ULTRIX-32
Guide to
System Environment Setup

Order No. AA-ME89A-TE

ULTRIX-32 Operating System, Version 3.0

Digital Equipment Corporation

Copyright © 1987, 1988 Digital Equipment Corporation
All Rights Reserved.

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

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

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

The following are trademarks of Digital Equipment Corporation:

DEC	Q-bus	VAX
DECnet	RT	VAXstation
DECUS	ULTRIX	VMS
MASSBUS	ULTRIX-11	VT
MicroVAX	ULTRIX-32	ULTRIX Worksystem Software
PDP	UNIBUS	digital

UNIX is a registered trademark of AT&T in the USA and other countries.

IBM is a registered trademark of International Business Machines Corporation.

MICOM is a registered trademark of Micom System, Inc.

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

Contents

About This Manual

1 Modifying System Files

4 Setting Up the Print System

5 Adding and Deleting Software Subsets

6 Monitoring and Managing System Performance

A Device Mnemonics

Figures

Examples

Tables

About This Manual

The objective of this guide is to provide you with information needed to set up and maintain a working environment for users at your site. The guide assists you as you set up your particular environment and presents guidelines from which you can develop specific procedures for your site.

Audience

The Guide to System Environment Setup is written for the person responsible for managing and maintaining an ULTRIX system. It assumes that this individual is familiar with ULTRIX commands, the system configuration, the system's controller/drive unit number assignments and naming conventions, and an editor such as vi(1) or ed(1). You do not need to be a programmer to use this guide.

Organization

This manual consists of 5 chapters, one appendix, and an index. The chapters and the appendix are:

Chapter 1: Modifying System Files
Identifies and describes the most commonly accessed ULTRIX system files and defines the file formats. The chapter also identifies system utilities that you can use to modify the files.

Chapter 2: Adding and Deleting Users
Identifies and explains how to use the various ULTRIX system utilities to add users.

Chapter 3: Adding Devices
Describes how to add devices using the MAKEDEV script.

Chapter 4: Setting Up the Print System
Explains how to set up the print system manually or, if you prefer, interactively with the lprsetup utility.

Chapter 5: Adding Software Subsets
Explains how to use the setld utility to add software subsets.

Chapter 6: Monitoring System Performance
Offers guidelines for monitoring system performance.

Appendix A Device Mnemonics
Lists the supported device mnemonics and explains how to obtain detailed reference page information on devices.

Related Documents

You should have the hardware documentation for your system and peripherals.

Conventions

The following conventions are used in this manual:

special	In text, each mention of a specific command, option, partition, pathname, directory, or file is presented in this type.
command(x)	In text, cross-references to the command documentation include the section number in the reference manual where the commands are documented. For example: See the cat(1) command. This indicates that you can find the material on the cat command in Section 1 of the reference pages.
literal	In syntax descriptions, this type indicates terms that are constant and must be typed just as they are presented.
italics	In syntax descriptions, this type indicates terms that are variable.
[ ]	In syntax descriptions, square brackets indicate terms that are optional.
. . .	In syntax descriptions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.

function	In function definitions, the function itself is shown in this type. The function arguments are shown in italics.
UPPERCASE	The ULTRIX system differentiates between lowercase and uppercase characters. Enter uppercase characters only where specifically indicated by an example or a syntax line.
example	In examples, computer output text is printed in this type.
example	In examples, user input is printed in this bold type.
%	This is the default user prompt in multiuser mode.
#	This is the default superuser prompt.
>>>	This is the console subsystem prompt.
.	In examples, a vertical ellipsis indicates that not all of the
.	lines of the example are shown.
.	
<KEYNAME>	In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal keyboard.
<CTRL/x>	In examples, symbols like this indicate that you must hold down the CTRL key while you type the key that follows the slash. Use of this combination of keys may appear on your terminal screen as the letter preceded by the circumflex character. In some instances, it may not appear at all.

Modifying System Files 1

1.1 Preparation

Before you install Version 3.0, you should back up your system, especially any site-specific files. Refer to the Advanced Installation Guide for more information on site-specific files. After installing the system, you should check the site-specific files against the files shipped with Version 3.0. Before restoring these files, be sure that all Version 3.0 specific information is in the files you restore. For example, previous to V1.2 the loopback driver was specified in the /etc/hosts and the /etc/networks files. From V1.2 on, the loopback driver must be specified in the /etc/rc.local file as well. The rc.local file that is installed with Version 3.0 contains the following correct loopback driver entries:

```
/etc/ifconfig ni0 `/bin/hostname` broadcast 128.45.45.0\
netmask 255.255.255.0
/etc/ifconfig lo0 localhost
```

1.2 Overview of System Files

This chapter describes some of the ULTRIX system files that you can create or modify after your system has been installed. These files perform a number of different functions; for example, they enable user log in, the setting up of mail aliases, and the display of a login message of the day.

The following sections describe:

- The password file
- The group file
- The terminal initialization file
- The file system table
- The sendmail aliases file
- The clock daemon table
- The message of the day file

Most system files are created automatically during the system installation. However, you can create or modify a file after the system has been installed. For example, to change login information for system users, modify the /etc/passwd and /etc/group files. To specify sendmail aliases, create or modify the /usr/lib/aliases file. In most cases you will be using a system editor, such as vi, to change the content of the files. Where applicable, the system tools available to modify the files are identified.

The following sections contain descriptions and sample entries for each of the listed system files:

1.3 The Password File

The /etc/passwd file is a data file that contains an entry for every user who has login privileges on your system. Although it is called the password file, it contains more than user passwords. Figure 1-1 illustrates the contents of a password file created for an ULTRIX system that runs the Yellow Pages.

```
root:R,r97fsje2oss:0:1:System PRIVILEGED Account:/:/bin/csh
field:Pa9rek3.1l5e:0:1:F S PRIVILEGED Account:/usr/field:/bin/csh
operator:sruF3.9ir,ePw:0:28:Operator PRIVILEGED Account:/opr:/opr/ops
guest:n3Rel9s22:10:33:Guest account:/tmpguest:/bin/date
jjd::34:10:John Joseph Doe:/usr/staff/jjd:/bin/csh
jws::24:10:John Walter Smith:/usr/staff/jws:/bin/csh
+:
```

Figure 1-1: Sample Password File

1.3.1 Modifying The Password File

The system automatically creates a generic /etc/passwd file at installation, but you must modify this file to include site and user specific information.

- By default, the /etc/passwd file is Read Only; only the superuser can edit the file itself. However, once login accounts exist, registered users can invoke system commands to change their password and default shell.
- The easiest way to modify password entries in the /etc/passwd file is with the passwd command. The superuser can change any user password. Individual users can change only their own password.
- The easiest way to add or delete users in the /etc/passwd file is with the adduser and removeuser commands. Only the superuser can invoke these commands. See Chapter 2 for a description of their use.

Each entry in the /etc/passwd file contains the following information: the user login name, the user password, the user id, the group id, a description of the user, the pathname of the user home directory, and the pathname of the default shell or command to be executed immediately following login. The format for each entry is:

name:[password]:user-id:group-id:description:home-directory:[shell]

name — The first field contains the user's login name (1 to 8 characters). The system uses this name in establishing login permission.

password — The second field contains the user's encrypted password. An entry in this field is optional. The system uses this password in verifying login permission.

user-id — The third field contains the user's identification number (user ID). The system uses this number to determine a user's identity. Once login permission is granted, the system internally translates the login name to this user ID number and uses it in identifying the user's processes and in determining owner access permission to files. This number must be unique for each user and should be less than 32000.

group-id — The fourth field contains the primary group identification number (group ID). The system uses this number to determine a user's default group classification. You can set up the permissions for any file so that users with the same *group-id* numbers can access the file, but those users with different *group-id* numbers cannot. Once login permission is granted, the system internally establishes the user's group ID number and uses it in determining group access permission to files. This number must be unique for each group and should be less than 32000.

A user may belong to a maximum of eight groups. The /etc/passwd file lists only the user's primary group. You can see the user's secondary groups by displaying the /etc/group file.

description — The fifth field contains additional user information. For example: user name, office location, office phone number, and home phone. The user name can be an ampersand (&), which means that the login name and the user name are the same. User's can maintain the *description* field with the chfn command.

home-directory The sixth field contains the absolute pathname to the user's home (initial working) directory. After establishing the appropriate user and group identification, the system uses this pathname to place the user in the named directory.

shell The seventh field contains the absolute pathname to the command that is to be executed immediately upon conclusion of the login process. This is normally a version of the shell (command interpreter) such as /bin/csh or /bin/sh. It can also be used to allow the user limited access to the system. By replacing a shell version with the full pathname of a particular command, the user will only be allowed to execute that command. Furthermore, the system will log the user out once he has executed that command.

An entry in this field is optional. If nothing is specified, the system automatically invokes /bin/sh.

1.3.2 Using the passwd Command

In the preceding sample, only some of the user entries contain an encrypted password in the *password* field. When creating accounts for new users, you are not required to supply a password. After users log in for the first time (no password is required when the password field is blank), they can create their own password using the passwd(1) command as described in the ULTRIX Reference Pages.

1.3.2.1 Improving System Security – To improve system security when modifying the /etc/passwd file, follow these procedures:

1. Preassign the password for all new users on your system. To do this:
   - Log on as superuser.
   - Enter the passwd command giving the user's login name as an argument. For example, to create a password for a user named francis, type:

```
# passwd francis
```

- When the passwd command prompts for the new password, type a simple password.

  The program encrypts this password and includes it in the /etc/passwd file. The user can then log in to the account with the assigned password and change it. See passwd(1) in the ULTRIX Reference Pages for additional details.

2. For file locking protection, use the vipw command to edit the /etc/passwd file. See vipw(8) in the ULTRIX Reference Pages for details.

1.3.2.2 Changing the Shell and Description Fields – A registered user can change the assigned login shell using the chsh command, giving his login name and the pathname of the desired shell as arguments. See chsh(1) in the ULTRIX Reference Pages for a description of this command and format.

Similarly, a registered user can change information in the description field with the chfn command, giving his login name as an argument. See chfn(1) in the ULTRIX Reference Pages for details.

1.3.3 Accessing the Yellow Pages Network

If your system includes the Yellow Pages software, you must make certain adjustments to your /etc/passwd file.

- If Yellow Pages is installed before you run uucpsetup or netsetup, you must edit the /etc/passwd file and move the plus colon entry (+:) to the last line of the file. The sample /etc/passwd file shown earlier in this chapter illustrates this case.
- If you want to use the uucp utility throughout your YP network, you must run uucpsetup on the YP master server machine and then remake the password maps.
- If you want to use the uucp utility for local use only, you can run uucpsetup on your YP client machine.

See the Guide to the Yellow Pages Service for further information.

1.4 The Group File

Data for groups and group members is stored in the system group file /etc/group. This data file allows users with dissimilar group IDs in the /etc/passwd file to access common files associated with a particular task, such as a large programming project. The system uses the /etc/group file to determine which users are in which groups and to establish access

permissions to those files created by group members.

To add groups or group members, edit the /etc/group file by creating a new group entry or adding the user name to an existing entry.

Each entry in /etc/group contains information that the system uses when translating individual group IDs to names and in verifying simultaneous membership in multiple groups. A user can be assigned to a maximum of eight groups. Each entry contains four fields of information which are delimited by colons. Also, each group file entry can be no more than 1024 characters in length.

1.4.1 The Group File Format

The format of the /etc/group file is:

group:password:group-id:name,name...

group The name of the group.

password The encrypted password. This field is not used, but it should not be empty. When creating a new entry, put an asterisk (*) in this field. The asterisk eliminates group password matching.

group-id The group identification number. The system uses this number to determine group access permissions to files. The user's primary *group-id* is listed in the passwd file. The users' secondary groups are listed here in the /etc/group file. The *group-id* must be unique and should be be less than 32000.

name The login names of the current group members, all of which are delimited by commas. Each group file entry can have as many as 200 members, provided that the total number of characters does not exceed 1024 characters. When you enter the member login names, it is acceptable to let the entry wrap onto the next line of the display screen.

1.4.2 Sample Group File Entries

The following is a sample entry from a /etc/group file:

```
research:*:25:jones,wilson
```

This entry contains information for a group called research. The password field contains an asterisk, which eliminates password matching. The group ID is 25, and current members are Jones and Wilson.

To add members to an existing group, edit the /etc/group file by adding the users' login names to the appropriate list. For example:

```
research:*:25:jones,wilson,smith,thompson
```

This entry now lists four current members: Jones, Wilson, Smith, and Thompson.

To delete either an existing group or an existing member from a group, edit the /etc/group file by deleting the appropriate entry or name.

For further information, see group(5) in the ULTRIX Reference Pages.

1.5 The Terminal Initialization File

You may have to modify a configured terminal for daily operations. To modify a configured terminal, edit the terminal initialization file /etc/ttys by modifying the information listed for the appropriate terminal entry.

Each entry in the /etc/ttys file contains information that the getty command uses during system boot to determine if a terminal is to be opened for login. There is one entry in the /etc/ttys file for each tty line. Each entry can contain 512 characters. Each entry in the /etc/ttys file contains five fields of information separated by spaces as described in the following section. For further information, see getty(8) and login(1) in the ULTRIX Reference Pages.

1.5.1 The Terminal Initialization File Format

The format of the /etc/ttys file is:

name command type status window="string" description

name The *name* field defines the terminal's special file (device) name, as listed in /dev. All terminals except network pseudoterminals, workstation pseudoterminals, and modems use the following naming convention:

tty[0-9A-Z][0-9]

The terminal names range from tty00 to tty99 and from ttyA0 to ttyZ9 for a total of 360 possible tty names.

Network pseudoterminals use the following naming convention:

pty[qrstu][0-9a-f] tty[qrstu][0-9a-f]

This convention yields a total of 96 matched pairs ranging from ptyq0 (ttyq0) to ptyuf (ttyuf).

Workstation terminals follow the same naming convention as network terminals which is defined as follows:

This yields a total of 16 matched pairs ranging from ptyv0 (ttyv0) to ptyvf (ttyvf).

The modem (dialup) lines have the following convention:

ttyd[0-9a-f]

This convention yields 16 dialup line names ranging from ttyd0 to ttydf.

command The *command* field contains the name of the command to be executed each time the terminal is initialized. This field also contains command arguments, if applicable. If the command has arguments, both the command and its arguments must be enclosed by double quotes. If there are no arguments, then quotes are unnecessary.

The *command* usually contains /etc/getty, which allows users to log in to the system. If the *command* field contains /etc/getty, be sure to include the terminal's baud rate. For further information on getty, see getty(8) and gettytab(5) in the ULTRIX Reference Pages.

If the terminal is a pseudodevice, then the *command* field should contain the string none.

type The *type* field defines the terminal type such as a vt100. If the terminal is a pseudodevice, then *type* should be network.

status The *status* field specifies the status of the terminal line. The terminal line can have up to four status values:

off or on
: If the status is off, then the *command* and its associated terminal line is disabled. For pseudodevices, set the status to off. if *command* is /etc/getty and the status is off, then the terminal is disabled for logins. If the status is on, then the *command* and its associated terminal line is enabled. If nothing is specified, then off is the default.

modem or nomodem
: If the status is modem then the terminal is a modem (dial-in and dial-out). If the status is nomodem, then the terminal is hardwired. If nothing is specified, then modem is the default.

secure
: If the terminal is secure, then superusers (root) can log in to the system on that terminal line. If nothing is specified, then the terminal line is not secure, meaning

that superusers cannot log in to the system on that line.

shared or not shared
: If the terminal is shared, then logins, tip, and uucp can use the same terminal (not simultaneously). If nothing is specified, then not shared is the default, meaning that logins require one terminal, and tip and uucp require another terminal. For further information, see tip(1) and uucp(1c) in the ULTRIX Reference Pages.

window="string"
: The quoted string is a window system process that init maintains for the terminal line. For further information, see init(8) in the ULTRIX Reference Pages.

description You should put any comments about the entry in this field, preceded by a #.

1.5.2 Sample Terminal Initialization File Entries

Here are sample entries from a /etc/ttys file:

```
console "/etc/getty std.9600" vt100    on  secure  # normal console
tty01   "/etc/getty std.9600" vt100    on  nomodem # direct connect tty
ttypa   none                  network off secure  # network tty
```

To modify a configured terminal, edit the /etc/ttys file and change the appropriate field of the appropriate entry. For example, to enable a configured terminal, modify the *status* field of the appropriate entry to read on. Similarly, to disable a configured terminal, edit the /etc/ttys file to modify the *status* field of the appropriate entry to read off.

Note

The default console entry in the /etc/ttys file is set up to be a 1200 baud hardcopy terminal. If you are using a faster device (such as a display terminal) for a system console, you must change the /etc/ttys file to a faster baud rate as shown in the preceding example. The next time you boot your system, init will automatically act on the modified entry.

To implement a change during multiuser mode, use the kill command to send a hangup signal to init. For example:

```
# kill -HUP 1
```

This command sends a hangup signal to init which rescans the /etc/ttys file and processes only those entries that have been modified. For further information, see kill(1) and init(8) in the ULTRIX Reference Pages.

1.6 The File System Table

When necessary, you may have to modify the file system table, /etc/fstab, to reflect required changes (for example, adding a new file system or changing the order in which your file systems are to be loaded, dumped, or checked). To modify the file system table, you should edit /etc/fstab by either creating a new entry, or adding or changing information for an existing entry.

Before editing /etc/fstab, you should familiarize yourself with its format. Each entry in /etc/fstab contains default information for the mount, dump, and fsck commands. For further information, see mount(8), dump(8) and fsck(8) in the ULTRIX Reference Pages.

1.6.1 The File System Table Format

Each entry in /etc/fstab contains seven fields of information that are delimited by colons. As defined in the /usr/include/fstab.h file, the format of an /etc/fstab file is:

spec:file:type:freq:passno:name:opts:

spec The *spec* field defines either the file system's block special file (device) name, or it defines a remote file system like Network File System (NFS). Examples are: /dev/ra0g for a local system and /usr/src@erie for a remote system.

file The *file* field defines the absolute pathname to the directory on which the file system is mounted. This is used as the default information for the mount command.

type The *type* field specifies the file system mode: rw (read-write), ro (read only), rq (read-write with quotas), sw (swap), and xx (ignore). If sw is specified and if the file system has been configured for such use, swapon (invoked by /etc/rc) makes that file system part of the system swap space. If xx is specified, the local file system is ignored, that is, not processed by mount, dump, or fsck.

freq The *freq* field specifies the file system dump frequency (every *n*th day). This is used as default information for the dump command. For NFS entries, this field should be a 0 value.

passno The *passno* field defines the file system pass number. This is used as the default order for the fsck command. Usually, only the root file system has a pass number of 1. The remaining file systems should be assigned higher pass numbers, which enables the fsck utility to simultaneously check file systems in parallel.

Before you create a new entry, first consider that all file systems on a single disk should have different numbers. Next, consider that the fsck utility can check file systems on different disks at the same time. Therefore these file systems can have the same pass number.

You should also be aware that NFS entries should have pass numbers of 0 so that local execution of certain commands, such as fsck and dump will ignore these particular /etc/fstab entries. The reason for this is because NFS file systems specified in the /etc/fstab file identify remote file systems that are usually maintained by someone at the remote system site.

name The *name* field defines specific file systems that you are mounting. Supported file systems are: UFS and NFS.

opts The *opts* field defines file system-specific options that are being passed to the file system being mounted.

To add an entry to the file system table, edit the /etc/fstab file.

1.6.2 Sample File System Table Entries

Here are two sample entries from an /etc/fstab file:

```
/dev/ra0g:/usr:rw:1:2:ufs::
/usr/src@erie:/usr/src:ro:0:0:nfs:bg
```

The first entry lists information for the ULTRIX File System (UFS), the local file system that resides on a disk's g partition and is mounted read-write on the /usr directory. This file system is dumped every day by dump and is checked by fsck on the second pass. The second entry shows a remote file system entry. Note that this file system is mounted read only and that the *freq* and *passno* fields are both zero so that locally executed commands such as dump and fsck will ignore these fstab entries.

Note

Because mount, dump, and fsck process the entries in order, according to the pass field, the order of the entries in the /etc/fstab table is important. For example, mount will fail if it is told to mount a file system on a directory that itself is not yet mounted. You should make sure that each entry in /etc/fstab is listed in a logical order. You should also ensure that NFS

entries have *freq* and *passno* field values of zero. For example:

```
/dev/ra0a:/:rw:1:1:ufs::
/dev/ra0h:/usr:rw:1:2:ufs::
/dev/ra0f:/usr/staff:rw:1:3:ufs::
/usr/man@suez:/usr/man:ro:0:0:nfs:bg:
```

For more information on the fstab file, refer to fstab(5) in the ULTRIX Reference Pages. For more information on NFS, refer to the Guide to the Network File System.

1.7 The Sendmail Aliases File

Each entry in the /usr/lib/aliases file contains information that the sendmail utility uses to route messages to users when an alias is specified in a mail message. To add a user name to this file, edit the /usr/lib/aliases file by either creating a new alias entry or adding a new name to an existing entry.

Before editing /usr/lib/aliases, you should familiarize yourself with its format.

1.7.1 The Sendmail Aliases File Format

Each entry in /usr/lib/aliases contains two fields of information separated by a colon. The first field is the alias name. The second field is a string of user names separated by commas. The format of the /usr/lib/aliases file entries is:

alias:user,user...

alias The *alias* field identifies the name of the alias group.

user The *user* field contains a string of user login names, which are separated by commas. This information lists the user names to which the mail is to be routed when the alias is specified. The list of users can extend beyond one line.

To add user names to the alias file, edit /usr/lib/aliases.

1.7.2 Sample Sendmail Aliases File Entries

Here is a sample entry from a /usr/lib/aliases file:

```
research:jones,wilson,smith,thompson
```

This entry lists the names (Jones, Wilson, Smith, and Thompson) to which sendmail routes messages when the research alias is specified.

To make these additions immediately become part of the sendmail aliases data base, use the newaliases command. For example:

```
# newaliases
```

This command reads the new information added to /usr/lib/aliases and rebuilds the sendmail aliases data base.

For further information, see newaliases(1), aliases(5), and sendmail(8) in the ULTRIX Reference Pages.

1.8 The Clock Daemon Table

The clock daemon table, /usr/lib/crontab, is a symbolically linked file that contains routine commands which the system clock daemon, cron, executes at the specified dates and times. For example, /usr/lib/crontab might contain routine backup commands as well as commands that cause the automatic removal of outdated or unused temporary files.

Once invoked during multiuser startup, the system activates the system clock daemon, cron, every 60 seconds. In turn, the system clock daemon executes those commands listed in the /usr/lib/crontab file (clock daemon table) that are scheduled for that time.

Since each entry in /usr/lib/crontab contains information that specifies a time and command sequence that is to be executed regularly, you should stagger the crontab entry times so that the processes are not all running at once. When appropriate and especially during anticipated periods of heavy user activity, you should also include the nice command in your crontab file entries so that the crontab commands execute at a lower priority.

To change the clock daemon table, edit /usr/lib/crontab. For further information on the clock daemon and prioritizing tasks, see cron(8) and nice(1) in the ULTRIX Reference Pages.

1.8.1 The Clock Daemon Table Format

Each entry contains six fields of information separated by spaces. The format of the entries in the /usr/lib/crontab file is:

minute hour day month weekday command

minute The exact minute that the command sequence is to be executed. The *minute* variable can be 0 through 59.

hour The hour of the day on which the command sequence is to be executed. The *hour* variable can be 0 through 23.

day The day of the month on which the command sequence is to be executed. The *day* variable can be 1 through 31.

month The month of the year on which the command sequence is to be executed. The *month* variable can be 1 through 12.

weekday The day of the week on which the command sequence is to be executed. The *weekday* variable can be an integer from 1 to 7. Monday equals 1 and Sunday equals 7.

command The command sequence that is to be executed. The *command* variable should contain the complete command sequence.

In addition, the first five fields may specify either a single time indicator, a multiple time indicator, a time range, or an asterisk. A single time indicator may consist of one or two consecutive digits such as 3 or 33. A multiple time indicator consists of a string of indicators separated by commas, such as 5,10,15,20. A time range consists of two indicators separated by a dash, such as 5-20. An asterisk field entry represents all times.

1.8.2 Sample Clock Daemon Table Entry

The following is a sample entry from the /usr/lib/crontab file:

```
0,30 * * * * /usr/lib/atrun
```

This entry specifies that the cron command is to invoke /usr/lib/atrun daily on the hour and half hour. For further information, see cron(8) in the ULTRIX Reference Pages.

1.9 The Message-of-the-Day File

You may have to provide the system users with information that is relevant to that day's operation. To do this, create a message-of-the-day using the /etc/motd file.

As a general rule, the first line of the message-of-the-day file should contain the adjective ULTRIX-32, the current version of the operating system, and the day's date.

After each login, the system displays the contents of the /etc/motd file at the user's terminal. The system displays this same message after each login until you either modify or delete the contents of /etc/motd.

Adding and Deleting Users 2

This chapter describes how to add and delete users. To add users to your system, use the adduser command. To remove users from your system, use the removeuser command. You can also add or remove users manually with the vipw command.

Note

If you install the Yellow Pages (YP) service as described in the Guide to the Yellow Pages Service, you cannot use the adduser or the removeuser commands. Refer to ypsetup(8yp) and to the Guide to the Yellow Pages Service for information on setting up the YP service. If the YP service is enabled, use the vipw command to make changes to the /etc/passwd file.

2.1 Adding Users

The adduser command is an interactive facility for adding new user accounts to the /etc/passwd system password file. It also adds user login names to the /etc/group file. For each new user account, the adduser command sets up a home directory with the C shell files and the Bourne shell file and a bin subdirectory. New accounts created in this way initially have no password.

To add a user to the system, type:

```
# /etc/adduser
```

The prompts request information about the user. The default answers are provided in square brackets for some of these entries. You can select these defaults by pressing the RETURN key.

The adduser command also adds users to the /etc/group file provided that the group and the directory for that group already exist. See Chapter 1 for more information on the group file.

Note

To add a user to a group other than users, you must create a directory for that group before issuing the adduser command. For information on adding directories, see mkdir(1) in the ULTRIX Reference Pages.

Be aware that if your system uses the Yellow Pages service, the following message appears at your terminal in response to the /etc/adduser command:

This system makes use of the Yellow Pages service for user account administration. Please add the new user by following the steps given in the Overview of Yellow Pages chapter of the Network Management Guide.

Example 2–1 depicts a sample interactive session with the adduser program. In this case, the system does not use the Yellow Pages service.

Example 2-1: Sample adduser Sequence

```
#/etc/adduser

Enter login name for new user (initials, first or last name): jal
Enter full name for new user: Jim A. Laker
What login group should this user go into [ users ] ? users1
Working ...

Enter another group that 'jal' should be a member of
(<RETURN> only if no more): users2
Working ...

Unknown group: users2.
Known groups are:
system          daemon          uucp            bin
news            users1          ris             users
guest           operator        ingres          staff

Do you want to add group users2 to the /etc/group file [yes] ?

Adding new group to /etc/group file...
```

(continued on next page)

```
Enter group number for new group [105]:

Enter another group that 'jal' should be a member of
(<RETURN> only if no more):
Working ...

Enter parent directory for jal [/usr/users]: /usr/users1

Adding new user ...

The new user account initially has no password.
#
```

This sequence of prompts and user entries shows that after the user's login name and full name were entered, the adduser command prompts for the user's login group, which has a default of users. Notice that when users2 was entered as a secondary group, the adduser command found that this group did not exist. The adduser command next displays the known groups and asks if users2 should be added to the /etc/group file. The final adduser prompt is for the entry of the user's parent directory. This is the user's home directory, which is the sixth field of the /etc/passwd file.

After the adduser command informs you that it has added the new user to the system, you can verify the addition by checking the /etc/passwd file. For example, type:

```
# more /etc/passwd
```

2.2 Deleting Users

Use the removeuser command to delete users from the system password file. The removeuser command also gives you the option of deleting the user's home directory and files. This command displays default responses in square brackets and enables you to select them by pressing the RETURN key. For further information, see removeusers(8) in the ULTRIX Reference Pages.

Note

You cannot use the removeuser command if you have enabled the YP service as described in the Guide to the Yellow Pages Service.

The removeuser command does not remove a user's name from the /etc/group file. To remove a user from a group, you must edit the /etc/group file.

To remove a user from the system password file, type:

```
# /etc/removeuser
```

The prompts request information about the user you want to remove from the system.

Example 2–2 shows the sequence of prompts and user entries that are typically entered to remove a user from the system.

Example 2-2: Sample removeruser Sequence

```
# removeuser

Enter login name for user to be removed: jal
This is what the entry in /etc/passwd looks like:

     jal::543:15:Jim A. Laker:/usr/users1/jal:/bin/csh

Is this the entry you wish to delete? y
Working ...

User jal removed.

Do you want to remove jal's home directory,
all subdirectories and files (y/n)? y

You should have backed up jal's files if you do not wish to lose them.

Are you sure that you want to remove jal's files (y/n)? y

Deleting /usr/users1/jal

#
```

This sequence of prompts and user entries shows you what will happen when you use the /etc/removeuser command to delete the user named jal from the system. After entering the user's login name, the removeuser command displays the /etc/passwd file entry and then prompts you to remove the entry. If you type y, the removeuser command removes the entry from the /etc/passwd file and prompts you to delete all subdirectories and files. Last, the command tells you that it is deleting /usr/users1/jal directory.

Adding Devices 3

This chapter describes how to add terminals, pseudoterminals, and device drivers to your system. It is organized as follows:

- Adding terminals and pseudoterminals
- Adding disk and tape drives

Before adding any device to your system configuration, check your system configuration file to see if a description of that device type is entered there. If the device description is in the configuration file, then you use the MAKEDEV script to add the particular device to your system. If not, you must rebuild and boot a new kernel after adding the device. The configuration file and building the kernel is described in the Guide to System Configuration File Maintenance.

When you add a new device to your system configuration with the MAKEDEV script, a special file is created for the device in the /dev directory. You can use the file command to display information about the special files in the /dev directory. For example, to display the status of each special file in the /dev directory, type:

```
% file /dev/* | more
```

For further information, see file(1) and MAKEDEV(8) in the ULTRIX Reference Pages.

3.1 Adding Terminals and Pseudoterminals

When adding lines and multiplexers, you must physically connect the devices as follows:

1. Stop system activity by using the /etc/shutdown command and then power down the system by turning off the processor or by issuing the correct console command. For further information, see the Guide to System Shutdown and Startup.
2. If appropriate for your device, set the control status registers (CSRs) and interrupt vectors on the board. Most DIGITAL devices have standard addresses. Refer to the appropriate device manual for the standard address.

3. Install the board.
4. Power up the machine.
5. Boot the system. Refer to the Guide to System Shutdown and Startup for information on booting your processor.

After physically connecting the terminal line or multiplexer, follow the procedures outlined in this text to complete the addition of the device.

3.1.1 Adding Terminals

The following procedure explains how to add a terminal multiplexer to your system. This procedure applies to terminal multiplexers for all machines.

To add a terminal multiplexer, refer to Appendix A for the MAKEDEV device mnemonic for your controller, then follow these steps:

1. Use the cd command to move into the /dev directory,

```
# cd /dev
```

2. Invoke the MAKEDEV script for your controller using the format:

```
MAKEDEV device#
```

   The device variable is the mnemonic device name for the terminal you are adding. The # variable is the device number 0 through 10. When using MAKEDEV, be sure to save its output since you will refer to this output when updating the /etc/ttys file.
3. Edit the /etc/ttys file to reflect the new tty lines that MAKEDEV created. For a discussion of the /etc/ttys file, refer to Chapter 1.
4. Rebuild and boot a new kernel, if necessary.

3.1.2 Adding Pseudoterminals

To add a pseudoterminal, refer to Appendix A for the appropriate MAKEDEV device mnemonic, then follow these steps:

1. Use the cd command to move into the /dev directory,

```
# cd /dev
```

2. Invoke the MAKEDEV script for your pseudoterminal using the format:

```
MAKEDEV pty#
```

   The # variable specifies which set (0 through 5) of pseudoterminals you want to create. Each single set creates 16 pseudoterminals, for a total of 96 in the system. By default, the installation creates the first two sets of pseudoterminals, pty0 and pty1. These lines are named /dev/ttyp0 through /dev/ttypf and /dev/ttyq0 through /dev/ttyqf.

Thus, to create the next logical set of pseudoterminals, named /dev/ttyr0 through /dev/ttyrf, invoke the MAKEDEV script as follows:

```
# MAKEDEV pty2
```

When using MAKEDEV, be sure to save its output since you will refer to this output when updating the /etc/ttys file.

3. Edit your /etc/ttys file to reflect the new pseudoterminal lines that MAKEDEV created for you. For a discussion of the /etc/ttys file, refer to Chapter 1.
4. Modify the system configuration file, /usr/sys/conf/*SYSTEM_NAME, then rebuild and boot a new kernel, if necessary. For further information, see the Guide to System Configuration File Maintenance.*

3.2 Adding Disk and Tape Drives

After you install the new controller at a DIGITAL standard address and successfully run the diagnostics, you must physically connect the devices as follows:

1. Stop system activity by using the /etc/shutdown command and then power down the system by turning off the processor or by issuing the correct console command. For further information, see the Guide to System Shutdown and Startup
2. If appropriate for your device, set the control status registers (CSRs) and interrupt vectors on the board.
3. Install the board.
4. Power up the machine.
5. Boot the system. Refer to the Guide to System Shutdown and Startup for information on booting your processor.

To add a drive to your software configuration, refer to Appendix A for the MAKEDEV device mnemonic for the drive you are adding, then follow these steps:

1. Use the cd command to change to the /dev directory:

```
# cd /dev
```

2. Invoke the MAKEDEV script for the controller using the format:

```
MAKEDEV device#
```

The device variable is the mnemonic device name for the drive you are adding. The # variable is the device number 0 through 31.

3. Modify the system configuration file, /usr/sys/conf/*SYSTEM_NAME*, *then rebuild and boot a new kernel, if necessary. For further information, see the Guide to System Configuration File Maintenance.*

This example shows how to add two disk drives to your system:

```
# MAKEDEV ra0 ra1
```

This example shows how to add two tape drives to your system:

```
# MAKEDEV tms0 tms1
```

3.3 Setting Up the System Console

The default console entry in the /etc/ttys file is set to **e** for the baud rate. For example:

```
console "/etc/getty e"
```

If you have a hard-copy console, you can accept this default. However, if you have a CRT console, you must change the entry in the /etc/ttys file to std.9600. For example:

```
console "etc/getty std.9600"
```

Setting Up the Print System 4

The ULTRIX–32 print system software supports a wide variety of printers configured within a local or remote network. This chapter describes the files and programs that enable this support and that you use in setting up, modifying, and maintaining your print system.

4.1 Print System Files and Programs

There are three broad categories of print system programs and files:

- Print system database and support
- Print system setup
- Print system control

The following sections briefly identify and describe the components of each category; subsequent sections in this chapter detail their use.

4.1.1 Print System Database and Support Files

The print system database and support files identify the elements of your print system. They specify what is available and how it is configured at your site. These files are:

- /etc/printcap – This file provides a descriptive database of your print system software and hardware. The ULTRIX-32 software provides a generic /etc/printcap file. You modify this file to define your configuration explicitly. At minimum, your /etc/printcap entries should include a definition of the available printer(s), the local or remote host(s), the print filter(s), the baud rate(s), the spooling directory(s), and the output device associated with each printer. All users have read access to this file; only the superuser has write access. See printcap(5) in the ULTRIX Reference Pages and Section 4.3.2 for more information.
- /etc/printcap.examples – This file contains examples of line printer definitions. You can use this file primarily as a sample or you can copy the file as a template for your /etc/printcap file.

- /dev/MAKEDEV – This shell script creates a device special file in the /dev directory for each line printer. Only the superuser has access to this script. See MAKEDEV(8) in the ULTRIX Reference Pages and Section 4.3.3.2 for more information.
- /usr/lib/lpd – This program, the line printer daemon, is a print spool handler. Normally, the program is invoked at boot time from the rc file. The daemon works with several system programs and files to coordinate and synchronize printer activity. The ULTRIX-32 software supplies this file. While you do not modify the file, you can specify spooling, logging, and locking activities via the command line. Only the superuser can access this program. See lpd(8) in the ULTRIX Reference Pages and Section 4.5.1 for details.
- /usr/lib/lpdfilters – This directory contains print filter files used by print programs. Only the superuser has write access to these files; all users have read and execute permissions. See Section 4.3.3.6 for more information plus a listing of supported print filters and their inclusion in the /etc/printcap file.
- /etc/hosts.lpd – This file contains the names and network addresses of local and remote machines that can access system printers. /usr/lib/lpd uses this file to check access privileges. See lpd(8) in the ULTRIX Reference Pages and Section 4.3.3.7 for more information.
- /etc/hosts.equiv – This file contains the names (but not the network addresses) of "trusted" remote hosts. Privileged access to a host machine is granted to root if the name of the host is contained within both the /etc/hosts.lpd file and the /etc/hosts.equiv files. See lpd(8) in the ULTRIX Reference Pages and Section 4.3.3.7 for more information.

4.1.2 The Print System Setup Program

The ULTRIX-32 software provides an automatic printer setup program to simplify the task of setting up and modifying your print system. lprsetup provides an interactive, easy-to-use method; it takes the information that you supply, creates or deletes the spooling directory, modifies the /etc/ttys file to prevent logins on the printer line, and creates or deletes the corresponding /etc/printcap entry. See lprsetup(8) in the ULTRIX Reference Pages and Section 4.4 for additional information.

4.1.3 The Print System Control Files

In general, a well planned and structured print system needs little maintenance. Nevertheless, the ULTRIX-32 software provides files and programs that simplify your work with the print system. For example:

- /etc/lpc – This program lets you control the operation of the line printer system. While most control functions are available only to the superuser, some functions can be accessed by general users. See lpc(8) in the ULTRIX Reference Pages and Section 4.5.2 for details.
- /etc/pac – This program generates accounting information about printer use at your site. Only the superuser can access this program. See pac(8) in the ULTRIX Reference Pages and Section 4.5.6 for details.
- /usr/ucb/lpr – This program lets you queue and submit files for printing. See lpr(1) in the ULTRIX Reference Pages and Section 4.5.3 for details.
- /usr/ucb/lpq – This program lets you examine the status of jobs currently on the print queue. See lpq(1) in the ULTRIX Reference Pages and Section 4.5.4 for details.
- /usr/ucb/lprm – This program lets you remove jobs from the print queue. See lprm(1) in the ULTRIX Reference Pages and Section 4.5.5 for details.

4.2 Choosing a Print System Setup Method

You have two options for setting up or modifying your print system:

1. You can set up your printer system manually. If you are familiar with ULTRIX-32 and the print system, or if you are making a small modification to your current setup, you might prefer to make changes manually. Section 4.3 describes this process.

2. You can use lprsetup, an interactive program that walks you through the process and provides on-line help. If you are setting up a print system for the first time, you might prefer this procedure. Section 4.4 describes the program and its use.

Choose the option that best complements your experience and the requirements of your site. The following sections describe each method and the related files and programs.

4.3 Manually Setting Up or Modifying Your Print System

To set up or modify your print system manually, you must first create or edit several files. To help you in this process, the following sections describe how to:

- Collect prerequisite information

- Identify the contents of the /etc/printcap file
- Modify the /etc/printcap file
- Select and define the logical device name for each printer
- Create a device special file for each printer
- Create and define a spooling directory for each printer
- Identify and define the baud rate for each printer
- Identify and define the output filter for each printer
- Identify and define additional printer symbol entries
- Identify and define remote printers
- Use the /etc/printcap.examples file
- Modify the /etc/ttys file

4.3.1 Collecting Prerequisite Information

Your primary goal is to build a data base of information about the type and capabilities of printers supported at your site. To build this data base, first collect the following information about your print system configuration:

- Determine which printers are attached via parallel lines, and which printers are attached via serial lines. This data affects your entries in both the /etc/ttys file and the /etc/printcap file.
- Identify the physical port to which each printer is attached. You need this information when creating special device files and when making printer specification entries in the /etc/printcap file.
- Identify each remote printer you intend to configure into your print system. You need this information when making remote printer specification entries in the /etc/printcap file.
- Identify the capabilities of each printer, including the required print filters. You need this information when making printer synonym entries in the /etc/printcap file.
- Identify the parity requirements of applications software in use at your site. You need this information when examining or modifying several printer synonym entries in the /etc/printcap file.

4.3.2 Identifying the Contents of the /etc/printcap File

You identify the local and remote printers within the /etc/printcap file. These definitions establish how the ULTRIX-32 software knows and controls access to each printer specified within your configuration. A generic version of /etc/printcap is created during the automated phase of

system installation. However, you must modify this file to define the specific local or remote printers you want accessed.

To modify /etc/printcap, you must understand each field in the line entries of the file. Be aware that:

- Each field in /etc/printcap is separated from the next by a colon.
- A backslash indicates that the next line is part of the same entry.
- The first line contains the printer's logical device name(s) and a description of the printer configuration for the /etc/printcap entry.
- Subsequent lines define the parameters for each printer. If you do not specify certain parameters in /etc/printcap, the print system uses a default value.

See printcap(5) in the ULTRIX Reference Pages for a complete description of the printcap fields and default values.

The following sample depicts an extract from an /etc/printcap file where the system configuration includes local and remote hosts with attached printers.

```
#
#       LINE PRINTERS
#
lp2|2|lab2|tucson's lg01 line printer in Lab:\
        :lp=:rm=tucson:rp=lp2:sd=/usr/spool/lp2:mx#0:
#
lp3|3|lab3|lg02 line printer in Lab:\
        :lp=:rm=phoenix:rp=lp3:sd=/usr/spool/lp3:mx#0:
#
#
#    LASER PRINTERS
#
ln|ln01|laser|lna|ln01a|primary ln01 laser printer:\
        :lp=:rm=tucson:rp=lna:sd=/usr/spool/lna:mx#0:
#
#ln|ln01|laser|lna|ln01a|primary ln01 laser printer:\
#       :lp=/dev/lph:sd=/usr/spool/lna:\
#       :dn=/usr/lib/lpd:\
#       :pp=/usr/lib/lpdfilters/ln01pp:\
#       :if=/usr/lib/lpdfilters/ln01of:\
#       :vf=/usr/lib/lpdfilters/ln01vf:\
#       :gf=/usr/lib/lpdfilters/ln01Sgf:\
#       :nf=/usr/lib/lpdfilters/ln01Snf:\
#       :pw#80:pl#66:mc#20:sf:
#
```

Figure 4-1 Sample /etc/printcap File Extract

4.3.3 Modifying the /etc/printcap File

Once you are familiar with the format of the /etc/printcap file, and have collected the necessary information about your print system configuration and components, you can modify the /etc/printcap file.

4.3.3.1 Identifying the Name and Number of the Printer – The first required item in any /etc/printcap entry is the name and number of the printer. For example,

```
lp0|lp10|parallel port line printer:
```

In this example, the lp0 entry is for a parallel line printer interface. Since lp0 is generally your primary line printer, it uses the /etc/printcap entry lp by default. If you use the command lpr, your file will print on this printer. You could alternately specify lpr -P0, which would also print your file.

You can enter alternate names for the printer, such as "letter", "draft" "LA-180 DecWriter III." If the name contains blanks or tabs, it should be specified last.

4.3.3.2 Choosing The Logical Device Name – You must relate the physical connection (the port) of each local printer to a logical device name for that printer.

Note

The port information should be in the Site Management Guide, prepared by the DIGITAL field service personnel who installed the printer hardware.

The logical device name is specified in the lp= field of the printer entry within /etc/printcap. For example, in the first sample within Figure 4-2, the logical device name associated with the LP25 printer is specified as :lp=/dev/lp:

To choose a logical device name for your printer, relate the port number to a logical device name. Logical device names have one of the following forms:

- /dev/lp*n* for printers attached by parallel interfaces
- /dev/tty*nn* for printers attached by all serial interfaces

In both cases, *n* specifies the port number. If your printer is an LG01, LG02, LP27, or LP29 attached to an LP11 interface, the installation has already created an entry called /dev/lp0, located in the /dev directory. If your printer is not configured this way, you must create entries in the /dev directory. The following section describes how to do this.

4.3.3.3 Creating a Device Special File – To create a device special file in the /dev directory for each line printer, you run the shell script MAKEDEV for each logical device. Use the logical device name as the parameter passed to MAKEDEV. For example:

```
# cd /dev
# MAKEDEV lp1
```

In this case, MAKEDEV creates a device special file for lp1 named lp1.

You must run MAKEDEV for each logical device name that does not already exist. Use the ls command to verify whether the device name exists. For example,

```
# ls -l lp1
lp1 not found
```

For most serial line printers, the device name will be an existing terminal, which can then no longer be used as a logical terminal. The resulting files become named special files for known devices.

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

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

```
:lp=/dev/lp0
```

4.3.3.5 Defining the Spooling Directory – Each line printer must have its own spooling directory. Use the parameter sd to define the spooling directory to the /etc/printcap data base. For example,

```
:sd=/usr/spool/lpd
```

Remember, you must create the spooling directory itself, unless you use the default directory /usr/spool/lpd.

4.3.3.6 Defining the Baud Rate – The br parameter specifies the baud rate for the printer. The baud rate is dependent upon the printer hardware and can be found in your printer hardware manual. The baud rate applies only to serial printers and not to parallel printers.

```
:br#4800
```

This example defines the baud rate 4800 to the /etc/printcap data base.

4.3.3.7 Defining the Output Filters – Every printer requires an output filter program. The supported output filters reside in the /usr/lib/lpdfilters directory. See Section 4 of the ULTRIX Reference Pages for a detailed description of each supported print filter. The more commonly used print filters include:

la75of	Used to filter text data destined for the LA75 dot matrix printer. Handles printer device dependencies and performs accounting functions.
lg02of	Used to filter text data destined for the LG02 line printer. Also handles printer device dependencies and performs accounting functions.
ln01of	Used to filter text data destined for the LN01 and LN01S laser printers. Handles printer device dependencies and performs accounting functions.
ln01pp	Used to filter text data destined for the LN01S laser printer. This filter enables the pr(1) print function and sends the required control sequences to the printer that set the size of the printable area.
ln03of	Used to filter text data destined for the LN03 and LN03S laser printers. Handles the printer device dependencies and performs accounting functions.
ln03rof	Used to translate ASCII to Postscript format, enabling text data printing on Postscript printers. Also handles printer dependencies and performs some accounting functions.
lqf	Used to filter text data destined for various letter quality printers including: LQP02, LP25, LP26, LP27, LA50, LA75, LA100, LA120, LA210, and LG01. Handles printer device dependencies and performs accounting functions.
lpf	Used to filter text data destined for the following impact printers: LP25,LP26, LP27,LA50, LA75,LA100, LA120, LA210, and LG01. Handles printer device dependencies and performs accounting functions.

xf	Used to transparently concatenate data directly to the printer. This filter can be used with any printer; it allows device specific escape characters and control sequences to reach the printer.

You define the output filter by including its path name in the /etc/printcap file. For example, if you are entering the printer specifications for an LN01 printer at your site and intend to use the ln01of output filter, enter

```
:of=/usr/lib/lpdfilters/ln01of:
```

in the /etc/printcap file. See Figure 4-2 for a sample of /etc/printcap print filter entries.

4.3.3.8 Defining the Remote Printers – To access a remote printer, you must define the parameters lp, rm, rp, and sd.

The argument for lp is null. For example, lp=:. The argument for rm is the name of the remote host machine. For example, rm=nin:. The argument for rp is the name of the remote printer at that host. For example, rp=lp2:. As entries in the /etc/printcap file, these fields would be:

```
:lp=:
:rm=nin:
:rp=lp2:
```

In this example, nin is the remote host machine and lp2 is the remote printer at that host. You must obtain these names from the remote system manager. The argument for sd is the spooling directory. For example,

```
:sd=/usr/spool/lpd
```

You should check your system's /etc/hosts.lpd file, to ensure that it contains the names of the remote machines in your network. If the remote system name is not in /etc/hosts.lpd, include it, and confirm with the remote host's system manager that your system users have access to that printer.

Print requests to a remote system are accepted only if your system's name has been entered in the /etc/hosts.lpd file. Note that an asterisk (*) at the start of any line in /etc/hosts.lpd allows remote print requests from all systems. Similarly, to allow remote systems to access your system's printers, enter their names in your system's /etc/hosts.lpd file.

4.3.4 Using the /etc/printcap.examples File

The ULTRIX-32 software provides you with a file containing examples of printer definitions. This file, /etc/printcap.examples, assists you in setting up your individualized file. It consists of commented entries for various types of printers and print capabilities. In addition, it offers brief instructions for using this file when creating or modifying your system's /etc/printcap file.

To access /etc/printcap.examples, type:

```
# more /etc/printcap.examples
```

In response to your entry, the system displays an expanded version of the sample file shown here.

```
###
# /etc/printcap.examples: /etc/printcap templates.
#
# This file contains examples of line printer definitions
# you can use to tailor the printcap entries you need.
#
# The entries are now comments.  Copy this file to /etc/printcap.tmp.
# Tailor the entries you can use, and delete the entries you do not
# need. When you are done, copy this file to /etc/printcap or append
# this file to /etc/printcap.  Should you make a mistake you cannot
# recover from while editing this file, you can copy the replica file
# /etc/printcap.tmp to /etc/printcap.examples, and begin again.
###

###
# ln | a local LN01 printer on a DMF32 parallel interface:\
#    :dn=/usr/lib/lpd:\
#    :if=/usr/lib/lpdfilters/ln01of:\
#    :lf=/usr/adm/lnd-errs:\
#    :lp=/dev/lp:\
#    :mc#20:\
#    :mx#2000:\
#    :of=/usr/lib/lpdfilters/ln01of:\
#    :pl#60:\
#    :pw#80:\
#    :sd=/usr/spool/ln:
###
```

Figure 4-2 /etc/printcap.examples File Extract

4.3.4.1 Modifying the /etc/ttys File – If your printer is connected via a serial line, you must edit the /etc/ttys file as follows:

1. Log on as superuser.
2. Open /etc/ttys and locate the tty*nn* entry that corresponds to the serial printer device specification. For example, in Figure 4-3, the line that corresponds to the serial printer is identified as tty02.
3. Edit the /etc/getty entry by changing the "on" entry to "off" if this has not already been done.

```
#
#
## dmf0 has the following 8 lines:
#
tty00    "/etc/getty 2"    vt100    off  nomodem secure    # unused
tty01    "/etc/getty 2"    vt100    on   nomodem secure    # vt100 in lab
tty02    "/etc/getty 2"    vt100    off  nomodem secure    # printer (ln01b)
tty03    "/etc/getty 2"    vt100    off  nomodem           # unused - spare
tty04    "/etc/getty 2"    vt100    off  nomodem           # unused - spare
tty05    "/etc/getty 2"    vt100    off  nomodem           # unused - spare
tty06    "/etc/getty 2"    vt100    off  nomodem           # unused - spare
tty07    "/etc/getty 2"    vt100    off  nomodem           # unused - spare
#
#
```

Figure 4-3: Sample /etc/ttys Extract

4.4 Using the Automatic Printer Setup Program

The lprsetup program is an interactive program for setting up and modifying the line printers on your system. You can use this program when you need to add or delete a printer from your system or when you need to modify the information relating to an existing printer. After you have entered the specification, lprsetup creates the spooling directory, modifies /etc/ttys to prevent logins on the printer line, and creates or deletes an entry in /etc/printcap for the specification that you made.

4.4.1 Preparation

Access to the lprsetup program is restricted to the superuser. You must log on as superuser before invoking the program.

When working with the lprsetup program, you will need to specify information about your system. Although the lprsetup program sets default values and provides on-line help, it simplifies your task if you are familiar with the hardware and software specifications that lprsetup needs. These items include:

Printer number	Specify a number from 0 to 99. The lprsetup default value, shown in square brackets [*n*], indicates the number of printers currently specified in the /etc/printcap file, plus the entry you are about to make. For example, if you currently have 2 printer specifications in the /etc/printcap file, the program displays a default value of 3, indicating that you are adding a third printer specification.
Printer name	Specify the name of the printer type -- la100, ln01, or lp27, for example. You can specify the name for a printer on a local or remote system. lprsetup uses this name to formulate a default response for later questions. The program displays a list of printer types from which you can choose. The default value is 'unknown.'
Printer synonym	Specify the name by which you can identify the printer. For example, 'draft' could represent an lg01 line printer.
Device pathname	Specify the name of the special file to open for output. For example, use /dev/lp for a parallel printer. For a serial line printer, enter /dev/tty*nn*, where *nn* specifies the printer number or any valid double-digit number. The program defaults to the next consecutive number when setting up this file. For example, if this is the third device on your print system, the program supplies a default value of /dev/tty03.
Printer baud rate	Specify the printer baud rate for serial printers only. The default value differs according to the printer type. Your serial printer hardware manual lists the required baud rate.

Accounting file name	Specify the (optional) accounting file used to keep track of the number of pages printed by each user. There should be a unique file name for each printer. For example, the accounting file name for the third line printer on your system could be /usr/adm/lp3acct. The program supplies a default value of /usr/adm/*xxnn*acct where *xx* specifies the printer type and *nn* specifies the printer number.
Spooling directory	Specify the directory where files are queued before they are printed. Each spooling directory should be unique. For example, you could specify /usr/spool/lpd3 for the third line printer. The program supplies a default value of /usr/spool/lpd*n* where *n* specifies the printer number.
Error log file	Specify the (optional) log file where printer errors are reported. Each printer's log file should be unique. For example, you could use /usr/adm/lp3err for the third line printer at your site. The program defaults to /usr/adm/lp*n*err where *n* specifies the printer number.
Printcap symbol	Specify the (optional) printcap symbol you want to modify or add. The program provides a listing of available symbols and on-line help. Each of these symbols represents a field in the /etc/printcap file. See printcap(5) for a listing of the fields and options.
Output filter	Specify the (optional) output filter to be used with the printer. The choice of output filter depends on the printer type. The lprsetup program provides an on-line help listing of available filters. To get this listing, type 'of' for output filter when the program prompts for a symbol name. Then type **help** to get the on-line listing.

Output flags	Specify the system's treatment of the printer line. If your users are working with software that requires parity changes (7 to 8 bit or vice versa) or generates both carriage return and line feed characters, you should check and possibly add or modify the settings for the 'fc', 'fs', 'xc' or 'xs' printcap symbols. For a complete discussion of print system flags, see tty(4) in the ULTRIX Reference Pages.

4.4.2 Using lprsetup

When you are ready to set up or modify your printer configuration, and are familiar with the hardware as well as the items listed in the previous table, invoke the lprsetup program.

1. Enter the lprsetup command in response to the system prompt:

```
# lprsetup
```

   The program responds with a banner and menu. For example:

```
ULTRIX-32 Printer Setup Program
Command < add modify delete exit view quit help >:
```

2. Select the menu item that you need to set up or modify your print system. The following sections describe how to do this.

4.4.2.1 Adding a New Printer to Your System – Choose the first menu item, add, to add a new printer to your system. Type the entire word, **add**, or abbreviate your choice by typing **a**. In response to either entry, the program displays a series of prompts, one at a time, for the following information: printer name, printer type, device pathname, accounting file, spooler directory, error log file, and printer synonym. At each prompt, lprsetup provides a default value. To accept the default value, press the RETURN key. To request a different value, simply enter the non-default value.

4.4.2.2 Modifying an Existing Printer Specification – Choose the second menu item, modify, to modify an existing printer specification. Type the entire word, **modify**, or abbreviate your choice by typing **m**. In response to either entry, the program prompts for a printer name. Enter the synonym name of a printer that currently exists in the /etc/printcap file.

If you are unsure of the synonym name, open the /etc/printcap file and view the contents. Locate the entry you intend to modify and check the synonym name. Exit the /etc/printcap file, return to the lprsetup program, and enter the required information.

4.4.2.3 Deleting an Existing Printer Specification – Choose the third menu item, delete, to delete an existing printer specification. Type the entire word, **delete**, or abbreviate your choice by typing **d**. In response to either entry, the program prompts for a printer name. Enter the synonym name of the printer you want deleted from the etc/printcap file.

If you are unsure of the synonym name, open the /etc/printcap file and view the contents. Locate the entry you intend to delete and check the synonym name. Exit the /etc/printcap file, return to the lprsetup program, and enter the required information.

4.4.2.4 Exiting From the lprsetup Program – Choose the fourth menu item, exit, to leave the lprsetup program. Type the entire word, **exit**, or abbreviate your choice by typing **e**. In response to either entry, the program exits and the system displays its prompt.

4.4.2.5 Viewing the /etc/printcap File – Choose the fifth menu item, view, to look at the current /etc/printcap file for your system. Type the entire word, **view**, or abbreviate your choice by typing **v**. In response to either entry, the program opens the /etc/printcap file for you.

4.4.2.6 Quitting the Current Operation or lprsetup Program – Choose the sixth menu item, quit, to return to the lprsetup menu. This item is particularly helpful when you make an error and want to start again. Be aware, however, that choosing this item during another operation causes the program to ignore the entries made up to that point in the operation. Type the entire word, **quit**, or abbreviate your choice by typing **q**. In response to your entry, the program returns to the main menu.

4.4.2.7 Getting On-line Help – Choose the seventh menu item, help, to get on-line help while you are working with lprsetup. Initially, the **help** item provides general information about the lprsetup program itself. Typing **help** during another operation results in your getting additional information about the operation you are currently performing. Type the entire word, **help**, or abbreviate your choice by typing **h** or **?**. In response to your entry, the program displays information relevant to the current operation.

4.4.3 Sample LA100 Line Printer Settings

The following sample depicts the setup of an LA100 serial line printer. This sample uses the printer number 0 for the LA100 and a baud rate of 4800 bps as specified in the hardware manual. This printer is connected on line tty02 and uses the default accounting file, spooling directory, error log file, and flags set for an LA100 printer. Thus, the printer settings you will find in the /etc/printcap are:

```
Printer #0
Symbol  value
  af    /usr/adm/lpacct
  br    4800
  fc    0177777
  fs    023
  lf    /usr/adm/lperr
  lp    /dev/tty02
  mx    200
  of    /usr/lib/lpdfilters/lpf
  pl    66
  pw    80
  sd    /usr/spool/lpd
```

4.5 Controlling Print Jobs

Once installed with the proper software, the line printer spooler needs little maintenance. Nevertheless, there are times when you will need to control print jobs. To do this, you must understand the files and commands that the system uses. The files and commands are:

- /usr/lib/lpd
- /etc/lpc
- /usr/ucb/lpr
- /usr/ucb/lpq
- /usr/ucb/lprm
- /etc/pac

The following sections describe how to use these files and commands.

4.5.1 The lpd Line Printer Daemon

The line printer daemon, lpd, provides network communications of print requests. It also provides the selection and start of specific print filters for specific print requests. The print filters process the varying input formats into printer-specific output format.

The line printer daemon interface is a task that runs itself and always stays running, ready for input. This daemon is generally started at boot time from the /etc/rc file. The lpd command invokes the line printer daemon. When users submit print jobs with the lpr command, the line printer daemon schedules jobs to be printed and notifies printers that have jobs waiting.

When first invoked, the line printer daemon checks the spooler directory /usr/spool/lpd for the existence of a lock file. The lock file is a signal to lpd that another job is currently printing. If a lock file is not present, lpd creates one to reserve access to the printer for that particular print job. Once the daemon creates the lock file, it scans the directory of files beginning with cf. These files are control files which represent print jobs. For example:

```
% lpr memo.1
% ls -l /usr/spool/lpd
total 3
-rw-rw----    1 daemon         86 July 9 11:11 cfA024myvax
-rw-rw----    1 dmf          2358 July 9 11:11 dfA024myvax
-rw-r--r--    1 root            5 July 9 11:11 lock
-rw-rw-r--    1 root           52 July 9 11:11 status
```

The control file beginning with cf contains print instructions and the data file beginning with df contains the formatted text. The lock file contains the process ID of the currently running daemon, while the status file contains a line describing the current printer status. For further information, see lpd(8) in the ULTRIX Reference Pages.

4.5.2 Controlling Printer Activity

The lpc command allows you to control the activity of the line printers and spooler queues listed in /etc/printcap. You can use the lpc command to do the following:

- Enable/disable a printer
- Enable/disable a spooler queue
- Alter order of queued jobs
- Display printer, queue, or daemon status

Note

You must be superuser to enable or disable a printer or queue, or to alter the order of queued jobs.

When invoked without arguments, the lpc command prompts for commands and arguments from the standard input.

The lpc command may be invoked to change the order of jobs in the print queue. To place an important job at the top of the queue, invoke lpc, use the topq command with the printer and job number as arguments. Then exit the lpc procedure. For example:

```
# lpc
lpc> topq lp 60
lp:
     queue order changed
lpc> exit
```

For a complete list of lpc commands and arguments, see lpc(8) in the ULTRIX Reference Pages.

4.5.3 Printing a File

The lpr command queues and submits a job for printing. If you have a file named memo.1, you can print the file using:

```
% lpr memo.1
```

The print command paginates the job before printing. To do this type:

```
% print memo.1
```

If you pipe the file to lpr, the file name is listed as standard input. For example:

```
% cat memo.1 | lpr
```

If no files are named, the standard input is read.

For further information, see lpr(1) in the ULTRIX Reference Pages.

4.5.4 Checking the Print Queue

The lpq command displays the current contents of the line printer queue and lists the jobs that have not yet printed. For example:

```
% lpq
lp is ready and printing
Rank      Owner     Job    Files                 Total Size
active    dmf       24     memo.1                23056 bytes
1st       dmf       25     (standard input)       6987 bytes
```

There are two jobs in the print queue belonging to user dmf. The active job is number 24, memo.1. The lpq command displays information in the order in which it is scheduled to print. For further information, see lpq(8) in the ULTRIX Reference Pages.

4.5.5 Removing a Job From the Queue

The lprm command allows you to remove a job from a queue. To locate the job number, and then remove a print job, type:

```
% lpq
lp is ready and printing
Rank   Owner   Job   Files              Total Size
active dmf     24    memo.1             23056 bytes
1st    dmf     25    /etc/printcap       6987 bytes
% lprm 24
dfA024myvax dequeued
cfA024myvax dequeued
```

When used without arguments lprm deletes the currently active job, if it is owned by you or if you are the superuser. If invoked with a user's name, it removes all print jobs owned by that user. For further information, see lprm(1) in the ULTRIX Reference Pages.

4.5.6 Generating a Report of Printer Use

Periodically, you should generate a report of your printer use. The pac command can be used only if you have specified an accounting file for each printer for which a report is wanted. To generate this report, use the pac command. For example:

```
# /etc/pac
```

This command displays a report detailing number of pages printed, feet of paper consumed, and total estimated cost per user. For further information, see pac(8) in the ULTRIX Reference Pages.

Adding and Deleting Software Subsets 5

This chapter describes how to use the setld program to list, load, add, delete, and verify software subsets. You can save disk space with the setld program by specifying and loading only the software subsets that you need.

For detailed information on the setld command, its options, and the command line syntax, see setld(8) in the ULTRIX Reference Pages.

Note

You must be the superuser to run the setld program.

5.1 Using setld Options

The setld command has specific options for listing, loading, adding, deleting, and verifying software subsets. Although each option is different, the setld command line syntax has certain common arguments, namely:

- The optional *path* variable specifies the destination of the subset. Specify this variable if you are loading the software to a file hierarchy that starts somewhere other than at root (/).
- The optional *subset* variable specifies the name of the subset.
- The *loc-code* variable specifies the device special file or mount point containing the media from which the subset or product is to be transferred.

If you specify a *path* variable, it must precede the command option. However, if you specify a *subset* or *loc_code* variable, it must follow the command option.

Note

All NFS file systems should be dismounted before using the setld command with the -a, -d, and -l options.

5.1.1 Listing Software Subsets

To get a listing of software subsets with the status of each and its description, use the setld command with the -i (inventory) option. The syntax is:

```
# /etc/setld [path] -i [subset]
```

For example, to display a list of subsets for a system running ULTRIX-32 Version 3.0, type:

```
# /etc/setld -i
```

You get a listing similar to this:

```
Subset      Status       Description

ULTACCT030               Accounting Software
ULTBIN030   installed     Kernel Configuration Files
ULTBASE030  installed     Base System
ULTCOMM030               Communications Utilities
ULTDCMT030               Document Preparation Software
ULTEXER030  installed     System Exerciser Package
        .
        .
        .
UWSFONT020  installed     X11/DECWindows Fonts
UWSX11020   installed     X11/DECwindows User Environment
```

To display the files included in a particular subset -- ULTBASE030, for example, type:

```
# /etc/setld -i ULTBASE030
```

You get a listing similar to this:

```
./vmb.exe
./usr/var/tmp
./usr/var/spool/rje
./usr/var/spool/mqueue/syslog
    .
    .
    .
./.cshrc
```

5.1.2 Loading Subsets

To load a software product to your system for the first time, use the setld command with the -l (load) option. The syntax is:

```
# /etc/setld [path] -l <loc_code>
```

In response, the setld program loads the mandatory subsets and the optional subsets selected during the installation.

For example, assume you want to load the available software subsets from the installation media located at drive 1 on a TK50 device to the hierarchy that has its root at /mnt. To do this, type:

```
# /etc/setld /mnt -l /dev/rmt1h
```

5.1.3 Adding Subsets

To add a subset to your system, use the setld command with the -a (add) option. The syntax is:

```
# /etc/setld [path] -a <loc_code> subset [subset...]
```

For example, assume you want to reinstall the UWSX11020 subset on your system. The software currently resides at /dev/rmt1h and you want to add it to your directory at /mnt. To do this, type:

```
# /etc/setld /mnt -a /dev/rmt1h UWSX11020
```

Note

Use the -a option only if the software product or individual subset has been loaded previously. See the preceding section, Loading Subsets, for instructions on loading subsets for the first time.

5.1.4 Deleting Subsets

To delete a subset from your system, use the setld command with the -d option. The syntax is:

```
# /etc/setld [path] -d subset [subset...]
```

For example, assume you want to delete the UUCP subset from your system. The subset currently resides at /mnt. To delete this subset, type:

```
# /etc/setld /mnt -d ULTUUCP030
```

5.1.5 Verifying Subset Installations

To verify that a subset is installed correctly, use the setld command with the -v option. The syntax is:

```
# /etc/setld -v [subset]
```

For example, to verify that the Communications Utilities subset has installed correctly, type:

```
# setld -v ULTCOMM030
```

If the subset is installed, you get this message:

```
Checking setld data files.
```

However, if the subset is not installed, you get this message:

```
Checking setld data files.

ULTCOMM030:  not currently installed
```

If you do not specify the optional *subset* variable, the setld program verifies the correctness of all installed subsets.

Monitoring and Managing System Performance 6

This chapter provides guidelines that you can use to monitor and to manage system performance. The chapter points out the system-level reports that you can produce to assist you in monitoring the system. It also offers guidelines for particular system management and optimization tasks.

Specifically, the chapter discusses:

- Managing system scheduling priority
- Generating system accounting information
- Checking Interprocess Communications Facilities Status

6.1 Managing Process Scheduling Priority

You can manage the system's process scheduling priority of a given process using the renice command. For example:

```
# /etc/renice +5 -u name
```

In this example, the renice command lowers the scheduling priority of the specified running processes, where name is the login ID of the user.

As the superuser, you can raise the scheduling priority of a user's processes by using a negative number instead of a positive number. For example:

```
# /etc/renice -5 -u name
```

You can raise or lower process scheduling priority on a scale from +20 to -20. For further information, see renice(8) in the ULTRIX Reference Pages.

6.2 Generating System Accounting Information

There are two types of system accounting information: accumulated and archived. During daily operations, system accounting information is accumulated so that you can keep track of day-to-day operations. With this information, you can keep track of:

- User logins
- Command usage
- Printer usage

This section also describes commands used to display archived system statistics.

6.2.1 Generating User Log-In Report

The system automatically maintains two log-in accounting files: /etc/utmp and /usr/adm/wtmp. The system records all active logins in /etc/utmp and accumulates a user log-in history in /usr/adm/wtmp.

You can generate a report of the system's login-history with the ac command:

```
# /etc/ac -p
```

Over time, /usr/adm/wtmp increases in size. After you generate a hardcopy of the file you should clear it. To clear the /usr/adm/wtmp file, use the cp command with the arguments:

```
# cp /dev/null /usr/adm/wtmp
```

This command copies /dev/null to /usr/adm/wtmp. That is, it reduces /usr/adm/wtmp to a zero-length file.

Note

The system automatically enables log-in history, but it accumulates a log-in history only if /usr/adm/wtmp exists. To disable the system log-in history, remove /usr/adm/wtmp.

For further information, see cp(1) and ac(8).

6.2.2 Generating Command Usage Report

During multiuser startup, /etc/rc normally enables system process accounting. When process accounting is enabled, the system records information on each executed process (command) in /usr/adm/acct. In some systems, system process accounting may be disabled to save disk space.

You can display the contents of the system's current process accounting file, /usr/adm/acct using the sa command. For example:

```
# /etc/sa
```

This report shows which commands are being used most often on the system.

The file /usr/adm/acct increases in size depending upon your system's activity. To manage space on your /usr file system, you should condense the process accounting information as necessary. To condense /usr/adm/acct, use the sa command with the -s option specified. For example:

```
# /etc/sa -s
```

This command merges the current information in /usr/adm/acct into the process history file, /usr/adm/savacct.

Note

To disable process accounting immediately, type:

```
# /etc/accton
```

To disable process accounting the next time the system reboots, comment out this line in the /etc/rc file by putting a # in the first column of the line on which the statement appears. This makes the accton line a comment which is not executed. For example:

```
# /etc/accton /usr/adm/acct; echo -n ' accounting' > /dev/console
```

For further information, see sa(8) in the ULTRIX Reference Pages.

6.2.3 Generating Printer Usage Report

Your system normally records all printer information in the default accounting file named in /etc/printcap.

To generate a report of your printer usage, use the pac command. For example:

```
# /etc/pac
```

The pac command displays a report detailing usage per user: number of pages printed, feet of paper consumed, and total estimated cost.

Note

The system enables printer accounting only if /etc/printcap names a default accounting file. For further information, see printcap(5) in the ULTRIX Reference Pages.

For further information, see pac(8) in the ULTRIX Reference Pages.

6.2.4 Generating Active System Report

In addition to those commands that are used to display accumulated system accounting information, the system has a number of commands that you can use to display active system statistics. For example:

- The iostat(1) command displays a report of current I/O statistics.
- The ps(1) command displays a report of the system's process status.
- The uptime(1) command displays a report of how long the system has been up.
- The vmstat(1) command displays a report of virtual memory statistics.
- The w(1) command displays a report of currently active users and what they are doing.
- The pstat(8) command displays various system tables.
- The netstat(1) command displays network activity.
- The nfsstat(8nfs) command displays activity on the Network File System (NFS).

For further information on these commands, see the ULTRIX Reference Pages.

6.3 Check Interprocess Communications Facilities Status

Some of the interprocess communications (IPC) facilities requested and used by various processes are not automatically released when the processes exit. You should alert system users to release the facilities after the processes are finished with them. The users can either include an explicit system call in the program to release the facilities, or they can release the facilities using the ipcrm command after the process exits. The facilities that must be released are shared memory, semaphores, and message queues.

If the users release the facilities properly, then there is no need to administer them. However, if the number of available message queues, shared memory, or semaphores becomes exhausted, then use the ipcs command to see the status of these resources. If necessary, you can release them by using the ipcrm command. However, make sure that the users are finished with the resources before you release them. You should also show the users how to clean up the resources after having finished using them.

This example shows how to use the ipcs command and gives a sample output:

```
# ipcs
IPC status from /dev/kmem as of Fri Jul 19 07:36:20 1985
Message Queues:
T     ID     KEY        MODE       OWNER    GROUP
*** No message queues are currently defined ***

Shared Memory
T     ID     KEY        MODE       OWNER    GROUP
m    400  1627788395  --rw-------   gdp      staff

Semaphores
T     ID     KEY        MODE       OWNER    GROUP
s      2  1644565427  --ra-------   gdp      staff
```

You must be logged in as superuser to release facilities owned by another user. To release the semaphore found in the example type:

```
# ipcrm -s 2
```

The **-s** indicates that a semaphore is to be released, and **2** is the unique ID of the semaphore that was reported in the **ipcs** output. For further information, see **ipcrm**(1) and **ipcs**(1) in the ULTRIX Reference Pages.

Device Mnemonics A

This appendix identifies and defines the mnemonics that are used to attach any hardware or software device to your system. The mnemonics are used by the /dev/MAKEDEV shell script to create the character or block special files that represent each of the devices. The mnemonics also appear in the system configuration file as described in the Guide to System Configuration File Maintenance.

Table A–1 lists the mnemonics in seven categories: generic, consoles, disks, tapes, terminals, modems, and printers. The generic category lists the mnemonics of a general nature and includes memory, null, trace, and tty devices. The consoles category lists the system console devices that the ULTRIX operating system uses. The disks, tapes, terminals, modems, and printers categories identify the appropriate mnemonics for those devices.

The description heading in Table A–1 identifies the corresponding device name. It does not define the mnemonic's use. For detailed information on the use of each mnemonic in relation to both the MAKEDEV script and the system configuration file, refer to the reference pages in Section 4 of the ULTRIX Reference Pages. If on-line reference pages are available, you can also use the man command. For instance, if you enter at the system prompt:

```
# man ra
```

the system displays the reference page for the Mass Storage Control Protocol (MSCP) disk controller driver. Where appropriate, the SYNTAX section of the reference page defines the device's syntax as it appears, or should appear, in the config file. Refer to /dev/MAKEDEV for additional software device mnemonics that MAKEDEV uses. Refer to MAKEDEV(8) in the ULTRIX Reference Pages for a description of the MAKEDEV utility.

You should note that Table A–1 uses the convention of an asterisk (*) beside a mnemonic and a question mark (?) beside a device name to mean a variable number. The range of the variable number is dependent on the particular device.

Table A-1: Devices Supported by MAKEDEV

Category	Mnemonic	Description
Generic	boot*	Boot and std devices by cpu number; e.g., boot750
	mvax*	All MicroVAX setups; e.g., mvax2000
	vaxstation*	A VAXstation 2000 setup; e.g., vaxstation2000
	std	Standard devices below with all console subsystems:
	drum	Kernel drum device
	errlog	Error log device
	kUmem	Kernel Unibus/Q-bus virtual memory
	kmem	Virtual main memory
	mem	Physical memory
	null	A null device
	trace	A trace device
	tty	A tty device
	local	Customer specific devices
Consoles	console	System console interface
	crl	Console RL02 disk interface for VAX 86?0
	cs*	Console RX50 floppy interface for VAX 8??0
	ctu*	Console TU58 cassette interface for VAX 11/750
	cty*	Console extra serial line units for VAX 8??0
	cfl	Console RX01 floppy interface for 11/78?
	ttycp	Console line used as auxiliary terminal port
Disks	hp*	MASSBUS disk interface for RM?? drives
	ra*	UNIBUS/Q-bus/BI/HSC MSCP disk controller interface
	ese*	UNIBUS/Q-bus/BI/HSC MSCP electronic ESE20 disk
	rb*	UNIBUS IDC RL02 disk controller interface for RB?? drives
	rd*	VAXstation 2000 and MicroVAX 2000 RD type drives
	rz	SCSI disks (RZ22/RZ23/RZ55/RRD40)
	rk*	UNIBUS RK?? disk controller interface
	rl*	UNIBUS/Q-bus RL?? disk controller interface
	rx*	VAXstation 2000 and MicroVAX 2000 RX type drives
Tapes	mu*	TU78 MASSBUS magtape interface
	tms*	UNIBUS/Q-bus/BI/HSC TMSCP tape controller interface
	rv*	UNIBUS/Q-bus/BI/HSC TMSCP optical disk
	ts*	UNIBUS/Q-bus TS11/TS05/TU80 magtape interface
	tu*	TE16/TU45/TU77 MASSBUS magtape interface
	st*	VAXstation 2000 and MicroVAX 2000 TZK50 cartridge tape

Category	Mnemonic	Description
	tz*	SCSI tapes (TZ30/TZK50)
Terminals	cxa*	Q-bus cxa16
	cxb*	Q-bus cxb16
	cxy*	Q-bus cxt08
	dfa*	Q-bus DFA01 comm multiplexer
	dhq*	Q-bus DHQ11 comm multiplexer
	dhu*	UNIBUS DHU11 comm multiplexer
	dhv*	Q-bus DHV11 comm multiplexer
	dmb*	BI DMB32 comm multiplexer including dmbsp serial printer/plotter
	dhb*	BI DHB32 comm multiplexer
	dmf*	UNIBUS DMF32 comm multiplexer including dmfsp serial printer/plotter
	dmz*	UNIBUS DMZ32 comm multiplexer
	dz	UNIBUS DZ11 and DZ32 comm multiplexer
	sh*	MicroVAX 2000, 8 serial line expansion option
	ss*	VAXstation 2000 and MicroVAX 2000 basic 4 serial line unit
	dzq*	Q-bus DZQ11 comm multiplexer
	dzv*	Q-bus DZV11 comm multiplexer
	lta*	Sets of 16 network local area terminals (LAT)
	pty*	Sets of 16 network pseudoterminals
	qd*	Q-bus VCB02 (QDSS) graphics controller/console
	qv*	Q-bus VCB01 (QVSS) graphics controller/console
	sm*	VAXstation 2000 monochrome bitmap graphics/console
	sg*	VAXstation 2000 color bitmap graphics console
Modems	dfa*	DFA01 integral modem communications device.
Printers	dmbsp*	BI DMB32 serial printer/plotter
	dmfsp*	UNIBUS DMF32 serial printer/plotter
	lp*	UNIBUS LP11 parallel line printer
	lpv*	Q-bus LP11 parallel line printer

Index

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
and New Hampshire,
Alaska or Hawaii
call **800-DIGITAL**

In Canada
call **800-267-6215**

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.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
PSG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01473

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809–754–7575

ULTRIX - 32
Guide to System
Environment Setup
AA-ME89A-TE

Reader's Comments

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. _____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

Zip Code

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

Digital Equipment Corporation
Documentation Manager
ULTRIX Documentation Group
ZKO3-3/X18
Spit Brook Road
Nashua, N.H.
03063

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line

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 _____

Zip Code

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

Digital Equipment Corporation
Documentation Manager
ULTRIX Documentation Group
ZKO3-3/X18
Spit Brook Road
Nashua, N.H.
03063

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line