provides an overview of the line printer facility.
- Subsection 5.3 briefly describes the mail facility.
- Subsection 5.4 describes how to set up a UUCP connection.

5.1. Fine Tuning the Disk Configuration

This section provides information on file system layout and gives suggestions for better system performance.

5.1.1. Disk Filesystem Partitions

Each physical disk drive can have up to eight partitions; ULTRIX-32 typically uses only three or four partitions.

Some of the partitions overlap. Therefore, you should choose partitions carefully. The cylinders given in Table 5.1 will help you with the decisions.

Table 5.1 gives the use of each partition for each type of drive. Throughout the table, ? refers to the drive unit number.

Table 5.1 Disk Drive Partitions and Their Typical Uses

Disk Drive	Part-ition	Typical Use	Starting Sector	Sector Length	Cylinders
RA60	ra?a	root file system	0	15884	0-94
	ra?b	paging & swapping area	15884	33440	95-294
	ra?c	access entire pack	0	400176	0-end
	ra?g		49324	82080	295-783
	ra?h	/usr file system	131404	268772	784-end
RA80	ra?a	root file system	0	15884	0-37
	ra?b	paging & swapping area	15884	33440	38-115
	ra?c	access entire pack	0	237212	0-end
	ra?g		49324	82080	116-305
	ra?h	/usr file system	131404	105808	306-end

112 ULTRIX-32 Installation Guide

RA81	ra?a	root file system	0	15884	0-22
	ra?b	paging & swapping area	15884	33440	23-69
	ra?c	access entire pack	0	891072	0-end
	ra?d		340670	15884	478-500
	ra?e		356554	55936	501-579
	ra?f		412490	478582	580-end
	ra?g		49324	82080	70-184
	ra?h	/usr file system	131404	759668	185-end
R80 (IDC)	rb?a	root file system	0	15884	0-36
	rb?b	paging & swapping area	16058	33440	37-114
	rb?c	access entire pack	0	242606	0-558
	rb?g		49910	82080	115-304
	rb?h		132370	110143	305-558
RK07	hk?a	root file system	0	15884	0-240
	hk?b	paging & swapping area	15906	10032	241-392
	hk?c	access entire pack	0	53790	0-814
	hk?g	/usr file system	26004	27786	393-813
RL02 (UNIBUS)	rl?a	root file system	0	15884	0-397
	rl?b	paging & swapping	15920	4480	398-510
	rl?c	access entire pack	0	20480	0-511
	rl?d		15920	4520	398-510
	rl?g	/usr file system	0	20440	0-510
RL02 (IDC)	rb?a	root file system	0	15884	0-397
	rb?b	paging & swapping area	16000	4480	398-510
	rb?c	access entire pack	0	20480	0-511
RM03	hp?a	root file system	0	15884	0-99
	hp?b	paging & swapping area	1600	33440	100-309
	hp?c	access entire pack	0	131680	0-822
	hp?d		49600	15884	309-408
	hp?e		65440	55936	409-758
	hp?f		121440	10080	759-822
	hp?g	/usr file system	49600	82080	309-822
RM05	hp?a	root file system	0	15884	0-26
	hp?b	paging & swapping area	16416	33440	27-81
	hp?c	access entire pack	0	500384	0-822
	hp?d		341696	15884	562-588
	hp?e		358112	55936	589-680
	hp?f		414048	86176	681-822

	hp?g	/usr file system	341696	158528	562-822
	hp?h		49856	291346	82-561
RM80	hp?a	root file system	0	15884	0-36
	hp?b	paging & swapping area	16058	33440	37-114
	hp?c	access entire pack	0	242606	0-558
	hp?d		49910	15884	115-151
	hp?e		68096	55936	152-280
	hp?f		125888	120466	281-558
	hp?g	/usr/file system	49910	192510	115-558
RP05	hp?a	root file system	0	15884	0-37
	hp?b	paging & swapping area	15884	33440	38-117
	hp?c	access entire pack	0	171798	0-410
	hp?d		2242	15884	118-155
	hp?e		65208	55936	156-289
	hp?f		121220	50424	290-410
	hp?g		2242	122320	118-410
RP06	hp?a	root file system	0	15884	0-37
	hp?b	paging & swapping area	15884	33440	38-117
	hp?c	access entire pack	0	340670	0-814
	hp?d		49324	15884	118-155
	hp?e		65208	55936	156-289
	hp?f		121220	219296	290-814
	hp?g	/usr file system	49324	291192	118-814
RP07	hp?a	root file system	0	15884	0-9
	hp?b	paging & swapping area	16000	66880	10-51
	hp?c	access entire pack	0	1008000	0-629
	hp?d		376000	15884	235-244
	hp?e		392000	307200	245-436
	hp?f		699200	308600	437-629
	hp?g		376000	631800	235-629
	hp?h	/usr file system	83200	291346	52-234

The first partition (partition a) on all drives is normally reserved for the root file system, a backup of the root file system, or a small file system such as /tmp. The second partition (partition b) is used for paging and swapping. The third partition is reserved for accessing the entire pack; this allows pack-to-pack copying and other actions which require access to the entire physical disk. The remaining partitions' uses vary according to the disk type and the way you setup your system.

114 ULTRIX-32 Installation Guide

The disk partition sizes for a drive are based on a set of four default partition tables. For further information, see `diskpart(8)`. The particular table used is dependent on the size of the drive. The a partition is the same size across all drives, 15884 sectors. The b partition is sized according to the total space on the disk. For drives less than about 400 megabytes, the partition is 33440 sectors, while for larger drives the partition size is doubled to 66880 sectors. The c partition is always used to access the entire physical disk, including the space at the back of the disk reserved for the bad sector forwarding table.

If the disk is larger than about 250 megabytes, an h partition is created with size 291346 sectors, and regardless of whether the h partition is created, the remainder of the drive is allocated to the g partition.

If you want to split up the g partition into a number of smaller file systems, you may use the d, e, and f partitions, all of which overlap partition g. The default sizes for these partitions are 15884 for partition d, 55936 for partition e, and the remainder of the disk for partition f.

Drives on UDA50 and IDC controllers do not completely follow these rules. For example, the swap partition on an RA81 is only 33440 sectors, and there are not any d, e, or f partitions available on an RA60 or RA80. See `uda(4)` for more information.

5.1.2. Space Available

The space available on a disk varies per device. Table 5.2 lists the amount of space available on the common disk partitions. For example, on an RA60 drive there are two partitions with available space; partition g has 41 megabytes available, and partition h has 139 megabytes available. In the table, ? refers to the unit number.

Table 5.2 Space Available on Common Disk Partitions

Drive	Partition	Size	Partition	Size
RA60	ra?g	41 Mb	ra?h	139 Mb
RA80	ra?g	41 Mb	ra?h	56 Mb
RA81	ra?g	41 Mb	ra?h	380 Mb
R80	rb?g	41 Mb	rb?h	550 Mb
RB80	rb?g	41 Mb	rb?h	56 Mb
RK07	hk?g	13 Mb		
RL02	rl?g	10 Mb		
RL02 (IDC)	rb?g	10 Mb		
RM03	hp?g	41 Mb		
RM05	hp?g	80 Mb	hp?h	145 Mb
RM80	hp?g	96 Mb		
RP05	hp?g	160 Mb		
RP06	hp?g	145 Mb		
RP07	hp?g	315 Mb	hp?h	145 Mb

Each disk also has a paging area, typically of 16 megabytes, and a root file system of 8 megabytes. The distributed system binaries occupy about 22 megabytes, and the major sources occupy another 25 megabytes. The binaries and sources overflow dual RK07 and dual RL02 systems, but fit easily on most other hardware configurations.

Disk sizes are measured in disk sectors (512 bytes), but the ULTRIX-32 file system blocks vary in size. All user programs report disk space in kilobytes and, where needed, specify disk sizes in terms of sectors. The `/etc/disktab` file used in making file systems specifies disk partition sizes in sectors. To override the default sector size of 512 bytes, use the `se` attribute.

5.1.3. Layout Considerations

You should consider these points when deciding how to arrange your disks:

- provide adequate space for your requirements
- maximize throughput
- consider the paging area carefully

As distributed, the system sizes the configured paging areas each time the system is booted. You can define multiple paging areas of different sizes so they will be interleaved. Drives smaller than 400 megabytes have 16 megabyte swap partitions; drives larger than 400 megabytes have 32 megabyte swap partitions.

Many common system programs (C, the editor, the assembler and so forth) create intermediate files in the `/tmp` directory. Therefore, you must make the file system where `/tmp` will be stored large enough to accommodate these common programs. If you have several disks, it is wise to mount `/tmp` in a root file system (partition a) on another disk. All programs that create files in `/tmp` usually delete them. However,

116 ULTRIX-32 Installation Guide

you should periodically examine /tmp and delete any old files that /tmp may have missed.

The disk controller configuration affects the efficiency with which the ULTRIX-32 system is able to use the CPU. For general time-sharing applications, try to split the root file system (/), system binaries (/usr), the temporary files (/tmp), and the user files among several disk arms, and interleave the paging activity among several disk arms.

For good performance, you must balance the disk load. There are at least five components of the disk load that you can divide among the available disks:

- root file system
- /tmp file system
- /usr file system
- user files
- paging activity

Table 5.3 shows some disk load possibilities for systems with two, three, and four disks. The numerical values in the table are the partitions. Where there is an *x*, you chose the partition. For example, on a system with two or three disks, you can load the archive files on any partition you chose. But on a system with four disks, you should load the archive files in the fourth partition.

Table 5.3 Disk Load Possibilities

file	two	three	four
system	disks	disks	disks
/	1	2	2
tmp	1	3	4
usr	1	1	1
paging	1+2	1+3	1+3+4
users	2	2+3	2+3
archive	<i>x</i>	<i>x</i>	4

The most important balance consideration is to even out the disk load as much as possible. You can even the load by decoupling file systems between which heavy copying occurs. It is much more important to have an instantaneously balanced load when the system is busy, than to have a long term average balanced load.

By experimenting with a few file system arrangements, you can greatly improve the system's performance. It is particularly easy to move the root, the /tmp file system and the paging areas. Place the user files and the /usr directory as space needs dictate and experiment with the other, more easily moved file systems.

5.1.4. File System Parameters

Each file system has parameters according to its block size, fragment size, and the disk geometry characteristics of the medium on which it resides. If you specify the wrong disk characteristics or choose inadequate file system parameters, the throughput (i/o seek time) may be substantially degraded or a significant amount of disk space may be wasted. As distributed, the file systems are configured as shown in Table 5.4.

Table 5.4 ULTRIX-32 File System Configurations

File System	Block Size	Fragment Size
/	8 Kbytes	1 Kbytes
user	4 Kbytes	512 bytes
users	4 Kbytes	1 kbytes

The root file system block size is large (eight Kbytes) so it will optimize the bandwidth to the associated disk. This is very important because the /tmp directory is normally part of the root file. The large block size is also important because many of the most heavily used programs are demand paged out of the /bin directory.

The /usr file system uses a four Kbyte block size with 512 byte fragment size in an effort to get high performance while conserving the amount of space wasted by a large fragment size.

The file systems for users have a four Kbyte block size with one Kbyte fragment size. The four Kbyte block size provides adequate bandwidth while the one Kbyte fragment size provides acceptable space compaction and disk fragmentation.

You may chose other parameters in constructing file systems, but keep in mind that there are many factors involved which interact in complex ways in choosing a block size and fragment size.

Larger block sizes result in better throughput to large files in the file system because larger i/o requests will then be performed by the system. However, you must also consider the average file sizes in the file system and the performance of the internal system buffer cache.

The system provides space in the inode for 12 direct block pointers, 1 single indirect block pointer, and 1 double indirect block pointer. If a file uses only direct blocks, you can optimize access time to it by maximizing the block size. If a file spills over into an indirect block, increasing the block size of the file system may decrease the amount of space used by eliminating the need to allocate an indirect block. However, if the block size is increased and an indirect block is still required, then more disk space will be used by the file because indirect blocks are allocated according to the block size of the file system.

You should consider two main points when you are selecting a fragment size for a file system. The first consideration is the major performance tradeoffs between an eight

118 ULTRIX-32 Installation Guide

Kbyte block file system and a four Kbyte block file system. The block size to fragment size ratio can not be greater than eight. This means that an eight Kbyte file system will always have a fragment size of at least one Kbytes. If a file system is created with a four Kbyte block size and a one Kbyte fragment size, then upgraded to an eight Kbyte block size and one Kbyte fragment size, identical space compaction occurs.

However, if a file system has a 4 Kbyte block size and 512 byte fragment size, converting it to an 8K/1K file system results in significantly more space being used. This implies that four Kbyte block file systems which might be upgraded to eight Kbyte blocks for higher performance should use fragment sizes of at least one Kbytes to minimize the amount of work required in conversion.

The second consideration when selecting the fragment size for a file system is the level of fragmentation on the disk. With a 512 byte fragment size, storage fragmentation occurs much sooner, particularly with a busy file system running near full capacity. By comparison, the level of fragmentation in a one Kbyte fragment file system is an order of magnitude less severe. On file systems where many files are created and deleted, the 512 byte fragment size is more likely to result in apparent space exhaustion due to fragmentation. That is, when the file system is nearly full, file expansion which requires locating a contiguous area of disk space is more likely to fail on a 512 byte file system than on a 1 Kbyte file system.

To minimize fragmentation problems of this sort, a parameter in the super block specifies a minimum acceptable free space threshold. If users (anyone but the superuser) attempt to allocate disk space and the free space threshold is exceeded, the user is returned an error as if the file system were actually full. By default, this parameter is set to 10%; you may change it by supplying a parameter to `newfs(8)`, or by patching the super block of an existing file system using `tunefs(8)`.

In general, unless a file system is to be used for a special purpose application (for example, storing image processing data), you should use the default values supplied. Remember that the current implementation limits the block size to at most eight Kbytes and the ratio of block size to fragment size must be in the range 1-8.

The disk geometry information which the file system uses affects the block layout methods used. The file `/etc/disktab` contains the data for most of the drives supported by the system (although not all of these drives are supported by Digital Equipment Corporation). When constructing a file system, use the `newfs(8)` program and specify the type of disk on which the file system resides. `/etc/disktab` also contains the default file system partition sizes, and default block and fragment sizes. You can override any of the default values by modifying `/etc/disktab`, or by using one of the `newfs` options.

5.1.5. Implement the Layout

To implement your disk layout, use the `newfs(8)` command to create each new file system. You must also add each file system to the `/etc/fstab` file so that it will be checked and mounted when the system is bootstrapped.

For example, consider a system with RM03s. On the first RM03 (hp0) you would put the root file system in partition hp0a, and the `/usr` file system in hp0g, which has more than enough space to hold it. The `/tmp` directory would be part of the root file system, because no file system will be mounted on it.

If there were only one RM03 in this example, then you would put user files in the hp0g partition along with the binaries.

If you had a second RM03 (hp1), you would create a file system in hp1g and put user files there, calling the file system `/mnt`. You would also need to interleave the paging between the two RM03 drives. To do this you would build a system configuration that specified the following:

```
config    vmunix    root on hp0 swap on hp0 and hp 1
```

You would then add the following lines to the `/etc/fstab` file:

```
/dev/hp1b::sw::
/dev/hp1g:/mnt:rw:1:2
```

You should keep a backup copy of the root file system in the **hp1a** disk partition.

To make the `/mnt` file system you would type:

```
# cd /dev
# MAKEDEV hp 1
# newfs hp1g rm03
.
.
.
information about the file system prints out
.
.
.
# mkdir /mnt
# mount /dev/hp1g /mnt
```

5.2. Set up the Line Printer System

If you have a line printer configured in to your system (that is, if there is an entry describing a line printer in the system's configuration file, and a device type in the `/dev` directory), the line printer facility will work as distributed.

120 ULTRIX-32 Installation Guide

The basic line printer system consists of these files and commands:

/usr/ucb/lpq	spooling queue examination program
/usr/ucb/lprm	delete jobs from a queue program
/usr/ucb/lpr	enter a job in a printer queue program
/etc/printcap	printer configuration and capability data base
/usr/lib/lpd	line printer daemon, scans spooling queues
/etc/lpc	line printer control program

The file `/etc/printcap` is a master data base describing line printers directly attached to a machine, and printers accessible across a network. `printcap(5)` describes the format of this data base and also indicates the default values for such things as the directory in which spooling is performed.

The line printer system handles multiple printers, multiple spooling queues, local and remote printers, and also printers attached via serial lines which require line initialization such as the baud rate.

Remote spooling via the network is handled with two spooling queues: one on the local machine, and one on the remote machine. When a remote printer job is initiated with `lpr(,)` the job is queued locally and a daemon process is created to oversee the transfer of the job to the remote machine. If the destination machine is unreachable, the job will remain queued until it is possible to transfer the files to the spooling queue on the remote machine. The `lpq()` program shows the contents of spool queues on both the local and remote machines.

To configure your line printers, consult `printcap(5)` and *4.2BSD Line Printer Spooler Manual*. A call to the `lpd(8)` program should be present in `/etc/rc`.

5.3. Set up the Mail System

The mail facility distributed with the ULTRIX-32 system will work properly as distributed.

The mail system consists of these commands:

/bin/mail	old standard mail program (from 32/V)
/usr/ucb/mail	UCB mail program, described in <i>mail(1)</i>
/usr/lib/sendmail	mail routing program
/usr/spool/mail	mail spooling directory
/usr/spool/secretmail	secure mail directory
/usr/bin/xsend	secure mail sender
/usr/bin/xget	secure mail receiver
/usr/lib/aliases	mail forwarding information
/usr/ucb/newaliases	command to rebuild binary forwarding database
/usr/ucb/biff	mail notification enabler

<code>/etc/comsat</code>	mail notification daemon
<code>/etc/syslog</code>	error message logger, used by sendmail

Mail is usually sent and received via the `mail(1)` command, which provides a front-end to edit the messages sent and received, and passes the messages to `sendmail(8)` for routing.

The routing algorithm uses knowledge of the network name syntax, aliasing and forwarding information, and network topology, as defined in the configuration file `/usr/lib/sendmail.cf`, to process each piece of mail.

Local mail is delivered through the `/usr/bin/mail` program, which adds the mail to the appropriate `/usr/spool/mail/username` mailbox. `/usr/bin/mail` uses a locking protocol to avoid problems with simultaneous updates. After the mail is delivered, the local mail delivery daemon `/etc/comsat` is notified, which in turn notifies users who have issued the `biff(1)` command that mail has arrived.

Mail queued in the directory `/usr/spool/mail` is readable only by the recipient. To send encrypted mail that only the recipient can read use the secret mail facility `xsend(1)`.

To set up the mail facility, read the instructions in the file `README` in the directory `/usr/src/usr.lib/sendmail` and then edit the necessary configuration files. Also, set up the file `/usr/lib/aliases` for your installation, and create mail groups as appropriate. Consult the document *Sendmail Installation and Operation Guide* (also included in this binder) and `sendmail(8)` for further information.

5.4. Set up a UUCP Connection

These are the basic steps for installing a `uucp` (UNIX to UNIX Copy) connection:

- Choose the hardware and install it.
- Set up the special files corresponding to that hardware.
- Install the `uucp` software and related files.

Read the document *UUCP Installation and Administration* for detailed instructions on how to set up a `uucp` connection.

5.5. Additional Software

This subsection describes how to extract the additional software from the second distribution tape and the third RK07 distribution disk pack.

If your distribution is on an RA60 pack or on RL02 packs then your system already has the additional software installed. If you find your system requires more disk space, you can decide which additional software files your users do not need and remove them without disturbing your system's performance.

122 ULTRIX-32 Installation Guide

The additional software is made up of six optional packages. Table 5.5 gives the name of each package in the order it appears on the distribution media (tape or RK07 disk), the approximate disk usage of each package in blocks, and a brief description of each package.

File	Disk Usage	Description
/usr/man	2400	on-line manuals, sections 1-8 of the <i>ULTRIX-32 Programmer's Manual</i>
/usr/games	2200	game programs
/usr/doc	4500	sources to UNIX documentation (minus sections 1-8)
/usr/lib/vfont	2600	font libraries for Varian or Versatec printer/plotters
/usr/lib/learn	900	learn scripts for learning how to use the ULTRIX-32 operating system
/usr/lib/lisp	1200	lisp documentation

Table 5.5 can help you decide whether you have room for some, all, or none of the additional software.

Note that the doc and vfont disk usage is given in the compacted form. If you want to use either of these files you should read the README file in the /usr/doc or /usr/lib/vfont directory.

None of the Additional Software files are necessary for normal operation of your ULTRIX-32 system, but they are useful because they contain on-line learning tools.

The next two subsections explain how to extract the Additional Software from the distribution tape and Rk07 pack.

NOTE: Everytime you execute the `newfs(8)` command, save the numbers generated. A good place to keep them is taped to the outside of the disk cover (if the pack is removable).

5.5.1. Extracting Additional Software from the Second Tape

If you have room on the /usr partition of your system disk, you can extract all or some of the files from the second tape and place them in /usr. Otherwise, you can place the files in a partition of another disk, and make symbolic links. See `df(1)` for more information about disk space usage.

5.5.1.1. If /usr of System Disk is Available

If you have enough room in the /usr partition of your system disk and want to extract all of the tape files, enter the following commands in the order given.

NOTE: If you are putting the file(s) on a partition that has not previously been used, you must do a `newfs(8)` first, and then edit the file `/etc/fstab`, adding an entry for the partition. (See subsection 3.1 of this document or *Guidelines for System Management* for information on how to edit `/etc/fstab`.) For example, if you were placing the file(s) on partition `g` of an RM03 disk in drive 1, type:

```
# newfs rhp1g rm03
```

and add the following line to `/etc/fstab`:

```
/dev/hp1g:/usr:rw:1:2
```

CAUTION: `newfs` destroys the data on the partition specified; therefore, use caution when using `newfs`.

If you want to skip some of the files, you must position the tape at the desired tape file before entering the commands to extract the file. To do this, do not issue the `tar` command in the sequence below, for each tape file you want to skip.

Each `tar` command (file extraction) will take about five minutes to complete.

```
# cd /usr                                (make /usr the current directory)
# tar xpf /dev/rmt12                       (extract on-line manuals)
# mt fsf

# tar xpf /dev/rmt12                       (extract user games)
# mt fsf

# tar xpf /dev/rmt12                       (extract UNIX documentation)
# mt fsf

# cd /usr/lib                              (make /usr/lib the current directory)
# tar xpf /dev/rmt12                       (extract vfont libraries)
# mt fsf

# tar xpf /dev/rmt12                       (extract learn scripts)
# mt fsf

# tar xpf /dev/rmt12                       (extract lisp manual)
```

124 ULTRIX-32 Installation Guide

When you have finished extracting files from the tape, enter:

```
# mt off ; sync
```

Then unload the tape from the tape drive, and store it in a safe place.

You have now completed the ULTRIX-32 operating system installation.

5.5.1.2. If /usr of System Disk is Not Available

If you do not have room in /usr of the system disk for the files on the second tape, then you need to make a site-specific decision about which files your users will need, and where those files will be placed. (See `df(1)` for more information about disk space usage.)

You can place the tape files in a partition of another disk, and then make a symbolic link from the directory where they would normally reside to where they actually reside.

Here is an example of how to extract the games tape file onto a system consisting of two RK07 drives. Assume there is no room in the `g` partition of the system disk (ROOT), but there is room in the `g` partition of the second RK07 (USERS). Mount the `g` partition onto the /usr/users directory. Then extract the games file in /usr/users/games. Last, make a symbolic link from /usr to /usr/users. Name the link games.

```
# /etc/mount /dev/hk1g /usr/users
# cd /usr/users
# mt fsf                               (to skip man - see below)
# tar xpf /dev/rmt12                   (extract the games)
# mt fsf
# ln -s /usr/games /usr/users/games
```

Note that in this situation you will be giving up user space by placing the additional software information in /usr/users. Therefore it is important that you only extract the additional software files that your users really need.

To skip some of the files you must position the tape at the desired tape file before entering the commands to extract the file.

Repeat the `tar`, `chmod`, `mt`, and `ln` commands, substituting the appropriate file name for each file you wish to extract from the tape.

Be sure you are in the correct current directory. In our example to extract /usr/games, /usr/users should be the current directory.

When you have finished extracting files from the tape, enter:

```
# mt offl ; sync
```

Then unload the tape from the tape drive, and store it in a safe place.

Check the `/etc/fstab` file to be sure there is an an entry for the partition. The entry ensures that the file system will be checked automatically and mounted on reboot.

Use the `cat(1)` command to see if the entry exists; if it does not, then edit the file and add it.

For example, in our case with the RK07 disks, first use `cat`:

```
/dev/hk0a::rw:1:1
/dev/hk0g:/usr:rw:1:3
/dev/hk1a:/mnt:rw:1:2
```

In this example there is no entry for `/dev/hk1g`, therefore you should edit the `/etc/fstab` file, adding a line like this:

```
/dev/hk1g:/dev:rw:1:3
```

You have now completed the ULTRIX-32 operating system installation.

5.5.2. Extracting Additional Software from the Third RK07

The system must be in single-user mode; type:

```
# /etc/shutdown -h
```

Spin down the ROOT pack in drive 0 and load the pack labeled ADDITIONAL SOFTWARE. When the ADDITIONAL SOFTWARE pack (in drive 0) is spinning and the READY light is lit, reboot the system to single-user mode (subsection 2.5.1).

After the system is rebooted type:

```
# /etc/mount /dev/hk1g /usr
```

The `/dev/hk1g` partition is normally mounted on `/usr/users`. In this particular instance, however, we are mounting it on `/usr` for convenience.

Repeat the following two commands, substituting the name of the file you want to extract for *file*, until you have extracted all the files your users will need (and that will

126 ULTRIX-32 Installation Guide

fit on your particular system):

```
# cd /usr
# tar xpf /dev/rhk0g file
```

After you have extracted all the files needed, type:

```
# sync
# halt
```

Spin down and unload the ADDITIONAL SOFTWARE pack in drive 0 and load the ROOT pack. When drive 0 is spinning and the READY light is lit, reboot the system to single-user mode (subsection 2.5.2).

After the system has been rebooted to single-user mode, check the filesystem and then mount all the devices:

```
# fsck
# /etc/mount /dev/hk1g
```

Then set up a symbolic link for each additional software file extracted. For example, to set up a symbolic link for games:

```
# ln -s /usr/users/games /usr/games
```

After all the symbolic links have been made bring the system up to multi-user mode:

```
# <CTRL/D>
```

You have now completed the ULTRIX-32 operating system installation.

APPENDIX A – Bootstrap Details

This appendix contains pertinent files and numbers for the ULTRIX-32 bootstrap procedure. If you have problems installing the distribution on your machine, you may find this material useful.

1. Contents of the Distribution Tapes

The tape distribution consists of two 1600bpi 2400' magnetic tapes. All tape files are blocked in 10 kilobytes records, except for the first file on the first tape which has 512 byte records.

The first distribution tape contains the following files. Note that the number of records given are approximate.

<u>Tape File</u>	<u>Records</u>	<u>Contents</u>
one	194	eight tape bootstraps and a <i>tp</i> (1) image containing the stand-alone-images of <i>boot</i> , <i>format</i> , and <i>copy</i>
two	300	“mini-root” file system image
three	380	<i>dump</i> (8) image of distribution root file system
four	440	<i>tar</i> (1) image of kernel objects
five	2111	<i>tar</i> image of <i>usr</i> filesystem binaries and libraries in <i>/usr</i>

The second distribution tape contains the following files. Note that the number of records and blocks given are approximate.

<u>Tape File</u>	<u>Records</u>	<u>Blocks</u>	<u>Contents</u>
one	243	2257	<i>tar</i> image of <i>/usr/man</i>
two	214	3409	<i>tar</i> image of <i>/usr/games</i>
three	523	5198	<i>tar</i> image of <i>/usr/doc</i>
four	576	5677	<i>tar</i> image of <i>/usr/lib/vfont</i>
five	92	795	<i>tar</i> image of <i>/usr/lib/learn</i>
six	48	483	<i>tar</i> image of <i>/usr/lib/lisp</i>

The distribution tapes were made with the shell scripts located in the directory */sys/dist*.

128 ULTRIX-32 Installation Guide

2. Control Status Register Addresses

The distribution uses many stand-alone device drivers that presume the location of a UNIBUS device's control status register (CSR). Table A-1 summarizes these values.

Table A-1

Device Name	Controller	CSR Address (octal)
ra	UDA50	0172150
rb	730 IDC	0175606
rk	RK11	0177440
rl	RL11	0174400
ts	TS11	0172520

All MASSBUS controllers are located at standard offsets from the base address of the MASSBUS adapter register bank.

3. Generic System Control Status Register Addresses

The *generic* version of the operating system supplied with the distribution contains the UNIBUS devices indicated below. The system recognizes these devices if the appropriate control status registers respond at any of the indicated UNIBUS addresses.

Device Name	Controller	CSR Addresses (octal)
hk	RK11	0177440
ra	UDA-50	0172150, 0172550, 0177550
rb	730 IDC	0175606
rl	RL11	0174400
dn	DN11	0160020
dz	DZ11	0160100, 0160110, ... 0160170
ts	TS11	0172520
dmf	DMF32	0160340
lp	LP11	0177514

If devices other than the above are located at any of the addresses indicated, the system may not bootstrap properly.

APPENDIX B – Installation Troubleshooting

This appendix lists and explains problems you might encounter while installing the ULTRIX-32 software distribution. This information is limited to the early steps in the installation process; that is, up to the point where the root file system is installed. If you have a problem installing the release consult this section for an indication of the problem before contacting your Digital representative.

1. Using the Distribution Console Medium

This section describes problems which may occur when using the programs provided on the distributed console medium: TU58 cassette or RX01 diskette.

boot prints "bad format"

The program you requested *boot* to load did not have a 407, 410, or 413 magic number in its header. This should never happen on a distribution system. If you were trying to boot off the root file system, reboot the system on the miniroot file system and look at the program on the root file system. Try copying the copy of *vmunix* on the miniroot to the root file system also.

boot prints "not a directory"

The boot program was unable to find the requested program because it encountered something other than a directory while searching the file system. This usually indicates no file system is present on the disk partition supplied, or the file system has been corrupted. First check to make sure you typed the correct line to boot. If this is the case and you are booting off the miniroot file system, the miniroot was probably not copied correctly off the tape (perhaps it was not placed in the correct disk partition). Try reinstalling the miniroot file system or, if trying to boot the true root file system, try booting off the miniroot file system and run *fsck* on the restored root file system to insure its integrity. Finally, as a last resort, copy the boot program from the miniroot file system to the newly installed root file system.

boot prints "read short"

The file header for the program indicated a size larger than the actual size of the file located on disk. This is probably the result of file system corruption (or a disk i/o error). Try booting again or creating a new copy of the program to be loaded (see above).

copy does not copy 300 records

If a tape read error occurred, clean your tape drive heads. If a disk write error occurred, the disk formatting may have failed. If the disk pack is removable, try another one. If you are currently running UNIX, you can reboot your old system and use *dd* to copy the miniroot file system into a disk partition (assuming the destination is not in use by the running system).

130 ULTRIX-32 Installation Guide

format, boot, or copy prints "unknown device".

The device specified is probably not one of those supported by the distribution; consult subsection 1.1. If the device is listed in subsection 1.1, the drive may be dual-ported, or for some other reason the driver was unable to decipher its characteristics. If this is a MASSBUS drive, try powering the MASSBUS adapter and/or controller on and off to clear the drive type register.

program can not be loaded.

Check to make sure the correct floppy or cassette is being used. If using a floppy, be sure it is not in upside down. If using a cassette on an 11/730, be certain drive 0 is being used. If a hard i/o error occurred while reading a floppy, try resetting the console LSI-11 by powering it on and off. If you can not boot the cassette's bootstrap monitor, verify the standard DEC console cassette can be read; if it can not, your cassette is broken – not uncommon.

program halts without warning.

Check to make sure you have specified the correct disk to format. On 11/750's, specifying a non-existent MASSBUS device will cause the program to halt as it receives an interrupt (standalone programs operate by polling devices).

If using a floppy, try reading the floppy under your current system. If this works, copy the floppy to a new one and begin again. If using a cassette on an 11/730, do likewise.

A MASSBUS disk was specified, but the associated MASSBUS drive type register indicates a drive of unknown type. This probably means you typed something wrong or your hardware is incorrectly configured.

WARNING: clock gained xx days -- CHECK AND RESET THE DATE!

WARNING: preposterous time in file system -- CHECK AND RESET THE DATE!

If you receive either one of these warnings, ignore it; you will set the date later when you boot the full root.

2. Booting the Generic System

This section contains common problems encountered when booting the generic version of the system.

system crashes during autoconfiguration

This is almost always caused by an unsupported UNIBUS device being present at a location where a supported device was expected. You must disable the device in some way, either by pulling it off the bus, or by moving the location of the console status register (consult Appendix A for a complete list of UNIBUS csr's used in the generic system).

system does not find device(s)

Either the device was at a non-standard address, or the device is not configured into the generic system. See Appendix A for a list of devices and their control status

register addresses.

Alternatively, certain devices are difficult to locate during autoconfiguration. A classic example is the TS11 tape drive which does not autoconfigure properly if it is rewinding when the system is rebooted. Tape drives should configure properly if they are off line, or are not performing a tape movement. Disks which are dual-ported should autoconfigure properly if the drive is not being simultaneously accessed through the alternate port.

system panics with "panic: iinit"

This occurred because the system was unable to locate the program /etc/init. The root file system supplied at the root device? prompt was probably incorrect. Remember that when running on the miniroot file system, this question must be answered with something of the form hp0*. If the answer had been hp0, the system would have used the a partition on unit 0 of the hp drive, where presumably no file system exists.

Alternatively, the file system on which you were trying to run is corrupted, or simply missing /etc/init. Try reinstalling the appropriate file system or installing a version of init.

system selects incorrect root device

That is, you try to boot the system single user with B/2 or B xxS but do not get the root filesystem in the expected location. This is most likely caused by your having many disks available more suited to be a root file system than the one you wanted. For example, if you have a up disk and an hk disk and install the system on the hk, then try and boot the system to single-user mode, the heuristic used by the generic system to select the root file system will choose the up disk. The following list gives, in descending order, those disks thought most suitable to be a root file system: hp, up, ra, rb, rl, hk (the position of rl is subject to argument). To get the root device you want you must boot using B/3 or B ANY, then supply the root device at the prompt.

3. Building Console Cassettes

This sections describes common problems encountered while constructing a console bootstrap cassette.

system crashes

You are trying to build a cassette for an 11/750. On an 11/750 the system is booted by using a bootstrap prom and sector 0 of the root file system. Consult tu((4)) for the appropriate reprimand.

APPENDIX C – Loading the Tape Monitor

This appendix indicates how to load the bootstrap monitor located on the first tape of the distribution tape. Loading the monitor should not be necessary, but instructions have been included as a fallback measure (in case it is not possible to read the distributed console medium.)

NOTE: In certain cases on an 11/730 system, the bootstraps supplied below may not work. Consult your Digital representative if you are unable to load the monitor on an 11/730 system.

To load the tape bootstrap monitor, first mount the magnetic tape on drive 0 at load point, making sure that the write ring is not inserted. Set the reboot switch to OFF on an 11/780 or 11/730. On an 11/750 set the power-on action switch to halt. (Generally an 11/780 or 11/730 system will have the reboot switch on, and an 11/750 system will have the power-on action set to boot/restart.)

If you have an 11/780 system enter:

```
>>> HALT
>>> UNJAM
```

Then, on any machine, enter the init(8) command:

```
>>> I
```

Then, key in at location 200 and execute either the TS, HT, TM, or MT bootstrap described in Table B.1, as appropriate. The machine's printouts are shown in regular print, explanatory comments are within (). (You can use <DELETE> to delete a character and <CTRL/U> to remove an entire line.)

Table B.1 Bootstrap Procedures

TS bootstrap

```
>>> D/P 200 3AEFD0
>>> D + D05A0000
>>> D + 3BEF
>>> D + 800CA00
>>> D + 32EFC1
>>> D + CA010000
>>> D + EFC10804
>>> D + 24
>>> D + 15508F
>>> D + ABB45B00
```

134 ULTRIX-32 Installation Guide

```
>>> D + 2AB9502
>>> D + 8FB0FB18
>>> D + 956B024C
>>> D + FB1802AB
>>> D + 25C8FB0
>>> D + 6B
```

(The next two deposits set up the addresses of the UNIBUS adapter and its memory; the 20006000 here is the address of the 11/780 uba0 and the 2013E000 the address of the 11/780 umem0)

```
>>> D + 20006000      (780 uba0)
      (780 uba1: 20008000, 750 uba: F30000, 730 uba: F26000)
>>> D + 2013E000      (780 umem0)
      (780 umem1: 2017E000, 750 umem: FFE000, 730 umem: FFE000)
>>> D + 80000000
>>> D + 254C004
>>> D + 80000
>>> D + 264
>>> D + E
>>> D + C001
>>> D + 2000000
>>> S 200
```

HT bootstrap

```
>>> D/P 200 3EEFD0
>>> D + C55A0000
>>> D + 3BEF
>>> D + 808F00
>>> D + C15B0000
>>> D + C05B5A5B
>>> D + 4008F
>>> D + D05B00
>>> D + 9D004AA
>>> D + C08F326B
>>> D + D424AB14
>>> D + 8FD00CAA
>>> D + 80000000
>>> D + 320800CA
>>> D + AAFE008F
>>> D + 6B39D010
>>> D + 0
```

(The next two deposits set up the addresses of the MASSBUS adapter and the drive number for the tape)

formatter. The 20012000 here is the address of
the 11/780 mba1 and the 0 reflects that the
formatter is drive 0 on mba1)

```
>>> D + 20012000 (780 mba1) (780 mba0: 20010000, 750 mba0: F28000)
>>> D + 0 (Formatter unit number in range 0-7)
>>> S 200
>>> S 200
```

TM bootstrap

```
>>> D/P 200 2AEFD0
>>> D + D0510000
>>> D + 2000008F
>>> D + 800C180
>>> D + 804C1D4
>>> D + 1AEFD0
>>> D + C8520000
>>> D + F5508F
>>> D + 8FAE5200
>>> D + 4A20200
>>> D + B006A2B4
>>> D + 2A203
```

(The following two numbers are uba0 and umem0; see
TS above for some hints on other values if your
TM is not on UBA0 on a 780)

```
>>> D + 20006000 (780 uba0)
>>> D + 2013E000 (780 umem0)
>>> S 200
>>> S 200
>>> S 200
```

MT bootstrap

```
>>> D/P 200 46EFD0
>>> D + C55A0000
>>> D + 43EF
>>> D + 808F00
>>> D + C15B0000
>>> D + C05B5A5B
>>> D + 4008F
>>> D + 19A5B00
>>> D + 49A04AA
>>> D + AAD408AB
>>> D + 8FD00C
```

136 ULTRIX-32 Installation Guide

```
>>>D + CA800000
>>>D + 8F320800
>>>D + 10AAFE00
>>>D + 2008F3C
>>>D + ABD014AB
>>>D + FE15044
>>>D + 399AF850
>>>D + 6B
```

(The next two deposits set up the addresses of the MASSBUS adapter and the drive number for the tape formatter. The 20012000 here is the address of the 11/780 mba1 and the 0 reflects that the formatter is drive 0 on mba1)

```
>>>D + 20012000
>>>D + 0
>>>S 200
>>>S 200
>>>S 200
>>>S 200
```

(no toggle-in code exists for the UT device)

If the tape does not move the first time you start the bootstrap program with S 200 you probably entered the program incorrectly (but also check that the tape is online). Start the procedure over again. For the HT, MT, and TM bootstraps you cannot see the tape move as you advance through the first few blocks as the tape motion is all within the vacuum columns.

Next, deposit the address of the tape MBA/UBA in RA. In RB, deposit the appropriate address of the device registers or unit number.

```
>>>D/G A 20006000 (for tapes on 780 uba0)
>>>D/G A 20008000 (for tapes on 780 uba1)
>>>D/G A 20012000 (for tapes on 780 mba1)
>>>D/G A 20010000 (for tapes on 780 mba0)
>>>D/G A F30000 (for tapes on 750 uba0)
>>>D/G A F2A000 (for tapes on 750 mba1)
>>>D/G A F28000 (for tapes on 750 mba0)
>>>D/G A F26000 (for tapes on 730 uba0)
```

For register B:

```
>>> D/G B 0          (for tm03/tm78 formatters at mba? drive 0)
>>> D/G B 1          (for tm03/tm78 formatters at mba? drive 1)
>>> D/G B 2013F550   (for tm11/ts11/tu80 tapes on 780 uba0)
>>> D/G B FFF550     (for tm11/ts11/tu80 tapes on 750 or 730 uba0)
```

Then start the bootstrap program with

```
>>> S 0
```

The console should display

```
=
```

You are now talking to the tape bootstrap monitor. At any point during the rest of the installation you can return to this section, reload the tape bootstrap, and restart the procedure. The console monitor is identical to that loaded from a TU58 console cassette. Follow the instructions in section 2 as they apply to this device. The only exception is that when using programs loaded from the tape bootstrap monitor, programs will always return to the monitor (the = prompt). This saves you typing in the toggle-in code for each program to be loaded.

For the TU58 monitor, the programs restart themselves. You must therefore restart the monitor to get the console prompt.

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, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575

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-----

digital



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-32™ Documentation Group
MK02-1/H10
Continental Blvd.
Merrimack, N.H.
03054

-----Do Not Tear - Fold Here and Tape-----

Cut Along Dotted Line

Notes:

Notes:

digital™