



Server Administration With A/UX 3.0.1

Advanced Concepts and Procedures

 Apple Computer, Inc.

This manual and the software described in it are copyrighted, with all rights reserved. Under the copyright laws, this manual or the software may not be copied, in whole or part, without written consent of Apple, except in the normal use of the software or to make a backup copy of the software. The same proprietary and copyright notices must be affixed to any permitted copies as were affixed to the original. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased (with all backup copies) may be sold, given, or loaned to another person. Under the law, copying includes translating into another language or format.

You may use the software on any computer owned by you, but extra copies cannot be made for this purpose.

The Apple logo is a registered trademark of Apple Computer, Inc. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

© Apple Computer, Inc., 1993
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010

Apple, the Apple logo, AppleShare, AppleTalk, A/UX, EtherTalk, ImageWriter, LaserWriter, LocalTalk, Macintosh, and MPW are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Apple Desktop Bus, Balloon Help, DocViewer, Finder, Macintosh Quadra, and MacX are trademarks of Apple Computer, Inc.

Adobe, Adobe Illustrator, and PostScript are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

MacWrite and MacPaint are registered trademarks of Claris Corporation.

Linotronic is a registered trademark of Linotype Company.

Microsoft is a registered trademark of Microsoft Corporation.

NFS is a trademark of Sun Microsystems, Inc.

NuBus is a trademark of Texas Instruments.

QuarkXPress is a registered trademark of Quark, Inc.

Retrospect Remote is a registered trademark of Dantz Development Corporation.

UNIX is a registered trademark of UNIX System Laboratories, Inc.

VT100 and VT102 are trademarks of Digital Equipment Corporation.

X Window System is a trademark of the Massachusetts Institute of Technology.

Yellow Pages is a registered trademark of British Telecommunications plc.

Simultaneously published in the United States and Canada.

Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products.

Contents

Preface: About This Guide / xix

1 Using CommandShell / 1

Introduction to CommandShell / 2

Making CommandShell active / 2

 Switching between CommandShell and the Finder / 3

Using CommandShell windows / 4

 Displaying CommandShell windows / 4

 Arranging CommandShell windows / 4

 Recording the contents of a window / 5

 Canceling the recording of window contents / 7

 Erasing recorded window contents / 7

 Closing CommandShell windows / 7

 Closing a window / 7

 Closing all open windows / 8

Printing the contents of a CommandShell window / 9

Using the A/UX System Console / 9

 Displaying the A/UX System Console window / 10

 Clearing the A/UX System Console window / 10

 Closing the A/UX System Console window / 10

 How A/UX alerts you to a message / 11

Customizing CommandShell windows / 12

 Presetting window defaults / 12

 Presetting the appearance of new windows / 13

 Saving the layout of your CommandShell windows / 14

 Restoring the window layout to the saved preferences / 15

Changing shells / 16

- Using Macintosh editing tools in CommandShell windows / 16
 - Selecting text / 16
 - Copying text / 17
 - Pasting text / 17
 - Changing the font or font size / 17
 - Saving a selection in a new file / 18
- Leaving CommandShell / 19

2 An Introduction to Using UNIX Commands / 21

- UNIX pathnames / 22
- Introduction to UNIX commands / 26
 - Using UNIX commands to navigate in the file system / 26
 - Finding your current directory / 27
 - Moving to a different directory / 27
- Options and arguments / 28
 - Options / 28
 - Listing the contents of a directory: The `ls` command options / 28
 - Arguments / 31
 - Using the `ls` command with an argument / 31
- Further useful UNIX operations / 32
 - Searching for a string of characters and redirecting output / 32
 - Looking up UNIX commands on line / 34
 - Moving down through the manual page display / 35
 - The “Name” section / 35
 - The “Synopsis” section / 35
 - The “Arguments” section / 36
 - The “Description” section / 36
 - The “Examples” section / 36
 - The “Files” and “See Also” sections / 36
 - Quitting a manual page / 37
 - Linking commands with a pipe / 37

3 System Startup / 41

About A/UX Startup / 42

Entering A/UX Startup / 43

A/UX Startup shell window / 44

Commands that run in A/UX Startup / 45

A/UX Startup menus / 46

Entering A/UX Startup on every startup / 46

Starting A/UX from A/UX Startup / 47

To interrupt a program running in A/UX Startup / 47

The autorecovery feature in A/UX Startup / 47

The autolaunch feature in A/UX Startup / 47

Starting up A/UX from a nondefault hard disk / 48

Changing the initial directory / 49

Checking your parity memory / 49

Protecting A/UX Startup with a password / 49

The startup sequence / 50

Customizing your system / 53

Naming the system / 53

Setting the system time / 54

Single-user and multi-user modes / 55

Single-user mode / 55

Multi-user mode / 56

The Autologin file / 57

Initial processes: `/etc/inittab` / 57

Changing run levels: `init` / 58

Configuring the kernel / 59

The `newconfig` command / 60

Changing buffer cache size / 62

Changing kernel parameters / 63

Logging out, restarting, and shutting down / 65

4 Backup Media and Devices / 67

Introduction / 68

Retrospect Remote limitations / 68

Backup media / 69

When to use paper / 69

Backing up files by dragging them to a floppy disk / 69

Backing up files using tapes / 70

Backup-device filenames / 71

Preparing and formatting a tape cartridge / 73

Selecting tapes for use with the internal tape drive / 74

Cleaning your tape drive / 74

5 Managing Disks / 75

Mounting file systems with UNIX commands / 76

Unmounting file systems / 77

If you change a disk's bus or SCSI ID / 77

Changing the name of a mounted volume / 78

Using an Apple CD-ROM drive / 79

Identifying the file system on a CD-ROM disc / 79

Using CD-ROM discs containing Macintosh file systems / 80

Using CD-ROM discs containing ISO file systems / 80

Using CD-ROM discs containing UNIX file systems / 80

Installing CD-ROM discs permanently on an A/UX file system / 81

Installing CD-ROM discs temporarily on an A/UX file system / 82

Opening files / 82

Opening files from the Finder / 82

Opening files from CommandShell / 83

Copying files to disk / 83

Files in a UNIX file system / 84

Files in a tar archive / 84

Files in a cpio archive / 86

Removing the CD-ROM discs / 86

If the CD-ROM was mounted permanently / 87

If the CD-ROM was mounted temporarily / 87

- Reclaiming disk space / 88
 - Removing unneeded files / 88
 - Trimming files that grow / 89
 - Compressing infrequently used files / 90
 - Usage notes / 91
 - Extensions to names of compressed files / 92
 - Compressing an archive of files / 92
- Automating server administration with the `cron` command / 92
 - User access to `cron` / 93
 - The `crontab` file format / 93
 - Adding `cron` commands / 94
 - Changing `cron` commands / 95
 - Removing `cron` commands / 96
- Using floppy disks that contain data in a non-Macintosh format / 96

6 Creating and Managing a TCP/IP Network / 99

- Overview of the TCP/IP networking software / 100
 - Using other network resources / 100
- Configuration parameters / 101
 - Choosing your host name / 101
 - Establishing your network number and IP address / 101
 - Determining the broadcast address for your network / 104
 - Determining the netmask for your network / 105
- Creating a simple network / 105
 - Adding the first computer to a two-computer network / 106
 - Adding the second computer to the two-computer network / 109
 - Testing the networking software / 110
 - Testing the loopback interface / 110
 - Using the `ping` command / 110
 - Logging in to another computer / 112
- Changing a network configuration parameter / 112
 - Modifying the `/etc/HOSTNAME` file / 113
 - Modifying the `/etc/NETADDRS` file / 113

- Other related files / 114
 - Modifying the `/etc/inittab` file / 114
 - Modifying the `/etc/hosts` file / 115
 - Modifying the `/etc/ftpusers` file / 116
 - Modifying the `/etc/networks` file / 116
 - Modifying the `/etc/printcap` file / 117
 - Modifying the `/etc/protocols` file / 117
 - Modifying the `/etc/servers` file / 117
 - Modifying the `/etc/services` file / 118
 - Modifying the `/etc/shells` file / 118
- TCP/IP network security issues / 118
 - Making your system more secure from network access / 119
 - Making your system easier to access from the network / 120
 - The `$HOME/.rhosts` file / 121
- Designing a complex network / 121
 - Routing and forwarding / 122
 - Subnets / 127
 - Subnets and broadcast addresses / 132
- Administering a TCP/IP network / 132
 - Determining network status / 132
 - Finding out what computers are running and who is logged in / 133
 - Determining if a specific host is up / 134
 - Determining that the IP address is correct / 135

7 Printing From CommandShell / 137

- Printer connections / 138
- Peripheral device drivers / 139
- Ports / 140
- Managing the `lpr` print spooler / 141
 - Definitions / 141
 - Setting up the print spooler / 143
 - The `printcap` database / 143
 - `lpr` commands / 147
 - Commands for general use / 147
 - Commands for line-printer administrators / 148

Setting up a nondefault printer in the `lpr` system / 150

Writing printer output filters / 151

Printing files / 152

Sending a file to a printer connected directly or through AppleTalk / 152

Sending a file to a remote UNIX printer / 153

Finding out how many files are waiting in the printing queue / 153

Canceling a printing job / 154

8 User and Group Administration for A/UX Users / 155

The A/UX working environment / 156

An introduction to the working environment / 156

New environment variables in A/UX 3.0.1 / 159

Shells and setup files / 160

The command-line prompt / 160

The C shell setup files / 161

The Bourne shell setup files / 161

The Korn shell setup files / 162

Using System Folders to customize environments / 162

Administrative login accounts and groups / 162

Administrative login accounts / 163

Administrative groups / 164

Administering user accounts / 165

The `/etc/passwd` file / 165

Format of `/etc/passwd` / 165

Adding a user / 167

Planning a user's working environment / 167

Adding users / 169

Removing users / 171

Voiding a user account / 171

Deleting a user account / 172

Moving a user's home directory within a file system / 173

Changing a user's shell / 173

Administering groups / 174

Creating groups / 174

The `/etc/group` file / 175

Multiple group memberships / 176

- Files, directories, folders, and permissions / 177
 - File permissions / 177
 - Directory and folder permissions / 178
 - Changing file permissions: `chmod` / 179
 - Symbolic terms / 179
 - Numeric terms / 181
 - Setting command permissions using `setuid` and `setgid` bits / 182
 - The sticky bit / 184
 - Setting default file permissions with `umask` / 184

9 Troubleshooting / 187

- Troubleshooting guidelines / 188
- Recovering from a system crash / 188
- Trouble while starting up A/UX / 189
- Accessing the A/UX file system from A/UX Startup / 192
- Reinitializing a disk with bad sectors / 193
- Screen locks, power failures, and emergencies / 194
- Troubleshooting user and group administration / 195
- Errors while changing your password / 197
- Troubleshooting the `lpr` system / 198
 - `lpr` error messages / 199
 - `lpq` error messages / 200
 - `lprm` error message / 201
 - `lpd` error messages / 201
 - `lpc` error messages / 201
- File-system problems / 202
- Kernel parameter problems / 204
- Other issues / 204
- Troubleshooting network problems / 206
 - Debugging the network hardware / 206
 - Checking transceivers / 206
 - Checking terminators / 206
 - Swapping hardware / 207

- Analyzing TCP/IP network software problems / 207
 - Analyzing network traffic / 207
 - Analyzing routing errors / 209
- Responding to common TCP/IP error messages / 209

10 Checking Your Data Consistency / 213

- Introduction to `fsck` / 214
- Overview of the A/UX file-system interface / 215
 - Blocks and bytes / 215
 - Inodes / 216
 - Direct and indirect blocks / 217
 - Additional information contained in inodes / 219
 - An example of inode and file access / 220
 - The superblock / 221
 - Block I/O / 221
 - The kernel buffer cache / 222
 - The `/dev` directory / 222
- How `fsck` works / 225
 - File-system updates / 225
 - Superblock / 225
 - Inode / 226
 - Indirect blocks / 226
 - Data block / 226
 - Free list / 227
 - `fsck` phases / 227
 - Phase 1: Check blocks and sizes / 227
 - Phase 2: Check pathnames / 227
 - Phase 3: Check connectivity / 227
 - Phase 4: Check reference counts / 228
 - Phase 5: Check cylinder groups / 228
 - `lost+found` directory / 228
- Using `fsck` / 228
 - When to use `fsck` / 228
 - `fsck`: A sample interaction / 230
 - Multiple file systems and `fsck` / 231

- fsck error messages / 234
 - fsck initialization error messages / 235
 - Phase 1: Check blocks and sizes / 240
 - Phase 1B: Rescan for more duplicates / 243
 - Phase 2: Check pathnames / 244
 - Phase 3: Check connectivity / 252
 - Phase 4: Check reference counts / 255
 - Phase 5: Check cylinder groups / 258
 - Cleanup / 260

Appendix A CommandShell Menu Items / 261

Activating CommandShell / 262

Activating the Finder / 262

The CommandShell environment / 263

The CommandShell File menu / 263

- New / 263
- Open... / 263
- Close / 263
- Save Selection... / 263
- Page Setup... / 264
- Print Selection... / 264
- Close All Windows / 264
- Quit / 264

The CommandShell Edit menu / 264

- Undo / 265
- Cut / 265
- Copy / 265
- Paste / 265
- Clear / 265
- Select All / 265
- Commando / 265

The CommandShell Window menu / 266

- Tile / 267
- Tile Horizontal / 268
- Tile Vertical / 268
- Standard Positions / 269
- Standard Size / 270
- Full Height / 270

- Zoom Window / 270
- Hide “Window Name” / 270
- Show All Windows / 270
- Previous Window / 270
- Rotate Windows / 271
- A/UX System Console / 271
- Names of all accessible CommandShell windows / 271
- The CommandShell Fonts menu / 271
- The CommandShell Commands menu / 272
 - Don’t Record Lines Off Top / 272
 - Clear Lines Off Top / 272
 - Redraw Screen / 272
- The CommandShell Preferences menu / 273
 - Notification Levels / 273
 - New Window Settings / 273
 - Active Window Settings / 273
 - Save Preferences / 273
 - Restore From Preferences / 273
- The Keys menu / 274
 - Keypad / 274
 - Cursor / 274
 - No Scroll / 274

Appendix B Network Contacts / 275

- The Internet / 276
- CSNET / 277
- BITNET / 277

Appendix C Information on Files in A/UX 3.0.1 / 279

- About A/UX files on your disk / 280
- A/UX files compared with UNIX files / 280

Glossary / 291

Index / 297

Figures and Tables

1 Using CommandShell

Figure 1-1 Windows in standard positions / 5

Figure 1-2 Recording window contents as they scroll off the top / 6

Figure 1-3 Viewing recorded window contents / 6

2 An Introduction to Using UNIX Commands

Figure 2-1 The pathname `/users/start/letter1` / 23

Figure 2-2 The relative pathname `start/letter1` / 25

Figure 2-3 Using a pipe to channel the output of Command 1 as the input to Command 2 / 37

3 System Startup

Figure 3-1 The startup screen from which to enter A/UX Startup / 43

Figure 3-2 A/UX System Console window with startup information / 43

Figure 3-3 A/UX Startup window with notification of file-system problems / 44

Figure 3-4 Booting dialog box / 46

Figure 3-5 General dialog box / 48

Figure 3-6 A/UX Startup password dialog box / 49

Figure 3-7 The Checking startup screen / 50

Figure 3-8 The Loading startup screen / 51

Figure 3-9 The Launching startup screen / 51

Figure 3-10 The “Checking root file system” startup screen / 51

Figure 3-11 The “Initializing device drivers” startup screen / 52

- Figure 3-12 The “Starting background processes” startup screen / 52
Figure 3-13 Memory control panel showing buffer cache adjustments / 63

4 Backup Media and Devices

- Table 4-1 Standard A/UX device files / 72

6 Creating and Managing a TCP/IP Network

- Figure 6-1 A two-computer network / 106
Figure 6-2 Two separate networks / 123
Figure 6-3 Two networks with `hostname2` acting as an Internet forwarder / 124
Figure 6-4 Two subnets with a netmask of `0xffffffff0` / 130
Table 6-1 Sample IP addresses / 102
Table 6-2 The parts of an IP address / 103
Table 6-3 Sample broadcast addresses / 104
Table 6-4 The default netmasks / 105
Table 6-5 Values of *interface-ID* in the `/etc/NETADDRS` file / 114
Table 6-6 The configuration files of computers on the sample network / 127
Table 6-7 Breakdown of a subnet netmask for each network class / 129
Table 6-8 The range of IP addresses for a class C subnetted network / 129
Table 6-9 Network configuration files on the sample subnets / 131
Table 6-10 Netmasks and broadcast addresses for subnets / 132

7 Printing From CommandShell

- Figure 7-1 Remote printing from CommandShell / 139
Figure 7-2 Signals on a serial port / 140
Figure 7-3 The printing queue / 142
Table 7-1 Printing with A/UX / 138

8 User and Group Administration for A/UX Users

Figure 8-1 Access classes / 177

Figure 8-2 Numeric values for `chmod` / 181

Table 8-1 File-type characters / 178

10 Checking Your Data Consistency

Figure 10-1 Inumber relationships / 217

Figure 10-2 Indirect blocks on a 4K file system / 218

Figure 10-3 File-directory connection through inodes / 220

Figure 10-4 A description of sample entries in `/etc/fstab` / 232

Figure 10-5 How `fsck` decides whether to check a file system / 233

Appendix A CommandShell Menu Items

Figure A-1 A CommandShell window / 262

Figure A-2 Tiled windows / 267

Figure A-3 Horizontally tiled windows / 268

Figure A-4 Vertically tiled windows / 269

Figure A-5 Standard window positioning / 269

Preface

About This Guide

This book explains how to use A/UX Version 3.0.1 to perform administration tasks for the Apple Workgroup Server 95. Usually one person is responsible for maintaining and modifying the way the system operates; this person is called the *server administrator*. The server administrator maintains the system using the privileged access account called `root`.

The tasks of the administrator vary widely depending on how his or her site uses the server. Some administrators will need to know very little of the information in this book, while others will need to know much more.

What you need to know

This guide assumes user-level experience with the A/UX (or UNIX®) operating system. Before you read this guide, you should know how to operate a Macintosh computer by using the Finder (as explained in the *Basic Skills* guide) and, you should be familiar with the information covered in *Setting Up and Managing Your Server*.

What's covered in this guide

This book covers the following topics:

- Chapter 1, “Using CommandShell,” introduces the CommandShell application for users new to A/UX.
 - Chapter 2, “An Introduction to Using UNIX Commands,” discusses the syntax of A/UX commands and gives examples of how to use them.
-

- Chapter 3, “System Startup,” discusses the A/UX Startup program, which allows you to configure certain startup parameters for A/UX 3.0.1. It also discusses different system states and tells you how to configure the kernel.
- Chapter 4, “Backup Media and Devices,” discusses backup strategies and media, and device files.
- Chapter 5, “Managing Disks,” covers procedures you can use to keep your disk free of unwanted administrative files and discusses using CD-ROM discs with the server.
- Chapter 6, “Creating and Managing a TCP/IP Network,” describes network-related issues that you need to resolve when you do not have an existing TCP/IP network to connect to, and want to create a network for the Apple Workgroup Server 95.
- Chapter 7, “Printing From CommandShell,” discusses the UNIX print daemon `lpr` and how to configure and administer A/UX 3.0.1 to use it. It also provides basic information on using the `lpr` command to print a file.
- Chapter 8, “User and Group Administration for A/UX Users,” discusses users and groups and their command shell environments. It also covers aspects of file and directory permissions.
- Chapter 9, “Troubleshooting,” covers various techniques for responding to hardware and A/UX 3.0.1 operating-system issues, error messages, and network failures.
- Chapter 10, “Checking Your Data Consistency,” describes in detail the `fsck` file-system checking program, including all the possible messages you may receive and how to respond to them.
- Appendix A, “CommandShell Menu Items,” provides a reference of all the capabilities of CommandShell.
- Appendix B, “Network Contacts,” describes how to contact the official network agencies if you wish to connect your system to the Internet.
- Appendix C, “Information on Files in A/UX 3.0.1,” lists the files that are part of A/UX but are not part of a standard UNIX operating-system implementation.

The guide also includes a glossary of terms and an index. For references to more information about A/UX 3.0.1 and the Apple Workgroup Server 95, see “For More Information” later in this preface.

Balloon Help

The Apple Workgroup Server 95 software comes with a feature called Balloon Help, which provides immediate on-screen help when you need it. If at any time you don’t understand the instructions in this guide—or if you see a window, button, menu, command, or icon that you want to know more about—use Balloon Help to get the help you need. For instructions on using Balloon Help, see the *Basic Skills* guide.

On-line documentation

The contents of this guide, and of most of the other documentation that comes with the Apple Workgroup Server 95, are provided in electronic form so that you can display them on your screen. If you prefer using on-line forms of documentation, you can use the on-line utility called Apple DocViewer, which comes with the Apple Workgroup Server 95. The user’s guide for Apple DocViewer is itself provided in on-line form, in the Documentation folder on your server. The *Apple DocViewer Quick Reference Card*, packaged with the Apple Workgroup Server 95, explains how to start Apple DocViewer and display the user’s guide.

Conventions used in this guide

The Courier font

Words that appear on the screen or that you must type exactly as shown are in the Courier font. For example, suppose you see this instruction:

Type `date` on the command line and press RETURN.

The word `date` is in the Courier font to indicate that you must type it. Suppose you then read this explanation:

After you press RETURN, information such as this appears on the screen:

Tues Oct 13 17:04:00 PDT 1992

In this case, Courier is used to represent the text that appears on the screen.

In cases where this book shows text that appears on the screen that you need to respond to, the output from the system is printed in the Courier font, and the text that you need to type is printed in Courier Bold.

The Courier font is used for the names of commands, domains, files, and directories. It is also used for the names of application programs that present an interface similar to that presented by the UNIX operating system (a character-based interface), such as the `login` program. (The names of application programs that present a Macintosh-style interface, such as the Finder, A/UX Startup, and Login programs, appear in the regular text font.)

All A/UX 3.0.1 manual page names are also shown in the Courier font. For example, the entry `ls(1)` indicates that `ls` is the name of a manual page in a reference manual. See “Manual Page Reference Notation,” later in this preface, for more information on the reference manuals.

Italics

When shown in text, commands often contain “placeholder” words or characters that appear in italics. These placeholders represent parts of a command for which you substitute different values when you actually enter the command, depending on what you want to do with the command. For example, in the sample command

```
cat file
```

the word *file* is a placeholder for the name of a file you want to display. If you wanted to display the contents of a file named `Elvis`, you would issue the `cat` command, typing the word `Elvis` in place of *file*. In other words, you would enter

```
cat Elvis
```

Sometimes italic placeholders are used in other contexts—for example, to represent text that appears on the screen or to represent the value of a field in a file. Here is a sample prompt that might appear on your screen:

Apple Computer, Inc. (*hostname*)
login:

In this prompt, the word *hostname* is a placeholder for the name of the host computer to which you can log in.

A/UX command notation

A/UX books use special notation to present A/UX commands. This notation is designed to reflect the syntax of the command—that is, to indicate how to enter the command so that its structure is legal and its components are interpreted properly.

Here is a brief description of each element of the A/UX command notation.

Element	Description
<i>command</i>	The command name. This element appears in the Courier font, like all other words or characters that you type, as explained earlier.
<i>option</i>	A character or group of characters that modifies the behavior of the command. Most options have the form <i>-option</i> , where <i>option</i> is a letter representing an option. Most commands have one or more options. Like the command name, option letters that you type appear in Courier.
<i>argument</i>	A value that modifies the behavior of a command, typically the name of a file or object that the command acts upon. When “placeholder” words or characters are used in text to represent arguments, they appear in italics.
[]	Brackets are used to enclose an optional item—that is, an item that is not essential for execution of the command. Brackets are not typed when using the command.
...	Ellipsis points are used to indicate that you can enter the argument preceding the ellipsis points more than once.

A typical A/UX command line comprises the command name, followed by options and arguments. For example, the `wc` command, which is used to count characters, lines, and words in a file, would look like this in an A/UX manual:

```
wc [-c] [-l] [-w] file...
```

In this example, `wc` is the command; `-c`, `-l` and `-w` are options, of which you can specify zero or more; *file* indicates that an argument consisting of a filename is required; and the ellipsis points (...) indicate that you can specify more than one filename argument.

To count the words in a file named `Priscilla`, you would use the `-w` option and replace the placeholder *file* with that filename. The command you enter would look like this:

```
wc -w Priscilla
```

Note that each command element is separated from the next element by a space; in A/UX command lines, elements must be separated by one or more spaces, tabs, or newline characters.

The elements of A/UX commands and the procedures for using them are discussed in *Basic Skills*.

Manual page reference notation

A/UX Command Reference, *A/UX Programmer's Reference*, *A/UX System Administrator's Reference*, *X11 Command Reference for A/UX*, and *X11 Programmer's Reference for A/UX* contain descriptions of commands, subroutines, and other related information. Such descriptions are known as *manual pages* (often shortened to *man pages*). Manual pages are organized within these references by section numbers. The standard A/UX cross-reference notation is

command (section)

where *command* is the name of the command, file, or other facility, and *section* is the number of the section in which the item resides.

- Items followed by section numbers (1M) and (8) are described in *A/UX System Administrator's Reference*.
- Items followed by section numbers (1) and (6) are described in *A/UX Command Reference*.
- Items followed by section numbers (2), (3), (4), (5), and (7) are described in *A/UX Programmer's Reference*.
- Items followed by section number (1X) are described in *X11 Command Reference for A/UX*.

- Items followed by section numbers (3X) and (3Xt) are described in *X11 Programmer's Reference for A/UX*.

For example, `cat(1)` refers to the command `cat`, which is described in Section 1 of *A/UX Command Reference*.

You can display manual pages on the screen by using the `man` command. For example, you could enter the command

```
man cat
```

to display the manual page for the `cat` command, including its description, syntax, options, and other pertinent information. To exit a manual page, press the SPACE bar until you see a command prompt, or type `q` at any time to return immediately to your command prompt.

On-line help and documentation

You can display information about server administration with A/UX 3.0.1 on your computer screen, and get on-screen help information about using A/UX 3.0.1, in several ways.

For more information

For more information on AppleShare, refer to the *AppleShare Pro Administrator's Guide*. If you are using A/UX 3.0.1 to perform AppleShare administration tasks, you will need to refer to this document only to learn more about A/UX or to troubleshoot problems.

For an overview of the concepts of server administration, refer to *Setting Up and Managing Your Server*.

Information about the optional (custom-installation) packages on the Apple Workgroup Server 95 system software installer CD-ROM discs is contained in other documents that must be purchased from Apple separately. These documents include

- *A/UX Essentials*
- *A/UX Network System Administration*
- *A/UX Command Reference*

1

Using CommandShell

This chapter describes how to use CommandShell, an application that provides access to the traditional UNIX[®] command-line interface. This chapter covers the following topics:

- making CommandShell active
- using CommandShell windows
- the A/UX System Console windows
- customizing CommandShell windows
- editing in CommandShell windows
- leaving CommandShell

Start by reading the first two sections of this chapter, “Introduction to CommandShell” and “Making CommandShell Active.” You can then consult any of the remaining sections that describe the tasks you want to perform.

Introduction to CommandShell

CommandShell allows you to display one or more windows in which you can communicate with A/UX by entering commands on a command line, as is traditionally done on most UNIX systems. When you do this, you can communicate with A/UX through any one of three standard shells:

- the C shell
- the Bourne shell
- the Korn shell

Like the Finder, a shell is a utility that establishes an interface between you and the operating system. It accepts the commands that you enter and passes them on to the operating system, which executes them. It also displays the operating system's responses. There are several standard shells, each of which interprets your commands and the operating system's responses somewhat differently than the others. Since the C shell is one of the most commonly used UNIX shells, the `adduser` command, which adds users to the system, uses it as the default shell when configuring user accounts. Information on changing shells is given later in this chapter.

Making CommandShell active

Before you read this section, you should start A/UX and, if prompted with a Login dialog box, log in. When the Finder is active, its menu bar appears at the top of the screen.

To make CommandShell active:

- **Choose CommandShell from the Application menu, which is at the right side of the menu bar.**



A window labeled “CommandShell 1” opens and becomes the active window.



When a CommandShell window becomes active, the CommandShell menu bar appears at the top of the screen. It contains commands that allow you to manipulate the CommandShell windows and their contents.

Switching between CommandShell and the Finder

You can return to the Finder by clicking anywhere in a Finder window, by clicking anywhere on the desktop, or by choosing Finder from the Application menu. You can then reactivate CommandShell by clicking anywhere in a CommandShell window or by choosing CommandShell from the Application menu.

IMPORTANT Each CommandShell window separately keeps track of the directory it is in and the user defined in it. Additionally, the folder that you are using in the Finder does not determine the current working directory of a CommandShell window. If you switch to another login name or another directory in a CommandShell window, and then leave that window and return to it later, it will still have the same login name in the same directory as when it was last used, regardless of what you have done in other CommandShell windows.

Using CommandShell windows

The following sections explain how to work with CommandShell using the menu commands. You can read through these sections sequentially, or you can find the task you want to complete and go directly to the applicable section.

Displaying CommandShell windows

When you first start CommandShell, a CommandShell window opens, unless you have previously closed all CommandShell windows.

To display new or additional CommandShell windows:

- **Choose New from the File menu, or press COMMAND-N.**

Either of these actions makes a new window open in front of the existing window or windows. The CommandShell windows are numbered in the order in which they are created (CommandShell 1, CommandShell 2, and so on).

Note: You can open up to 15 windows. It is best to use a large-screen monitor when you work with many windows at the same time.

Arranging CommandShell windows

You can arrange CommandShell windows on the screen in any of three ways:

- *Standard positions:* When you create a new window, it appears in front (and slightly below and to the right) of the previously created window, partially obscuring the windows behind it, as shown in Figure 1-1.
- *Tiled positions:* You can tile the windows to arrange them alongside each other with their edges touching, like tiles on a floor. Tiling windows reduces the size of the windows so that they fit on the screen.
- *Manually arranged positions (resizing and dragging):* You can arrange CommandShell windows on your desktop manually by using their size boxes to resize them and by dragging their title bars to move them to new locations on the desktop.

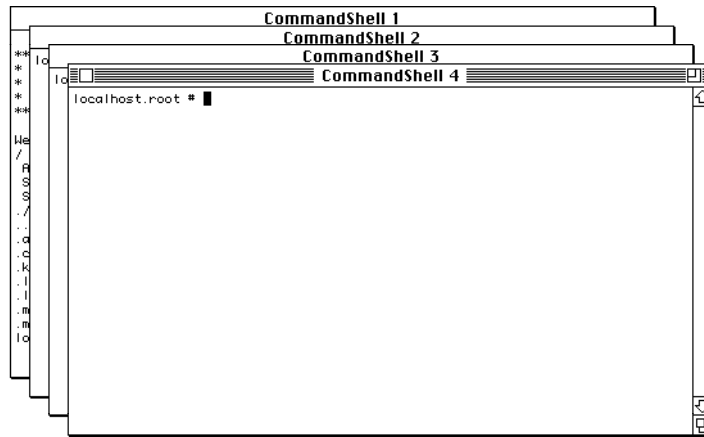


Figure 1-1 Windows in standard positions

WARNING Unlike standard Macintosh windows, CommandShell windows do not preserve text that is obscured when they are reduced in size (because of tiling or resizing). The obscured text is lost; you cannot retrieve the contents by enlarging the window.

For further information on tiling, sizing, moving, and hiding CommandShell windows, see Appendix A, “CommandShell Menu Items.”

Recording the contents of a window

As you type new lines in a CommandShell window, the top lines can scroll past the top edge of the window, as shown in Figure 1-2. CommandShell automatically records a preset number of these lines. You can review them by scrolling upward. By default, CommandShell records 1000 lines.

For information on how to change the 1000-line default, see “Customizing CommandShell Windows” later in this chapter.

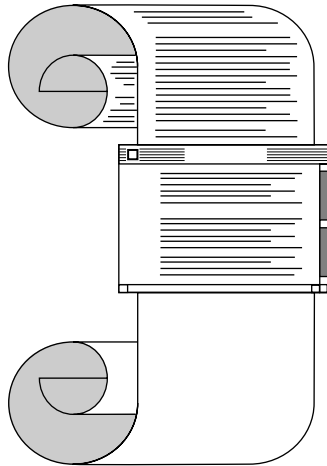


Figure 1-2 Recording window contents as they scroll off the top

Figure 1-3 shows how the window looks after you use the scroll bar to redisplay the recorded lines. The dotted line across the window marks the top of the most recent screen. It separates the active area (the material below the dotted line) from the recorded area (the material above the dotted line). Material in the active area may be overwritten if you start a full-screen editor or use the `clear` command, but material in the recorded area will remain available for viewing.

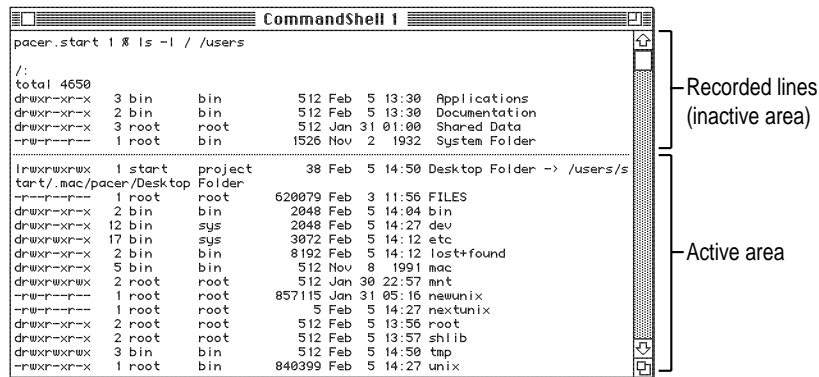


Figure 1-3 Viewing recorded window contents

Note: CommandShell does not save the recorded lines for use after you close the window. If you have accumulated data that you want to save, select it and copy it to a file in a text-editing or word-processing program. See “Saving a Selection in a New File” later in this chapter.

Canceling the recording of window contents

To stop CommandShell from recording the top lines of the active window:

- **Choose Don't Record Lines Off Top from the Commands menu.**

This command causes CommandShell to stop recording lines as they scroll past the top of the active window. Any lines already recorded are saved and are still available for review. The command name in the Commands menu changes to Record Lines Off Top. Choose this command to begin recording the lines again in the same window.

Erasing recorded window contents

To erase recorded lines:

- **Choose Clear Lines Off Top from the Commands menu.**

The recorded lines are erased and are no longer available for review. The scroll bar becomes dimmed. This command operates only on the active window.

Closing CommandShell windows

Before you can close a window, it must be the frontmost, or active, window. (The active window has horizontal stripes running across its title bar.) To make a window active, click anywhere in the window, or select the window from the Window menu.

Closing a window

IMPORTANT Unless you save the contents of a window to a file before you close the window, CommandShell discards the contents when you close the window. It is often helpful to save these detailed transcripts of commands you have entered and responses you have received. For information on saving the text of A/UX command lines to a file, see “Saving a Selection in a New File” later in this chapter.

To close an active window:

- **Choose Close from the File menu, press COMMAND-W, or click the close box of the window.**

In each case, the CommandShell window disappears from the screen.

If you close a window in which a process or program is running, the process or program halts. An alert box warns you of this situation and gives you the option of keeping the window open.



To keep the named window open and the program or process running, click Cancel. If you click OK, all running jobs are halted.

Closing all open windows

To close all the open windows at once:

- **Choose Close All Windows from the File menu.**

The CommandShell windows disappear from the screen.

If one or more of the windows contains a process or a program that is running, the process is halted. An alert box warns you of this situation. Respond as described in the preceding section.

Printing the contents of a CommandShell window

If you have a printer connected to your computer either directly or through an AppleTalk network, you can print text from a CommandShell window.

To print text from a CommandShell window (for instance, a series of commands that you want to save in hard copy):

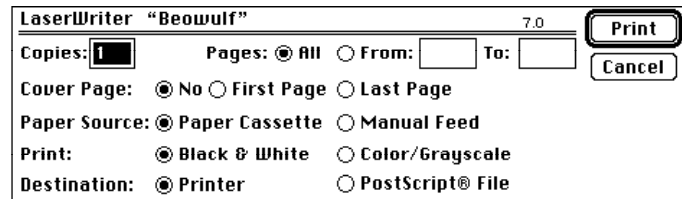
- 1 **Drag the I-beam pointer across text you want to print, and release the mouse button at the end of the text block.**

The selected text appears highlighted.

To set printing specifications for the selection, use the Page Setup command in the File menu before performing step 2.

- 2 **Choose Print Selection from the File menu.**

A print-file dialog box appears.



- 3 **Click Print to print, or click Cancel to cancel the printing command.**

Using the A/UX System Console

When you are working in A/UX, the system might generate messages regarding its status. In a traditional UNIX environment, these messages appear on a console terminal. Typically, the system administrator monitors these messages, which alert the administrator to take action to avert or remedy a problem. In A/UX, the system messages appear in the A/UX System Console window, a CommandShell window that you display by choosing its name from the Window menu.

Note: If you quit the CommandShell application, and therefore do not have a System Console window available, the system messages will appear directly on your screen, starting from the upper-left corner, writing over your other windows.

If you accidentally quit CommandShell, you can start it up again by choosing the CommandShell menu command from the Apple (🍏) menu.

Displaying the A/UX System Console window

To view the contents of the A/UX System Console window:

- 1 Choose CommandShell from the Application menu.**
This action displays an active CommandShell window and makes the CommandShell menu commands available.
- 2 Choose A/UX System Console from the Window menu, or press COMMAND-0 (that is, the number “zero”).**

This action displays the A/UX System Console window.

Clearing the A/UX System Console window

To clear the messages from the A/UX System Console window:

- **Choose Clear Lines Off Top from the Commands menu.**
This action clears all the information from the window. You should clear this window periodically so that you can tell when new messages appear.

Closing the A/UX System Console window

Close the A/UX System Console window just as you close any CommandShell window. (See “Closing CommandShell Windows” earlier in this chapter.) Note that unlike other CommandShell windows, the A/UX System Console window does not discard its contents when you close it. Closing the System Console window is equivalent to “hiding” it.

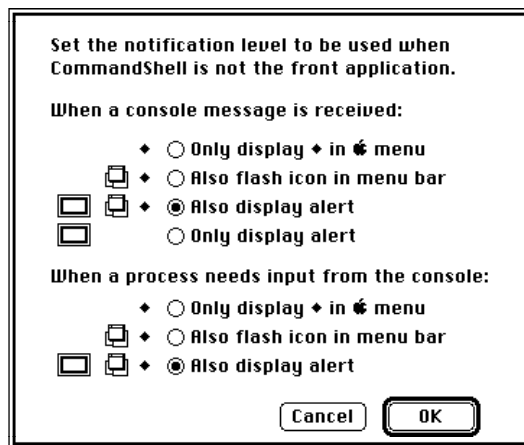
How A/UX alerts you to a message

If A/UX needs to bring a message to your attention, an icon flashes in the menu bar and an alert box appears. If CommandShell is not the active application, a diamond-shaped symbol also appears alongside CommandShell in the Application menu. Choose CommandShell, and then choose A/UX System Console from the Window menu to display the alert message.

To change the default settings to suppress the alert box, the flashing menu-bar icon, or both:

1 Choose Notification Levels in the Preferences menu.

The Notification Levels dialog box appears.



2 Select the desired combinations and click OK.

Customizing CommandShell windows

If you use CommandShell windows often, you might find it convenient to establish an ideal arrangement of windows, so that as soon as you log in to A/UX, the windows appear exactly the way you want them to.

Presetting window defaults

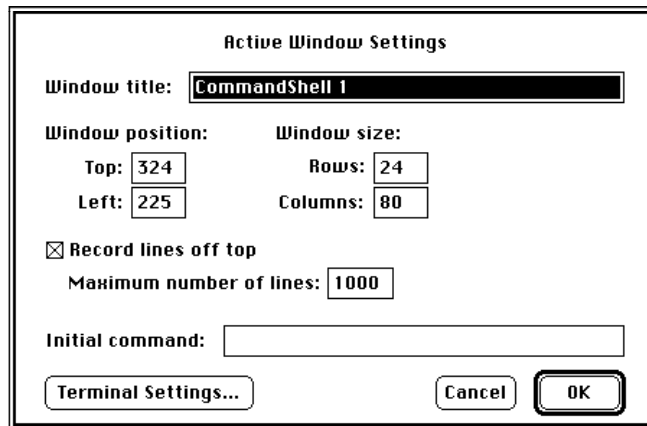
The Active Window Settings command in the Preferences menu gives you the following options:

- save an initial command to execute upon activating CommandShell
- change the title of the window
- change the number of lines to record

To preset window defaults:

- 1 **Choose Active Window Settings in the Preferences menu.**

The Active Window Settings dialog box appears.



The name of the active window is highlighted in the text box at the top of the dialog box.

- 2 **Change any or all of the settings in this dialog box.**

Notice that the checkbox “Record lines off top” is selected by default. If you click the checkbox, the X disappears and recording of lines is turned off. You

can also change the maximum number of lines recorded, but only if “Record lines off top” is selected.

In the “Initial command” text box, you can enter a command to run automatically as soon as you activate CommandShell.

3 Click OK.

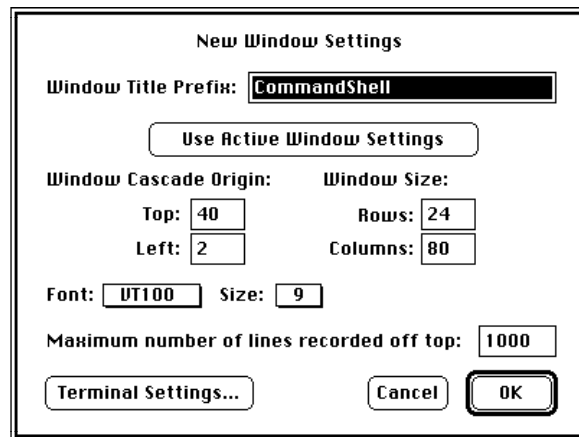
The new settings affect the active window. If you saved the new settings by choosing Save Preferences from the Preferences menu, the new settings become the new defaults for displayed windows. See “Saving the Layout of Your CommandShell Windows” later in this chapter.

Presetting the appearance of new windows

To change the default settings that control the appearance of new windows before you create them:

1 Choose New Window Settings from the Preferences menu.

The New Window Settings dialog box appears.



2 Change any or all of the settings, or use the active window settings.

You can extend the active window settings (described in the preceding section) to all new windows. To do so, click the Use Active Window Settings button.

If you want to label new windows with a prefix other than “CommandShell,” type the prefix in the Window Title Prefix text box.

You also can preset the window size, the screen font and size to be used in the window, and the maximum number of lines to be saved by typing your choices in the appropriate boxes. The boxes next to Font and Size are pop-up menus—press them to select a different font and size from their menus.

To preset a default position for the upper-left corner of the first window of a group of stacked windows, enter the coordinates (in pixels) in the Window Cascade Origin text box.

3 Click OK.

Your new settings take effect.

Saving the layout of your CommandShell windows

The Save Preferences command in the Preferences menu allows you to save the following specifications:

- notification-level preferences (described in “How A/UX Alerts You to a Message,” earlier in this chapter)
- the positions and sizes (in rows and columns) of all CommandShell windows
- the stacking order of the windows
- the titles of the windows
- the fonts and point sizes of text in the windows
- whether a given window is visible or hidden
- whether a given window is “zoomed” (that is, enlarged to full-screen size with the Zoom Window command in the CommandShell Window menu) and the sizes (in pixels) of its zoomed and “unzoomed” states
- whether lines are to be recorded off the top of the screen, and the number of lines to be recorded (discussed in “Recording the Contents of a Window” and “Presetting the Appearance of New Windows,” earlier in this chapter)
- the initial command to be run in the window (discussed in “Presetting Window Defaults,” earlier in this chapter)
- terminal emulation information

To save the settings you choose:

1 Arrange the windows just as you want them to be.

Be sure that you have chosen any special formatting features that you want (such as fonts).

2 Choose Save Preferences from the Preferences menu.

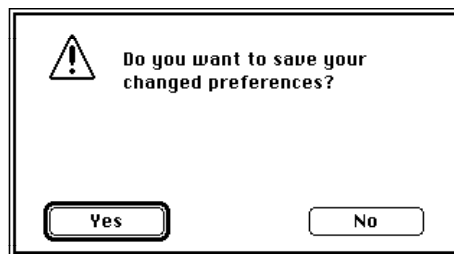
Restoring the window layout to the saved preferences

If you have saved your window preferences but have changed the window layout on your screen, you can instantly restore your preferred layout.

To restore the layout:

■ **Choose Restore From Preferences from the Preferences menu.**

If you have changed your window preferences with the Active Window Settings or the New Window Settings dialog box, but have not saved them by choosing Save Preferences in the Preferences menu, you are asked whether you want to save the changed preferences when you attempt to log out or shut down the system.



Click Yes or press RETURN to save the changed preferences, or click No to abandon them.

Changing shells

You can change the shell that you automatically get when you log in by using the `chsh` command. For further information on this command, see `chsh(1)` in the on-line manual pages.

To change shells temporarily, while you are logged in, use one of these methods:

- **Change to the Korn shell by displaying a CommandShell window and entering `ksh`.**
- **Change to the Bourne shell by displaying a CommandShell window and entering `sh`.**
- **Change to the C shell by displaying a CommandShell window and entering `csh`.**

Using Macintosh editing tools in CommandShell windows

You can use some of the standard Macintosh editing procedures while working in CommandShell windows. The editing procedures allow you to paste a series of commands or a long, complex command into another window without having to retype it.

The sections that follow describe how to copy or paste the contents of a window using the mouse and the menu commands. This material applies to command lines, system messages, and the output of commands. If you are using a text-editing application, use the editing tools provided by the application.

Selecting text

You must select text before you perform the editing procedures.

To select text:

- 1 Position the I-beam at the beginning of the text you want to select.**
- 2 Press and hold down the mouse button, and drag the I-beam across the text.**
- 3 When the block of text is completely selected, release the mouse button.**

Text is highlighted when it is selected.

Copying text

To copy text to the clipboard:

- 1 **Select the text you want to copy.**
- 2 **Choose Copy from the Edit menu, or press COMMAND-C.**

The highlighted text is copied to the Clipboard.

Pasting text

Once you have copied text to the Clipboard, you can paste the text into a CommandShell window.

To paste text into a window:

- 1 **Click at the point in the window where you want to paste the copied text.**
The insertion point should appear where you click.
- 2 **Choose Paste from the Edit menu, or press COMMAND-V.**

The text is pasted at the insertion point, as long as the insertion point is not in the middle of an active window (while a shell is active) or in the scrolled-off area.

You can paste the Clipboard contents as many times as you want, until you replace it by copying or cutting something else.

Changing the font or font size

You can change the appearance of all the text in a CommandShell window by changing the font or font size. The choices available in the Fonts menu are the fixed-width fonts in the System file of the active System Folder. At the bottom of the list of available font sizes in the Fonts menu, the last item, Other, allows you to select any font size that does not appear in the list.

To change the font or font size:

- **Choose the font or font size you want from the Fonts menu.**

All text in the window changes to the new font or font size.

- If you choose **Other**, a dialog box appears. Type the font size in points, and click **OK**.



The text in the window changes to the size you type.

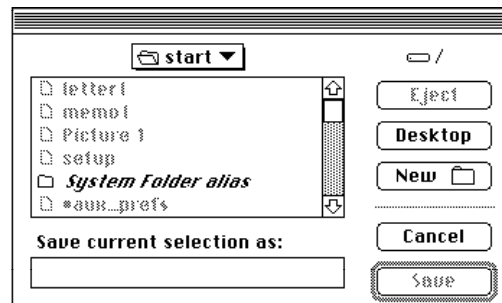
Saving a selection in a new file

You can save the contents of a CommandShell window in a text file.

To save the text displayed in a CommandShell window:

- 1 **Select the text you want to save.**
- 2 **Choose Save Selection from the File menu.**

A directory dialog box appears. Use the directory list and the pop-up menu above the list to change directories, if needed.



- 3 **Type a name for the new file.**
- 4 **Click Save.**

Leaving CommandShell

Make sure you save any work that you have completed in CommandShell windows before you return to the Finder in A/UX.

There are several ways to leave CommandShell:

- **Choose Hide CommandShell from the Application menu.**

This action hides all the windows and sets CommandShell aside until you reactivate it. This method reduces screen clutter while you work with other applications.

- **Choose another application from the Application menu.**

This action moves CommandShell windows to the background of the screen. The application you have chosen becomes active, and one of its open windows moves to the foreground.

- **Choose Quit from the File menu.**

Choosing this item turns off CommandShell. The only reason for turning off CommandShell is to load new fonts into the System file, which you cannot do unless all applications are turned off. If you do turn off CommandShell, be sure to restart it immediately after you finish loading the fonts.

Note: If CommandShell is turned off, and the monitor is set to black and white, the operating system displays error messages on the desktop, over existing windows and graphics. If the monitor is set to color or grayscale, the system displays garbled characters across the desktop, because there is no System Console window to put the messages in.

By default, there is a CommandShell alias in the Apple Menu Items folder (located in the System Folder) that allows you to turn CommandShell on again by choosing CommandShell from the Apple menu.

2

An Introduction to Using UNIX Commands

This chapter teaches you how to use UNIX commands to perform a variety of useful UNIX operations. It also shows you how to consult the on-line manual pages (often called *man pages*).

If you are unfamiliar with UNIX commands, read the opening sections of this chapter in sequence, through the section “Options and Arguments.”

UNIX pathnames

This section describes the way the UNIX operating system specifies the location of files and directories in the UNIX file hierarchy.

The folders of the Macintosh environment correspond to the directories of the UNIX world. Because this section describes the traditional UNIX way of using pathnames in an A/UX CommandShell window, the word *directory* is used here.

The UNIX file system is organized in a hierarchy that is traditionally represented as an inverted tree. Organized from top to bottom, the file system begins at the top with a directory labeled with a slash character (/), called the *root directory* because it is the “root” of the file system. Branching downward from the root directory are the rest of the directories and files in the system. The root directory contains a number of directories that store the operating-system files and programs. There is only one UNIX file hierarchy for a given system. All additional directories, file systems, and files fall within this hierarchy.

Because of its hierarchical organization, A/UX uses pathnames to refer to files and directories. A pathname is a name that describes where the file or directory is located in the hierarchy.

Figure 2-1 gives an example of this file-system hierarchy, showing the path that leads from the root directory to the file `letter1` and gives rise to the pathname `/users/start/letter1`.

Here, the file named `letter1` is located within the `start` directory, which is located within the `users` directory, which, in turn, is located in the root directory (indicated by the / character).

An absolute pathname of a file shows the complete path from the root to the file in question, by listing all directories that lead from the root directory and concluding with the filename itself. The first item in an absolute pathname is always the slash character, which indicates the root directory. Thereafter, the slash character is used to separate the names of the directories.

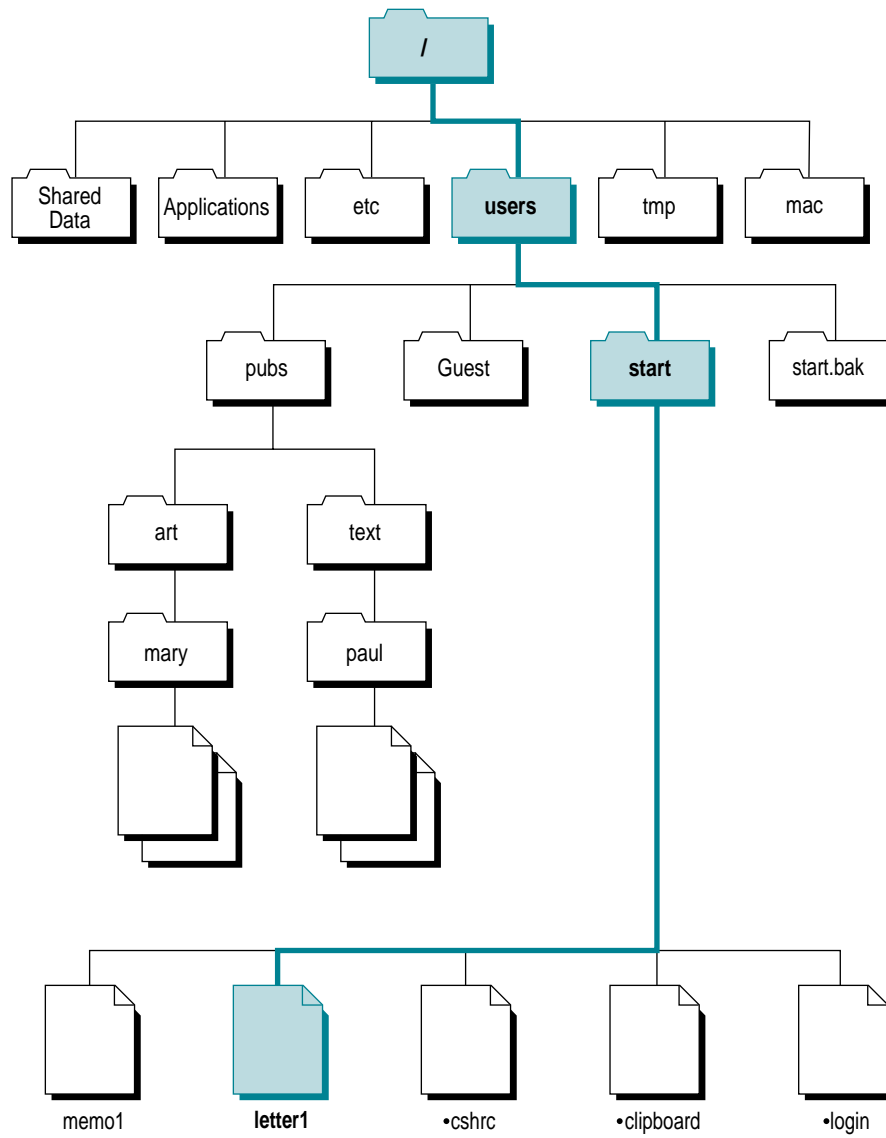


Figure 2-1 The pathname `/users/start/letter1`

You can tell, for example, that `/users/start/letter1` is an absolute pathname because it begins with the slash character, and is thus tracing the path from its origin at the root directory. The subsequent slash characters separate the names of the `users` and the `start` directories, and the names of the `start` directory and file `letter1`.

A relative pathname lists the directories leading to the file, beginning with the directory you are currently in, rather than always starting from the root directory. Figure 2-2 shows a relative pathname of `letter1`, when the user is already in the directory `/users`. Note that a relative pathname never begins with the slash character, because it doesn't start at the root directory. This convention can be compared with leaving out the area code in a telephone number if you are calling the number from within that same area code.

The directory in which you are currently working is called the *current directory*. When you change directories, the directory you move to becomes your new current directory. For example, if the current directory is `/users`, then `start/letter1` is a relative pathname of `/users/start/letter1`.

The name of the file or of the directory whose path is being described is the last component of a pathname.

If you are typing UNIX commands in the CommandShell command-line interface, and you want to specify a file that is in your current directory, just type the filename. You do not need to specify its directory, because you're already working in that directory. (For information on CommandShell, see Chapter 1, "Using CommandShell.")

Two dots (`..`) are an abbreviation for *the directory that contains the current directory*. For example, if you are in `/users/start`, you can specify the `/users` directory with two dots (`..`) as a shortcut.

Once you understand how UNIX pathnames work, you can easily learn to find your way around a UNIX hierarchy while working with the command-line interface.

Note: The folder that you have been using in the Finder does not determine the directory in which you are working when you are using a CommandShell window. Also, each Command Shell window has its own current directory.

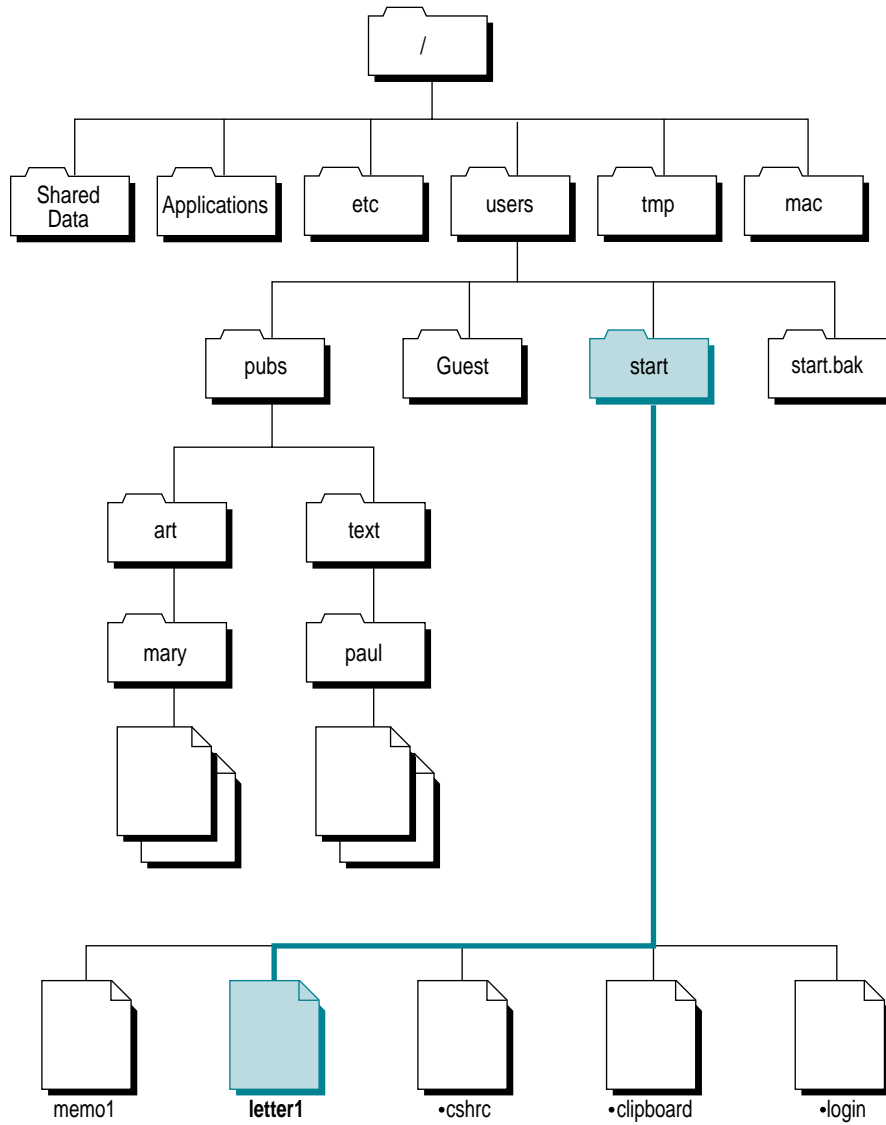


Figure 2-2 The relative pathname `start/letter1`

Introduction to UNIX commands

Chapter 1, “Using CommandShell,” tells you how to use CommandShell, the UNIX command-line interface. When you enter a command on the command line, you are instructing A/UX to run a program that performs a specific task or a specific set of tasks. This method is the traditional way of running commands in the UNIX operating system.

IMPORTANT As you go through the rest of this chapter, be sure that you enter UNIX commands exactly as specified. Don’t add or remove spaces, don’t change a character from uppercase to lowercase, or vice versa, and be sure to use minus signs where appropriate.

A command consists of a special kind of file called an *executable file* or *program*—that is, a file that contains a set of instructions that makes the computer perform a specific task. A/UX contains over 500 such commands.

For example, there is a command called `pwd`, which stands for *print working directory*. This command displays the pathname of the directory in which your CommandShell window is currently working (your current directory).

To perform the following exercises, be sure that you are logged in to the `start` account. (The password for the `start` account is `my.passwd`.) If you do not have a CommandShell window displayed, choose CommandShell from the Application menu. If a CommandShell window does not appear, choose New from the File menu, and a new CommandShell window appears.

Note: If you are working in the Finder and you open a new CommandShell window, the current directory of the CommandShell window is your home directory. If you bring a previously used CommandShell window to the foreground of the screen, that window is still in the directory it was in when last used. The folder that you have been using in the Finder does not determine the directory in which your CommandShell windows are working.

Using UNIX commands to navigate in the file system

You can use pathnames to determine where you are in the A/UX file system and to move from directory to directory in CommandShell. Log in to the `start` account to practice the following tasks on your system.

Finding your current directory

To determine what directory you're using:

- Enter the `pwd` command by typing `pwd` and pressing **RETURN**.

The pathname of your current directory (`/users/start`) appears in response to the `pwd` command.



```
localhost.start 1 $ pwd
/users/start
localhost.start 2 $ █
```

The `pwd` command caused the program (the executable file) called `pwd` to run—that is, to execute its instructions. The results of the command are commonly called the *output*. In the case of the `pwd` command (as is true of many commands), the output is displayed on the screen. In traditional UNIX parlance, one says that the output is directed to the screen. This is called the *standard output*.

Later in this chapter you will see that you can instruct a command to send its output somewhere other than to the screen. For example, the information can be sent to a file to be saved or to a printer to be printed. This rechanneling of data is called *redirection*.

Moving to a different directory

Suppose you use the `pwd` command and find that you are working in the root directory (indicated by a `/`) when you really want to be working in the `/users/start` directory. To change from one directory to another, use the `cd` (change directory) command.

To change to `/users/start`:

- **Enter this command:**

```
cd /users/start
```

IMPORTANT Be sure to type the space after `cd`. The space marks the end of the command and the start of the command argument (described in the next section).

If you enter `pwd` now, the response reads as follows:

```
/users/start
```

When you use the `cd` command, you can use the absolute pathname or relative pathname of the destination directory, whichever is easiest for you. If you aren't sure where you are relative to the destination directory, use an absolute pathname.

You can always return to your home directory by entering the `cd` command without specifying any directory.

Options and arguments

Options and arguments modify the way a command runs and specify the files (or other objects) upon which the command operates.

Options

Most UNIX commands can run in a variety of ways to produce a variety of results. These variations in the way a command works are called *options*.

Listing the contents of a directory: The `ls` command options

The `ls` command causes the computer to list all the files in your current directory. However, there are various ways in which you can list the files. If you use the `ls` command without any of its options, only the names of the files are listed, and they are listed in a single column.

Suppose you are working in CommandShell. To check that you are in the directory `/users/start`:

- 1 Enter the `pwd` command to see what directory you are in.

The command line should display the following response:

```
/users/start
```

- 2 If `/users/start` is not what appears, enter the following command:

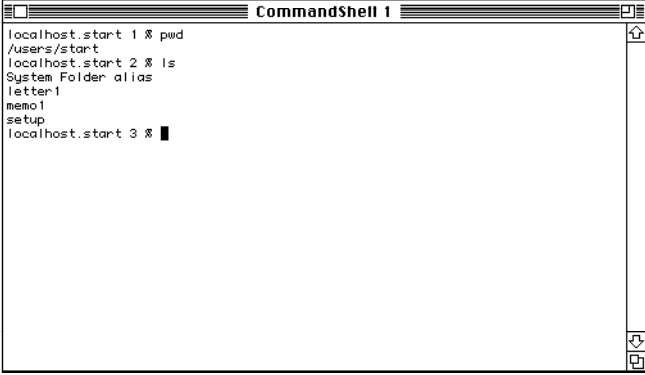
```
cd /users/start
```

- 3 Repeat step 1 to be sure that you are in the `/users/start` directory.

To list the files in the directory `/users/start`:

- Enter `ls` on the command line of a CommandShell window.

The resulting display looks something like the one in the following figure.



```
CommandShell 1
localhost.start 1 & pwd
/users/start
localhost.start 2 & ls
System Folder alias
letter1
memo1
setup
localhost.start 3 & █
```

But what if there are too many filenames to fit in one CommandShell window, so that the filenames at the top of the list scroll off the top of the window? You might want to avoid having to use the scroll bar to scroll back to the top of the list. Instead, you can break the list into columns so that more of the filenames can fit on one screen.

You can do this with the `-C` option (which stands for *columns*; note that this is an uppercase `C`). Whenever you type an option in the command line, it is preceded by a hyphen (sometimes referred to as *minus*); in conversation, the `-C` option is called *minus C*.

IMPORTANT Because UNIX is case sensitive, be sure that you do not use a lowercase `c` for the `-C` option. Some options call for lowercase characters and some call for uppercase characters.

The following procedures introduce you to two of the `ls` command options.

To list the contents of the `/` directory in multiple columns:

- 1 Enter `cd /` to move to the root directory.
- 2 Enter `ls -C` on the command line in the CommandShell window.

Be sure that you leave a space between the command and the option (that is, between the `ls` and the `-C`).

The screen displays output that looks something like this.

```
localhost.start 1 % ls -C /
Applications    etc             sa
Documentation   lib            shlib
Shared Data     lost+found     tmp
System Folder  alias         mac           unix
Desktop Folder mnt           users
FILES          newunix       usr
bin            nextunix
dev            root
```

If you want to know how large a file is, who owns it, or other technical information about the file, you can use the `ls` command with the `-l` option (note that this is a lowercase `ell`) for a long listing.

To get a long listing of the contents of the `/users/start` directory:

- Enter `ls -l` on the command line.

Be sure to type a space between the `ls` command and the `-l` (lowercase `ell`) option. The output looks something like the following:


```
localhost.start 2 % ls -l
total 12
-rw-r----- 1 start project 1486 Jan 16 20:49 System Folder alias
-rw-r--r-- 1 start project 1289 Jun 23 1992 letter1
-rw-r--r-- 1 start project 883 Jun 23 1992 memo1
-rwxr-xr-x 1 start project 94 Jun 23 1992 setup
```

The first four columns give permissions and ownership information. The next three columns tell the size (in bytes), the date on which the file was last changed, and the filename.

Arguments

Technically speaking, an argument is any item on the command line that modifies the way the command works. This guide, however, uses the word *argument* as it is commonly used, to mean the file or other object upon which the command acts. Some commands must have an argument, whereas others (such as `ls`) might or might not require an argument, and others (such as `pwd`) rarely or never take an argument.

Using the `ls` command with an argument

If you are working in the `start` directory and you want to display a list of the contents of the root directory, you use the `ls` command with the name of the root directory (the `/` character) as the argument.

To display a list of the contents of the root directory from the `/users/start` directory:

- **Enter `ls /` on the command line.**

Note the space between the command and its argument.

A listing of the contents of the root directory appears on your screen.

To list a single file by using its name as the argument:

- **Enter `ls -l letter1` on the command line.**

A single line appears that gives you the long listing for the file `letter1`, but for no other files in the directory.

Further useful UNIX operations

The following pages show you how to use several useful features of the UNIX operating system.

Searching for a string of characters and redirecting output

One useful UNIX command is `grep`; it is used for finding a string of characters in any file you specify. For example, suppose you need to glance at all references to the word *sales* in the files that are in the `start` directory.

Instead of looking at the results of the search on the screen, you can save the information in a separate file so that you have it available for future reference. This requires redirecting the output of the search to a file rather than displaying the results on a screen.

To redirect the output:

1 In a `CommandShell` window, type `grep`. Don't press `RETURN` yet.

2 Type a space, then type `-i` as an option.

A complete listing of the options can be found in the on-line manual page. Useful options include `-i` (ignore uppercase and lowercase) and `-n` (include the relative line number of the occurrence). For this example, use the `-i` option.

3 The `grep` command needs to be followed by a search pattern. Type the characters you are looking for, in this case, `sales`.

4 Decide which files you want to search, and add them to the command line. For this example, you should search `letter1` and `memo1` in the `start` directory.

```
/users/start/letter1 /users/start/memo1
```

Don't forget to type spaces between the pathnames.

5 To redirect the output to a new file, add the redirection symbol (`>`) and the name of the file:

```
>/users/start/salessearch
```

Your command line should look like this (although it will most likely be all on one line):

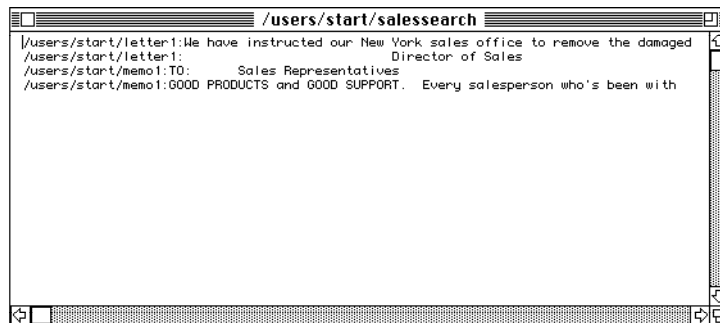
```
grep -i sales /users/start/letter1 /users/start/memo1  
>/users/start/salessearch
```

6 Press RETURN to run the command.

The following table explains the meaning of each item on the command line.

Command-line item	Meaning
<code>grep</code>	Search for a pattern of characters.
<code>-i</code>	Ignore case.
<code>sales</code>	Search for this pattern of characters.
<code>/users/start/letter1</code>	Search through these files.
<code>/users/start/memo1</code>	
<code>></code>	Redirect output to the following file.
<code>/users/start/salessearch</code>	Create this file and send the results of the search to it.

To check the results, return to the desktop by clicking the `start` folder or by choosing Finder from the Application menu. Note that the `start` folder now contains a file named `salessearch`. If the `start` folder is not open, double-click its icon to open it. Double-click the `salessearch` icon to open the text file. A TextEditor window opens to display the text file.



The listing shows you the full pathname of the file, followed by a colon, and the line that the characters are on.

Looking up UNIX commands on line

You can display on your screen reference information about any command. The reference information is organized into entries called *manual pages* (also known as *man pages*).

The term *manual page* might lead you to assume that each entry is only one page long, but that is not the case. Some of these entries are dozens of pages long. The manual pages are available in printed form in the A/UX 3.0 reference manuals.

To display a manual page:

- **Enter this command:**

```
man commandname
```

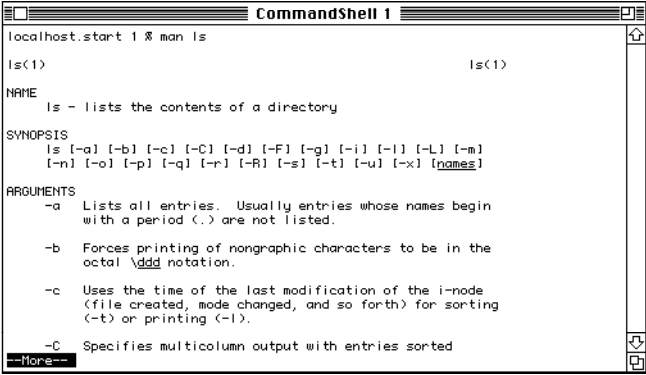
Type the name of a command (for example, `ls`) in place of *commandname*.

The text that appears on screen has somewhat simpler formatting than the text that you read in the *A/UX Command Reference*. For example, terms that are in italic type in the printed book are underlined on screen.

To examine the manual page for the `ls` command:

- **Enter `man ls` on the command line.**

The manual page text appears, as shown here.



```
localhost.start 1 % man ls
ls(1)                                     ls(1)
NAME
  ls - lists the contents of a directory
SYNOPSIS
  ls [-a] [-b] [-c] [-C] [-d] [-F] [-g] [-i] [-l] [-L] [-m]
    [-n] [-o] [-p] [-q] [-r] [-R] [-s] [-t] [-u] [-x] [names]
ARGUMENTS
  -a Lists all entries. Usually entries whose names begin
    with a period (.) are not listed.
  -b Forces printing of nongraphic characters to be in the
    octal \ddd notation.
  -c Uses the time of the last modification of the i-node
    (file created, mode changed, and so forth) for sorting
    (-t) or printing (-l).
  -C Specifies multicolumn output with entries sorted
--More--
```

Moving down through the manual page display

To show the next screen of text or advance line by line:

- Press the **SPACE** bar to display the next screen of text.
- Press **RETURN** to advance the text one line at a time.

To review material that has been scrolled off the top of the screen, use the scroll bar.

The “Name” section

Each manual page begins with the “Name” section, which gives a brief summary of what the command does. Here is an example:

```
ls - lists contents of a directory
```

The “Synopsis” section

The “Synopsis” section follows the “Name” section. It shows you the syntax of the command—that is, how the elements of the command line can be put together. It looks rather mysterious at first view, but it becomes clearer once you know what to look for. The `ls` synopsis reads as follows:

```
SYNOPSIS
ls [-a] [-b] [-c] [-C] [-d] [-F] [-g] [-i] [-l] [-L] [-m]
[-n] [-o] [-p] [-q] [-r] [-R] [-s] [-t] [-u] [-x] [name]
...
```

Don’t be discouraged by the long list of bracketed items. The synopsis summarizes the way the `ls` command is used on the command line. It will look clearer if you consider its separate components one at a time.

<code>ls</code>	You begin by typing the <code>ls</code> command itself.
<code>[-a] ... [-x]</code>	These are the options. You continue creating the command line by typing as many of the bracketed options as are appropriate.

Note that you do not type the brackets on the command line. The brackets are a conventional way of indicating that the bracketed item is optional.

As explained earlier in this chapter (in the section “Options and Arguments”), you can use the `ls` command without any options.

[*name*] Next, you type the argument, if necessary. The fact that it is bracketed indicates that it, too, is optional. In this case, *name* refers to name of a directory or a file.

... Ellipsis points indicate that more than one *name* argument can be specified.

The “Arguments” section

This section is essentially a list, with explanations, of all the options and arguments that can be used with the command. Notice that the options and arguments are listed together, in one section, and are arranged alphabetically. Some of the items in the list of `ls` command options are familiar to you. Here are two examples:

```
-C    Specifies multicolumn output with entries sorted
      vertically.
-p    Puts a slash (/) after each filename if that file
      is a directory.
```

The “Description” section

This section offers a full description of the operation of the command. Its length and complexity vary from command to command, depending on the complexity of the command’s use.

The “Examples” section

Not every manual page has an “Examples” section. The manual page for `ls` presents `ls -l` as an example, which is discussed earlier in this chapter. (See “Options and Arguments.”) When examples are provided, they are useful for seeing how the command is used.

The “Files” and “See Also” sections

There are several sections that appear in only a few manual pages (“Status Messages and Values,” “Warnings,” and “Limitations”). The “Files” section and the “See Also” sections, however, appear in most manual pages.

If you are not involved in the technicalities of UNIX, the “Files” section is unlikely to be of use to you. It describes the files that the command uses.

The “See Also” section can be useful to you. It cross-references other commands that are associated in one way or another with the command described by the current manual page, or whose uses parallel or supplement the command.

Quitting a manual page

To redisplay the command prompt without paging through the entire manual page, you can quit the manual page.

To quit the manual page:

- Press **CONTROL-C**.

Linking commands with a pipe

A/UX allows you to take the output from running one command and to channel it (or *pipe* it) into another command. Using this pipe feature, you can string commands together in order to run a given set of data through several processes, just as oil in a refinery moves from process to process through pipes. Figure 2-3 is a conceptual picture of how this process works.



Figure 2-3 Using a pipe to channel the output of Command 1 as the input to Command 2

The pipe symbol is a vertical bar (|), which is often found on the backslash (\) key on the keyboard, although the position of this key varies on different keyboards.

Earlier in this chapter, in “Options and Arguments,” you learned that the command `ls -l` produces a long listing of files and subdirectories in a directory, giving you several items of information in addition to the names of the files and directories. One of those items of information is the date on which the item was last changed.

The next exercise shows you how to set up a pipe to use the output of one command as the input to another. The task is to find all items in the `start` directory that were created or last changed in the year 1992. To do this, you first use the `ls -l` command to create a long listing of the contents of the `start` directory, because a long listing contains the dates on which the items in the list were created or changed; then you use a pipe to channel the output of the `ls` command to the `grep` command; then you use the `grep` command to search the output of the `ls -l` command (the list) for the characters 1992.

To find all items in the `start` directory that were created or last changed in the year 1992:

- 1 On the command line in a CommandShell window, type the following command, without pressing RETURN:**

```
ls -l /users/start
```

This is the list command with the option that creates a long listing, and the argument that specifies which directory's contents are to be listed. For the purposes of this exercise, use the absolute pathname of the `start` directory (`/users/start`).

- 2 Press the SPACE bar, type a pipe character (|), and then press the SPACE bar again.**

The command line now looks like this:

```
ls -l /users/start |
```

- 3 Type `grep`. Press the SPACE bar, and then type the search pattern 1992.**

In this exercise you will not use any of the options, so you need only type the character string to be searched for.

The text that is being searched is the output of the `ls -l` command, which you are channeling into the `grep` command with the pipe.

The command line should be

```
ls -l /users/start | grep 1992
```


4 Press RETURN.

In the default `start` directory, the window should display the following response:

```
-rw-r--r-- 1 start project 1289 Jun 23 1992 letter1
-rw-r--r-- 1 start project 883 Jun 23 1992 memo1
-rwxr-xr-x 1 start project 94 Jun 23 1992 setup
```

The pipe feature is useful for performing complex tasks that require processing information through several commands and that would take too much time and work to perform one command at a time.

3

System Startup

This chapter describes the application A/UX Startup. A/UX Startup provides a shell from which to use a set of utilities to troubleshoot and manipulate A/UX while the computer is still governed by the Macintosh Operating System (OS).

This chapter also discusses

- the system-startup procedures and dialog boxes
 - the different levels of operation of A/UX
 - how to build a kernel and change kernel parameters
 - shutting down the system
-

About A/UX Startup

The Apple Workgroup Server 95 is shipped with A/UX Startup as a startup application, so it is automatically started when the system is turned on. A/UX Startup is the Macintosh program that checks your hard disk for irregularities and then starts A/UX. It is usually used for troubleshooting, because it allows you to change files in the A/UX operating system before A/UX is actually running. It also contains some configuration menus. A/UX Startup is located on the disk partition called *MacPartition*.

By default, and as long as there is nothing wrong with the system, A/UX Startup completes the entire process of starting up A/UX with no intervention on your part. But you can interrupt A/UX Startup before it starts A/UX and enter the A/UX Startup shell to use the A/UX Startup utilities.

Many of the same commands and usage conventions available in A/UX are also available in A/UX Startup: shell variables, comments, and input and output redirection. For more information on A/UX Startup's capabilities, see `StartupShell(8)`.

IMPORTANT The new SCSI device naming scheme in A/UX 3.0.1 does not apply to commands in A/UX Startup. This includes accessing all devices, as well as commands such as `fsck`. This means that you cannot access hidden devices when in A/UX Startup. For a discussion of the SCSI naming scheme and how devices may become hidden, see *Setting Up and Managing Your Server*.

The command-line interface of A/UX Startup provides a user with all the privileges of the `root` account. Therefore, it is a good idea to protect A/UX Startup by selecting the “Password checking” checkbox in the General dialog box so that only users with the password to the `root` account can use that account. You don't need to enter this password to start up A/UX; password protection is enforced only for entering the A/UX Startup shell. (For more information, see “Protecting A/UX Startup With a Password” later in this chapter.)

Entering A/UX Startup

To enter A/UX Startup, watch for the status box (shown in Figure 3-1) that appears when you turn on your system. Click the Cancel button in this status box to cancel the A/UX startup process and display the A/UX Startup shell window. (Another way to cancel the startup sequence is to press COMMAND-PERIOD [.] while the Cancel button is visible.)



Figure 3-1 The startup screen from which to enter A/UX Startup

As startup progresses, the Cancel button changes to a Messages button. When the Messages button is active, you can click it to view the A/UX System Console window, which shows information on what is happening during the startup process, as shown in Figure 3-2.

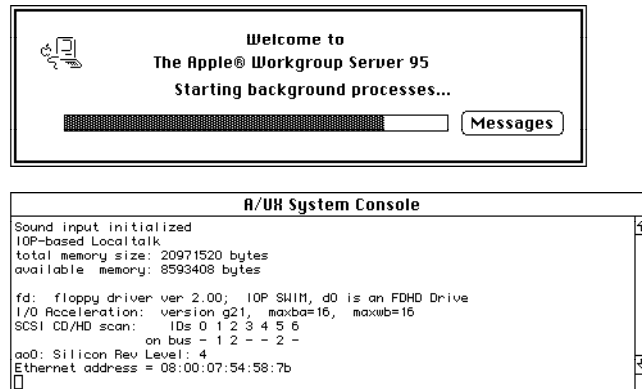


Figure 3-2 A/UX System Console window with startup information

Once the Cancel button has changed to the Messages button, you can no longer enter A/UX Startup. If you still wish to enter A/UX Startup, but you missed the Cancel button, wait until A/UX has started up, and then choose Restart from the Special menu.

WARNING Do not press the programmer’s switch, the restart switch, or the power switch while the system is starting up. This action could adversely affect the integrity of your data.

A/UX Startup shell window

This section describes the shell window, menus, options, and A/UX-like commands available in A/UX Startup.

Once the A/UX Startup shell window is displayed, a command-line prompt appears. The default prompt is

```
startup#
```

You can change the prompt to another string by redefining the shell variable `PS1`. Other shell variables that you can customize are `PATH`, `TZ`, `HOME`, and `ROOT`. See the `sh(1)` on-line manual page for more information on customizing shell variables.

IMPORTANT When you cancel the startup process for A/UX, the A/UX Startup shell window opens. If the top lines of the window denote a “dirty file system” (a file system that was not shut down correctly), as shown in Figure 3-3, you must run the `fsck` command on that file system before changing any files. Failure to run `fsck` may result in corruption of the file system. See Chapter 10, “Checking Your Data Consistency,” for more information.

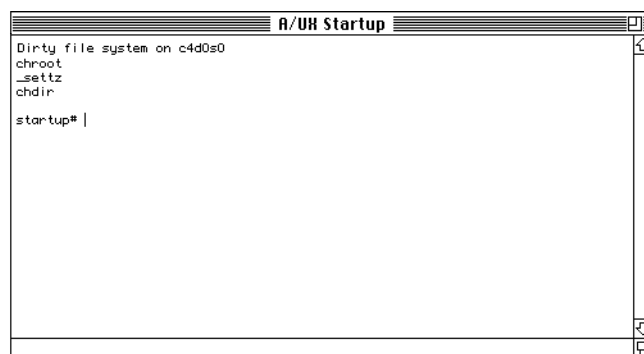


Figure 3-3 A/UX Startup window with notification of file-system problems

Commands that run in A/UX Startup

Two sets of commands are available to you in A/UX Startup. Both sets are similar to equivalent commands available in A/UX. One set is available from MacPartition.

cat	ed	newfs	tunefs
chgrp	fsck	od	ufs:fsck
chmod	kconfig	pname	ufs:fsdb
chown	launch	rm	ufs:mkfs
cp	ln	stty	
cpio	ls	svfs:fsck	
date	mkdir	svfs:fsdb	
dd	mknod	svfs:mkfs	
dp	mv	tar	

The second set is built into A/UX Startup.

auto	chroot	help	shutdown
autolaunch	default	powerdown	unauto
autorecovery	echo	pwd	unexport
boot	eject	readonly	unset
cd	exit	restart	
chdir	export	set	

Because these commands allow you to work on A/UX files from outside A/UX, they are especially useful for troubleshooting. For example, you might want to edit `/etc/inittab` without first starting up A/UX if you suspect that the contents of that file are causing problems when you start the system. For a description of this procedure, see “Accessing the A/UX File System From A/UX Startup” in Chapter 9, “Troubleshooting.”

A/UX Startup also provides a concise help facility, which is available if you type
help

Help for an individual command is available if you type
help *name*

where *name* is the name of one of the commands just listed.

You can also refer to the individual command descriptions in the manual pages, using the man command when you are logged in to A/UX.

A/UX Startup menus

The A/UX Startup menu bar contains the usual Apple, File, and Edit menus, as well as the Execute and Preferences menus, which allow you to configure A/UX. A brief description of each available configuration follows. For additional information, see StartupShell(8) in the manual pages.

Entering A/UX Startup on every startup

Although A/UX Startup is run every time you start up the system, the system does not stop at the A/UX Startup shell window by default. If you want to change this default setting, choose Booting from the Preferences menu of A/UX Startup. Click the “Automatically Boot at startup” checkbox in the Booting dialog box to remove the checkmark. (See Figure 3-4.) If this checkbox is not selected, the server will stop at the A/UX Startup shell window when you start up the system.

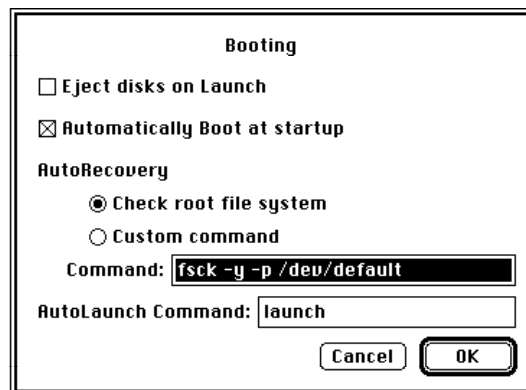


Figure 3-4 Booting dialog box

Starting A/UX from A/UX Startup

Once you have completed your work in A/UX Startup, you can start A/UX without having to restart the computer. To do this, either press `COMMAND-B` or choose `Boot` from the `Execute` menu.

The `Boot` command causes the `AutoRecovery` command to be executed, and then causes the `AutoLaunch` command to be executed if `AutoRecovery` is successful. (See Figure 3-4, the `Booting` dialog box, for the settings for these commands.) Choosing this menu item has the same affect as entering the `boot` command at the A/UX Startup command line.

Note: To have the system eject all floppy disks when A/UX starts running, select the “Eject disks on Launch” checkbox in the `Booting` dialog box.

To interrupt a program running in A/UX Startup

To interrupt the currently running program in A/UX Startup, press `COMMAND-PERIOD`, or `COMMAND-K`, or choose `Kill` from the `Execute` menu.

The autorecovery feature in A/UX Startup

The `AutoRecovery` command in A/UX Startup is run as one of the first steps in the startup process. By default, the “Check root file system” button is selected in the `Booting` dialog box, and the command shown in the `Command` text box performs a file-system check (by executing `fsck`) on the root file system. You can also select the `Custom Command` button and type the command of your preference in the `Command` text box. (Figure 3-4 shows the `Booting` dialog box.)

You would run this command from A/UX Startup if the system displayed error messages about the consistency of the root file system.

Choosing `AutoRecovery` from the `Execute` menu also runs the `AutoRecovery` command.

The autolaunch feature in A/UX Startup

The `AutoLaunch` command value set in A/UX Startup determines which version of `launch` is used to start A/UX from A/UX Startup. The `AutoLaunch` `Command` text box in the `Booting` dialog box displays the current value. (See Figure 3-4.) To change the value, select this box and edit the text. For example, to display console messages during the startup sequence, enter `launch -v` in

this field. These messages can be very useful when you are troubleshooting the system. See `launch(8)` in the manual pages. Choosing AutoLaunch from the Execute menu also runs this command. By default, the command is `launch`.

Starting up A/UX from a nondefault hard disk

You can specify a hard disk, other than the default, from which to start A/UX. To do this, choose General from the Preferences menu of A/UX Startup. In the General dialog box (shown in Figure 3-5), the Root Directory text box displays the value of the built-in `ROOT` variable, which defines which disk starts up A/UX 3.0.1.

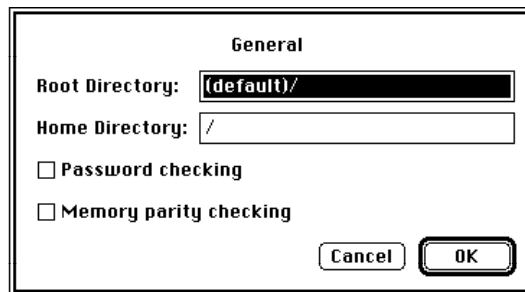


Figure 3-5 General dialog box

To change the value of the variable, select this box and edit the text. To start A/UX from the same hard disk as A/UX Startup, use

```
(default)/
```

To start A/UX from a different hard disk, use

```
(ID, 0, 0)/
```

where *ID* is the SCSI ID of the hard disk that contains A/UX. When you change the value in this window, the system makes the corresponding change in the AutoLaunch Command text box in the Booting dialog box.

Note: Because the format for the Root Directory text box does not allow you to designate a SCSI bus number, you must choose a SCSI ID that is not hidden. For a discussion of hidden SCSI devices, see *Setting Up and Managing Your Server*.

Changing the initial directory

The Home Directory text box in the General dialog box displays the value of the built-in HOME variable. To change the value of the variable, select this box and edit the text. This variable determines the initial directory (on the disk designated in the Root Directory text box) upon entering A/UX Startup.

Checking your parity memory

The “Memory parity checking” checkbox in the General dialog box should be selected if you want to use parity memory. Select this option only if you have upgraded the PDS card in your Apple Workgroup Server 95 exclusively with 9-bit parity SIMMs.

WARNING If you select the “Memory parity checking” checkbox in the General dialog box, and your system has 8-bit SIMMs or a mixture of 8-bit and 9-bit SIMMs installed, your system will crash. See *Setting Up and Managing Your Server* for more information on parity memory.

Protecting A/UX Startup with a password

The “Password checking” box in the General dialog box regulates whether a password is required for a user to enter A/UX Startup. When this option is selected, the dialog box shown in Figure 3-6 appears, requesting the root account name and password. This feature prevents unauthorized users from entering A/UX Startup, where they would have root access permissions.

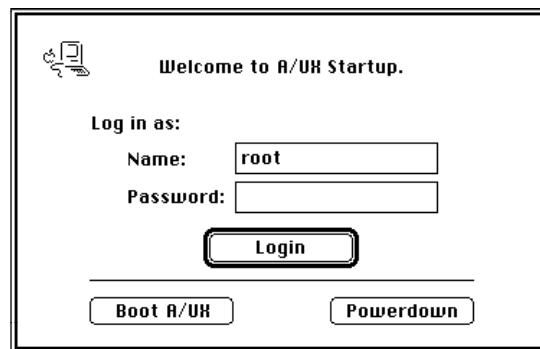


Figure 3-6 A/UX Startup password dialog box

The startup sequence

During the startup process, several startup screens are displayed. When you turn the power on, the computer automatically starts up in multi-user mode unless you designate otherwise, as described in “Single-User and Multi-User Modes” later in this chapter.

The first two startup screens that appear contain a Cancel button, as shown in Figure 3-7. When this Cancel button is active, you can click it to cancel the startup process and display the A/UX Startup shell window. The remaining startup screens contain a Messages button in place of the Cancel button. When the Messages button is active, you can click it to view the A/UX System Console window. This window shows information about what is happening during the startup process.



Figure 3-7 The Checking startup screen

At this point, the computer is still being run by the Macintosh OS. By default, A/UX Startup’s AutoRecovery command runs `fsck` for the root file system when the root file system “dirty” flag is set. (This flag indicates possible file-system damage.) If `fsck` fails, A/UX Startup has found a problem with one or more of the files in the file system. For additional information, see `fsck(1M)` and Chapter 10, “Checking Your Data Consistency.”

During the next phase, A/UX Startup loads the A/UX kernel into memory, preparing to hand off the system operation to A/UX. You can still cancel the startup process, return to the Macintosh OS, and operate the A/UX Startup utilities. (See Figure 3-8.)

Note: The Cancel button operates during the checking and loading phases.



Figure 3-8 The Loading startup screen

During the next phase, A/UX Startup passes control of the system to A/UX and the kernel is initialized. When launching starts, the Cancel button is disabled and is then replaced by the Messages button, as shown in Figure 3-9. The A/UX startup procedure can no longer be stopped. The progress bar does not change during this phase. As part of normal operation, the screen blinks momentarily.



Figure 3-9 The Launching startup screen

After A/UX has been launched, the root file system is checked again. (See Figure 3-10.) The `/etc/sysinitrc` script verifies that the root file system is not damaged; it then mounts the root file system and executes `autoconfig` and any device driver startup scripts. If `autoconfig` detects that the system hardware configuration has changed since the last startup (for example, if you have moved or removed a video card), `autoconfig` creates a new kernel. You are then prompted to restart the system. When you restart A/UX, the new kernel is used.



Figure 3-10 The “Checking root file system” startup screen

Note: The `autoconfig` command will not detect the addition of a new card.

Device drivers installed in the kernel are initialized after the file system is checked. (See Figure 3-11.) If `/etc/HOSTNAME` exists, the host name and domain name of the system are set. The default name is `localhost` and the default domain is `localdomain`. If you have configured the system to start up in single-user mode (instead of multi-user mode), you will enter it after this step.

See Chapter 7, “Printing From CommandShell,” for a discussion of device drivers.



Figure 3-11 The “Initializing device drivers” startup screen

In the final startup phase, general system housekeeping commands contained in the `/etc/brc` and `/etc/rc` scripts are executed, and file systems other than the root file system that cleanly passed `fsck` checks are mounted. Next, the `init` process creates additional background processes as specified in the `/etc/inittab` file. (See Figure 3-12.) When all the background processes have started, multi-user initialization is complete, and if the `Autologin` file does not exist in `/mac/sys/Login System Folder/Preferences`, the Login dialog box is displayed. At this point, the system is ready for use.



Figure 3-12 The “Starting background processes” startup screen

Customizing your system

It is customary to give your server a name to distinguish it from other systems on your network. For your server to keep time correctly, it is also important to customize A/UX for the time zone in which your server is located.

Naming the system

The default name that A/UX gives to your system is `localhost`. You can rename the system by choosing another name. This name can contain up to 14 characters and must begin with a letter; it can include uppercase and lowercase letters as well as numbers and underscore characters.

A/UX and AppleShare configure the name of your system in different places. If you are only using AppleShare, and not using TCP/IP, you do not need to rename `localhost`.

There are two ways you can change this name. If you connect the system to a TCP/IP network, you are prompted for a system name, or host name, when you configure the network software with the `newconfig` command. You may have already done this when setting up your system. (See Chapter 6, “Creating and Managing a TCP/IP Network,” for more information.)

You can also name the system by creating or editing the file `/etc/HOSTNAME`. You must have `root` privileges to change this file. This file should contain two fields, separated by spaces or a tab. The name of your system is the word in the first field of this file, and the name of your domain is in the second field. (For information on what a domain is, see Chapter 6, “Creating and Managing a TCP/IP Network.”) Change the name in the first field and save the file. For example, to name the system `picasso`, the `/etc/HOSTNAME` file might look like this:

```
picasso      none
```

This change takes effect the next time you restart the system. For more information on system names, see `HOSTNAME(4)`.

Setting the system time

The A/UX operating system and the Macintosh Operating System normally keep track of the correct time without intervention. You need to set the time for A/UX in any of the following situations:

- You are first setting up your system, as described in *Setting Up and Managing Your Server*.
- You are bringing the system to another time zone. (You do not need to set the time when standard time changes to daylight-saving time.)
- The time is incorrect.

The system should always have the correct time. If you are logged in as `root`, you can set the time in A/UX (with the `date` command) and have it take effect for both A/UX and the Macintosh Operating System, or you can change the system time through the General Controls control panel. A/UX uses the time and date to monitor and control a number of activities. Without the correct time, the A/UX system cannot properly perform the following tasks:

- log system events, such as mail activity and logins
- record file activity, such as the creation, modification, or accessing of a file
- track file status with utilities such as `make`
- schedule system and user tasks, such as removing core files and making file backups
- run the Network File System (an optional package on the system software installer CD-ROM)

To change the time, use the `date` command; to change the time zone, use the `settimezone` command. See `date(1)` and `settimezone(1M)` in the manual pages for more information on these commands.

Single-user and multi-user modes

Unlike the Macintosh OS, A/UX has different levels of operation, called *run levels* (also called *run states* or *modes*). The system automatically starts up in multi-user mode, the preferred level for most activities.

The two basic run levels—single-user and multi-user—probably suffice for normal operation. Descriptions of these two levels follow. Other run levels are available, but are generally not used. See `init(1M)` for more information on the available states in which your A/UX system can operate.

Single-user mode

Single-user mode is one of several run levels available on the system. As its name implies, only one person has access to the computer. This access is available only from the System Console window. In single-user mode, most daemons are not active, so you cannot use printers, modems, or TCP/IP networking on the system. This state, called the *quiet state*, is necessary for the performance of administrative tasks. Here are two examples of such tasks:

- You are using `fsck` to check the integrity of file systems other than the root file system. (The root file system should be checked only from A/UX Startup.)
- You are copying user files to backup media by means of A/UX backup utilities.

If you want a Finder interface available while you are working in single-user mode, enter this command:

```
/mac/bin/mac32
```

Otherwise, you must work with the System Console window's command-line interface. The person with access to the System Console window while the system is in single-user mode has all the privileges of the `root` account. You can create or destroy anything on the system—files, directories, or programs—with just a few keystrokes. This level of power is necessary for administrative tasks, but you need to use it selectively. Unless you need the advantages of the quiet state of the system, run A/UX in multi-user mode. This strategy decreases the possibility of making a mistake that could damage the system, and avoids exposing the system to users who should not have the privileges of `root`.

To start A/UX in single-user mode manually, display the A/UX Startup shell window, as described earlier in this chapter. (See “Entering A/UX Startup.”) Enter the following command. (Note the option is a capital S.)

```
launch -S
```

To enter single-user mode automatically each time the system is started, you must edit the file `/etc/inittab`, or change the AutoLaunch Command entry in the A/UX Startup Booting dialog box to `launch -S`. You would do this only on a temporary basis to troubleshoot a specific problem with the system. Change the line

```
is:2:initdefault:
```

to

```
is:s:initdefault:
```

Notice that here a lowercase `s` is used.

If you are not sure whether you are in single-user mode or multi-user mode, enter the command `who -r` from CommandShell. If you receive the following output, denoting run-level 1, you are in single-user mode:

```
.  run-level 1      date      1      0      2
```

Multi-user mode

The default and preferred run level for A/UX is multi-user mode. This run level is recommended even if you are the only user on a machine, because it guards against inadvertent system damage. In multi-user mode, most system programs and activities are available.

To enter multi-user mode from single-user mode, enter on the command line

```
init 2
```

You should see

```
INIT: New run level: 2
```

If the `Autologin` file is configured, and this process is successful, the Finder environment is started. (If the `Autologin` file does not exist, the Login dialog box appears.) The system is now running in multi-user mode, and you can log in.

To go from multi-user mode to single-user mode, use the command-line form of the `shutdown` command.

The `Autologin` file

If a file called `Autologin` exists in the `/mac/sys/Login System Folder/Preferences` directory, and it contains a valid user name, then the Finder environment is started with the permissions of that user—that is, as though that user had logged into the system through the Login dialog box. Thus, by creating and editing the `Autologin` file, you can control the permissions available upon startup.

Initial processes: `/etc/inittab`

A/UX, like all UNIX systems, allocates system resources to users and programs through the use of processes. The process that the system creates upon startup, which has a process identification number of 1, is called `init`. (For more information on viewing processes, see `ps(1)`.) The following material provides an overview of some of the processes executed as the system starts. For more complete information, refer to the `init(1M)` manual page.

After the `init` process is created, it creates, or “spawns,” child processes. These child processes help `init` manage all the activities of the system.

One of the first activities `init` does is to check the file `/etc/inittab` for other tasks to perform. In A/UX, the next task is to execute the `/etc/sysinitrc` shell program, which performs basic functions such as setting the system’s time zone.

Next, `init` brings the system to the default initial run level. This is the level to which `init` will take A/UX unless interruptions are made while the system is starting. The line in `inittab` that shows the default run level is

```
is:2:initdefault:      #First Init State
```

Note that the 2 in this line denotes run level 2, which is multi-user mode. Once the initial run level is determined, `init` processes only those `inittab` entries whose *run-level* field is the same as the run level currently in effect.

If you view the contents of `inittab` on your system, you see that the following processes are invoked by `init`:

<code>/etc/bcheckrc</code>	A startup script that runs <code>fsck</code> on those file systems other than the root file system. See Chapter 10, “Checking Your Data Consistency,” for information on <code>fsck</code> .
<code>/etc/brc</code>	A startup script that sets the permissions on pseudo-terminals (used when logging in over a TCP/IP network).
<code>/etc/rc</code>	A startup script that mounts any non-root file systems and performs other general housekeeping tasks. For a discussion of file systems, see Chapter 10.
<code>/etc/getty</code>	A process that enables logins on serial ports and the console.

To understand more fully what each of these startup scripts accomplishes, you may want to look at them and try to understand where they fit into the A/UX Startup process (described in “The Startup Sequence” earlier in this chapter). This will aid your understanding of the system, and will help you to pin down the point of failure on startup, should a failure ever occur.

The format of the `inittab` file is discussed in `inittab(4)`.

Changing run levels: `init`

In addition to initial run-level selection, it is possible to change the run level by invoking `init` on the command line in `CommandShell`.

When running `init` from the command line, it is possible to disrupt the activities of users currently logged in. For instance, running the `init` command could disrupt an operating modem if you change the entry in `/etc/inittab` that manages the port the modem is connected to. Before making any changes to `/etc/inittab`, know your run levels and active processes, and make sure that changing them will not disrupt user activity. See `init(1M)`.

Once the system is started and running at the default run level, you can change the run level. As `root`, enter

```
init run-level
```

where *run-level* is an argument to `init` that may be either a value from 0 to 6, or the letters `s` or `S`. See `init(1M)`.

When you enter this command, `init` scans `/etc/inittab`, halts processes that should not be active in the new run level except for those that spawn daemons (see the discussion of `init 0` in the next paragraph), and activates those entries whose *run-level* value is the same as that of the new run level, leaving all other processes untouched.

If you make a change in the `inittab` file that you want the system to recognize, but do not want to change the run level of the system, enter

```
init 0
```

This command forces `init` to reread `/etc/inittab`. If you want to stop a process that is not controlled by an entry in `/etc/inittab`, use the `kill` command.

Configuring the kernel

Each UNIX system has a file that contains the kernel. The kernel is made up of a number of parts, or modules, each of which is responsible for managing a piece of the system. Some of these modules provide process scheduling, and others control aspects of the system such as disk drives or memory. The kernel is conventionally called `/unix` on A/UX and other operating systems that are derived from the AT&T System V version of the UNIX operating system.

It is not possible to view the kernel, which is a binary file, in the same way that you can view a text file, but you can obtain a list of the modules that are in the kernel. To do this from CommandShell, enter this command:

```
module_dump /unix
```

For each card installed in a NuBus™ slot, the kernel must contain a module that controls (or “drives”) the card. When you remove a card, the next time you start the computer, the system automatically builds a new kernel without

the absent card's module. If you change the slot position of the card, the system also automatically makes a new kernel when you restart the computer to reflect the new position.

When you add a NuBus card to the system, the system needs to be reconfigured to recognize the addition so that the hardware can be accessed. This is not done automatically. You must build a new kernel that incorporates the necessary modules with the `newconfig` command.

The `newconfig` command

The `newconfig` command provides an easy way to build a new kernel to include new features in the kernel or to support new hardware in the system. If new hardware is added to the system and there is no module in the kernel for that hardware, the system cannot use the hardware until you run `newconfig`. You must be logged in as `root` to run the `newconfig` command.

The `newconfig` program runs both the `newunix` and the `autoconfig` programs automatically. (See `newunix(1M)` and `autoconfig(1M)`.) The `newconfig` program takes predefined software modules and either installs them into or removes them from a new kernel. The new kernel is automatically called `/unix`. For a complete list of the available software modules and capabilities of `newconfig`, see `newconfig(1M)`. Note that other software manufacturers may add to the modules that `newconfig` can incorporate into the kernel, and that A/UX cannot use a Macintosh OS driver. It can only use drivers written for use with A/UX.

Note: You do not need to add a device driver to add terminals, printers, modems, built-in Ethernet port support, tape drives (DAT, nine-track, DCAS, QIC, and Apple Tape Backup 40SC), CD-ROM drives, hard disks, or Apple floppy disk drives to the A/UX system. Support for these devices is already built into the standard A/UX kernel. When adding peripheral devices not supplied by Apple, refer to the vendor documentation for installation of device drivers.

Before using `newconfig` to make a new kernel, be sure you have completed any hardware installation or modifications.

To run the `newconfig` program:

1 Check which modules are currently configured in the kernel.

Enter the command

```
module_dump /unix
```

Note that `newconfig` does not prevent you from making a kernel that does not work. Although A/UX provides a way to reconstruct a working kernel in case the new kernel fails, it is often easier to make a copy of the current kernel before running `newconfig`. Because the new kernel does not take effect until you restart the system, you may want to use the `module_dump` command to check the contents of the kernel both before and after running the `newconfig` command.

2 Run the `newconfig` command.

To add a new software module or driver to the kernel, specify the device driver or software module you want to add to the kernel as an argument to `/etc/newconfig`. For example, the command

```
/etc/newconfig -v debugger
```

installs the module for the A/UX kernel debugger support to the kernel. More than one module can be installed at a time; for example

```
/etc/newconfig -v nfs debugger
```

installs the Network File System and the kernel debugger support to the kernel.

To remove a software module from the kernel, specify the module that you want to remove as an argument to `newconfig` by putting the word `no` before the name of the module. Note that there is not a space between the word `no` and the module. For example,

```
/etc/newconfig -v nodebugger
```

removes the `debugger` module from the kernel.

To view any errors `newconfig` may report, you may want to use the `-v` option to `newconfig`:

```
newconfig -v
```

3 Restart the system to use the new kernel.

Choose Restart from the Special menu. Problems could occur if you continue to run the system after you have changed the kernel but have not yet started up with it.

Changing buffer cache size

In A/UX 3.0, the kernel variable `NBUF` was set to 0 to indicate that the amount of memory used to store disk files, while they are actively being used, should be 10 percent of available memory (excluding the memory used to store the kernel). This memory is called the *disk buffer cache*.

In A/UX 3.0.1, the kernel variable `NBUF` is set to -1 to allow the disk buffer cache to be up to 50 percent of memory, as long as at least 8 MB of memory is left available for use by the rest of the system. This allows the system to allocate memory dynamically for better system performance.

Although your Apple Workgroup Server 95 is shipped to you with default kernel settings suitable for most server applications, in some situations you may be able to increase performance by changing the buffer cache size. For example, if the server is only running applications that don't access disks through the cache (such as a database manager that accesses disks through raw devices), you can reduce the buffer cache to free up more memory for additional UNIX system processes. On the other hand, if the server is handling a few large files and users are experiencing slow response when they use the server, you can try to improve the processing speed of the server by increasing the buffer cache size.

With traditional UNIX operating systems, checking and changing the buffer cache size (a function of the kernel's `NBUF` and `SBUFSIZE` parameters) requires you to use commands such as `kconfig -a -n /unix | grep NBUF`. Unless you know exactly what you are doing, adjustments to the `kconfig` command can cause problems, because it's possible to set the buffer cache too low or too high. A/UX simplifies these adjustments by letting you change the buffer cache size through the Memory control panel, and the system ensures that the buffer cache size stays within acceptable limits.

You set the buffer cache in the Memory control panel, located in the Control Panels folder in the Apple menu. (See Figure 3-13.) You must be logged in as `root` in order to change the buffer cache size; however, any user can view the control panel settings.

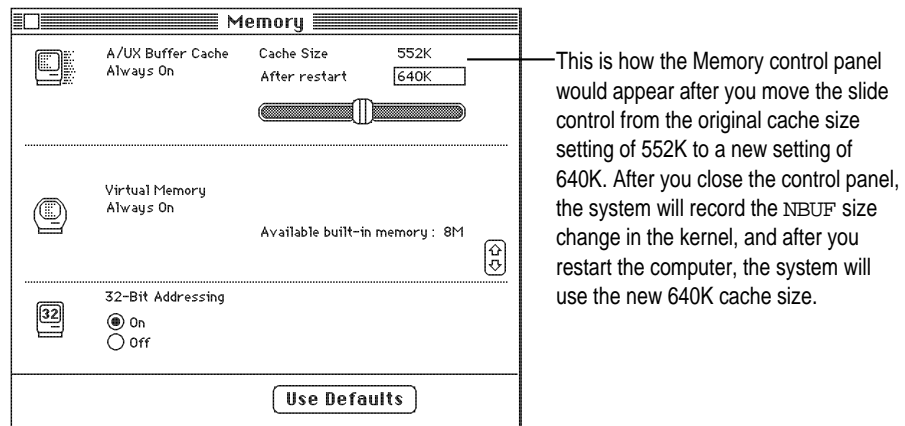


Figure 3-13 Memory control panel showing buffer cache adjustments

To adjust the buffer cache:

- 1 **Move the Cache Size slide control to adjust the size of the buffer cache, or click the Use Defaults button to use the preset size.**
- 2 **Restart the system to use the new buffer cache size.**

To restart the system, choose Restart from the Special menu.

Changing kernel parameters

Under normal conditions, kernel parameters do not need to be changed. The `newconfig` command will alter these parameters when it is building a kernel, based on the type of kernel it is building. Sometimes, however, you may exceed the kernel-configured limits on your system. Such limits are evident from error messages that appear. The following chart shows some of these messages.

Message	Parameter requiring an increase
inode: table is full	NINODE
file: table is full	NFILE
proc: table is full	NPROC
m_expand returning 0	NMBUFS

If you encounter one of these messages, use the `kconfig` command to increase one of these kernel parameters. When increasing these parameters, it is best to increase the value by 50 or 100 and then try running the system for a while. If the error message still occurs, increase the value a little more and then test again. It is possible to degrade the performance of the system if you increase these values too much. (See the `kconfig(1M)` manual page.)

For example, to increase the value of `NMBUFS` on the system:

- 1 **Check the current value of `NMBUFS` by entering this command:**

```
kconfig -av | grep NMBUFS
```

The output should appear something like this:

```
NMBUFS = 1000 ... numbers ... | Buffers for networking
```

- 2 **Enter the `kconfig` command:**

```
kconfig
```

You will not receive an additional prompt from the `kconfig` command. Enter the next command from the left margin:

```
NMBUFS = 1100
```

- 3 **Press `CONTROL-D` to stop running the `kconfig` command.**
- 4 **Check the new value of the parameter in the `/unix` file with this command:**

```
kconfig -av | grep NMBUFS
```

The output should now appear similar to this:

```
NMBUFS = 1100 ... numbers ... | Buffers for networking
```

5 Restart the system.

You must restart the system to put the new parameters into effect, because `kconfig` changes the parameters of the `/unix` file on the disk but does not change the parameters in the currently running kernel residing in memory.

Logging out, restarting, and shutting down

To log out, restart, and shut down the computer, use the choices available from the Special menu in the Finder.

WARNING When you finish using A/UX and you want to turn the power off or return to the Macintosh OS, you must use the Restart or the Shut Down menu item in the Special menu. If you fail to shut down the system in this way, you risk damaging or losing disk information. To shut down the system, you must be logged in as `root`, or you must know the password for the `root` account.

4

Backup Media and Devices

Backing up means copying the data on your hard disk to an alternative medium, such as a floppy disk or a magnetic tape. This allows you to restore the data to your hard disk later, if necessary.

The Apple Workgroup Server 95 comes with the product Retrospect Remote, from Dantz Development Corporation. This product allows you to back up your data through the Finder. For more information on how to use Retrospect, refer to the Retrospect documentation.

How often you perform backups and what you back up depend on your site's particular needs. For a complete discussion of the issues you need to consider to decide on a suitable backup strategy, see Chapter 7, "Managing Your Server," in *Setting Up and Managing Your Server*.

This chapter covers

- general information about different backup media
 - A/UX device filenames for backup devices
 - choosing and formatting tape cartridges for the server
-

Introduction

Your system software installer CD-ROM acts as your permanent backup copy of all the operating-system files as shipped. Therefore, you need to back up only files that you change or add to your Apple Workgroup Server 95.

After you have configured your root file system, you may want to perform an update backup (a backup of only the files that have changed). Then, with the CD-ROM disc of the initial system, plus your update backup, you will have a full initial backup of your customized root file system. You can do the initial update backup with either Retrospect Remote or CommandShell utilities. Usually, Retrospect Remote is considered the easiest to use of these choices.

A/UX includes standard UNIX backup utilities, which can be used through CommandShell—utilities such as `cpio`, `tar`, `dd`, `volcopy`, `pax` and `dump`. For more information on how to use these utilities, refer to their on-line manual pages.

Retrospect Remote limitations

In general, Retrospect Remote will back up any file on your Apple Workgroup Server 95. However, there are some exceptions:

- data outside the normal file system, such as those in raw disk partitions, boot blocks, and partition images
- multiple files and directories whose names are not unique in the first 31 characters
- symbolic links, except as a reference
- hard linked directories

Because Retrospect is a Macintosh application that utilizes the Macintosh File Manager to access files, it does not differentiate uppercase from lowercase. Thus, the file `FILENAME` appears to have the same name as `filename`.

IMPORTANT If you use raw disk partitions, such as those that can be created for databases to achieve faster data manipulation, you must back up those file systems with a utility that can understand raw disk partitions, such as `dd` or specific database utilities. See the manual page for `dd` for more information on that utility.

Be sure to refer to the Retrospect documentaion for further restrictions and details.

Backup media

You can make backups on a variety of media. A/UX 3.0.1 supports backups of files from both Macintosh (HFS) and UNIX file systems to floppy disks, hard disks, cartridge tapes, or nine-track magnetic tape. Remember to verify data on backed-up disks before you put them into storage. You can verify the data by trying to read it back from the backup disk.

When to use paper

For certain system files it is handy to have paper records of the contents of the files, as a further protection against catastrophic loss that might include the loss of your backups. Treat paper records with the same security precautions as your backups, as they are likely to contain confidential information. The following system files are candidates for paper copies:

```
/etc/fstab      /etc/hosts     /etc/inittab
/etc/group     /etc/passwd
```

Backing up files by dragging them to a floppy disk

Use floppy disks to store backups when

- backing up a small amount of data, such as several small data files
- making a daily backup, that is, backing up files changed that day

You can drag icons of files onto floppy disks to back them up, if you are backing up just a few files. You can also use Retrospect Remote and the `cpio`, `tar`, `pax`, and `dump.bsd` utilities to make backups on floppy disks.

Dragging file icons onto Macintosh-formatted disks is by far the easiest way to back up files. However, there are two limitations to this method. First, it is an effective process only if the files are relatively small. Second, A/UX files that are copied in this way do not retain their permissions or dates, so this method isn't recommended for files that are typically used in the A/UX file system, such as programs and shell scripts.

When these files are dragged back into the A/UX file hierarchy, they are owned by the currently logged-in user and given the permissions specified by the current `umask` value. (See Chapter 8, "User and Group Administration for A/UX Users," for a discussion of the `umask` parameter.) Dragging icons for files created by Macintosh applications should not result in permission problems.

Backing up files using tapes

Tapes can hold up to 5 gigabytes (GB) of data, depending on the technology used. Therefore, use tapes when you back up large amounts of data—for example, when you make a full backup of your system. Use tapes when

- making weekly or monthly backups
- backing up large amounts of data, such as entire file systems

Tape formats compatible with A/UX 3.0.1 include

- MRS-quality 3.5-inch DDS-DC DAT
- 1/4-inch QIC
- 8 mm
- nine-track

The optional DDS-DC DAT tape drive is fully integrated into the Apple Workgroup Server 95. Refer to your Retrospect documentation for more information about backing up to tape drives.

See "Preparing and Formatting a Tape Cartridge" later in this chapter for instructions on formatting a tape in the Macintosh OS.

Backup-device filenames

When you use most backup utilities, you must specify the location of the files you want to back up and where the backups should be recorded. In A/UX, all peripheral devices (hard disk drives, floppy disk drives, tape units, terminals, and so on) are known to the system through device files located in the `/dev` directory.

A particular device can be referred to by more than one device filename, which may refer to different capabilities or behavior, as shown in Table 4-1. For example, the floppy disk device `/dev/fd/d0e` ejects a disk after it has finished writing to or reading from the disk. The device `/dev/fd/d0w` waits for a disk to be inserted before writing to or reading from the disk. The disk device `/dev/fd/d0` automatically determines the density of a disk inserted into drive 0, and the device `/dev/fd/d0m800` ejects any disk that is not specifically an 800K disk.

There are two varieties of device files: block devices and character devices (also known as *raw devices*). Block devices use the kernel buffer cache mechanism, whereas character devices do not.

Whether you access the device as a block device or as a character device depends on the requirements of the utility you are using. Table 4-1 lists some examples of names of device files to which you can back up your files.

WARNING A user (especially the user of the `root` account) should never attempt to write data directly by using the name of a device file that accesses a disk or tape drive. Failure to use the appropriate utilities can result in complete data corruption. The tape-device filenames should be used only on backup-utility command lines.

Table 4-1 Standard A/UX device files

Device filename	Peripheral type	Device file type
<code>/dev/floppy0</code>	Floppy disk drive 0	Block device
<code>/dev/rfloppy0</code>	Floppy disk drive 0	Character device
<code>/dev/floppy1</code>	Floppy disk drive 1	Block device
<code>/dev/fd/d0</code>	Floppy disk drive 0	Block device
<code>/dev/fd/d0e</code>	Floppy disk drive 0, eject when done	Block device
<code>/dev/rfd/d1</code>	Floppy disk drive 1	Character device
<code>/dev/rfd/d1ew</code>	Floppy disk drive 1, wait for disk, eject when done	Character device
<code>/dev/dsk/c8d0s0</code>	Floppy disk drive 0	Block device
<code>/dev/rdsk/c8d0s0</code>	Floppy disk drive 0	Character device
<code>/dev/dsk/cxdysz</code>	Hard disk SCSI ID <i>x</i> , drive <i>y</i> , slice <i>z</i>	Block device
<code>/dev/rdsk/cxdysz</code>	Hard disk SCSI ID <i>x</i> , drive <i>y</i> , slice <i>z</i>	Character device
<code>/dev/dsk/cb0xdysz</code>	Hard disk on SCSI bus <i>b</i> , SCSI ID <i>x</i> , drive <i>y</i> , slice <i>z</i>	Block device
<code>/dev/rdsk/cb0xdysz</code>	Hard disk on SCSI bus <i>b</i> , SCSI ID <i>x</i> , drive <i>y</i> , slice <i>z</i>	Character device
<code>/dev/rmt/tcx*</code>	Tape drive SCSI ID <i>x</i>	Character device
<code>/dev/rmt/tcxn</code>	Tape drive SCSI ID <i>x</i> , no rewind	Character device
<code>/dev/rmt/tcx.1600</code>	Tape drive SCSI ID <i>x</i> , 1600-bpi density	Character device
<code>/dev/rmt/tcx.6250</code>	Tape drive SCSI ID <i>x</i> , 6250-bpi density	Character device

* `/dev/rmt/tcx` is shipped as SCSI ID 2 on SCSI bus 3 (internal PDS card).

As shown in Table 4-1, you can use `/dev/dsk/c8d0s0` to refer to floppy disk drive 0. This nomenclature reflects the SCSI naming scheme. Devices in the `/dev/dsk` and `/dev/rdsk` directories are named according to their SCSI bus, SCSI ID, subdrive, and slice number in the following format:

`/dev/dsk/cb0xdysz`

The value of *b* is the number of the SCSI bus (1–4). The value of *x* is the SCSI ID of the hard disk; the value of *y* is the number of the subdrive at that SCSI ID; and the value of *z* is the slice number associated with a particular disk partition. Subdrives are usually used only for floppy disk drives.

Note: The SCSI buses are numbered as follows:

SCSI bus 1	Main logic board internal bus
SCSI bus 2	Main logic board external bus
SCSI bus 3	PDS card internal bus
SCSI bus 4	PDS card external bus

For example, `/dev/dsk/c100d0s0` signifies SCSI ID 0, the first drive, and the first assigned UNIX partition on the first bus. The same partition on an external hard disk connected to the main logic board external bus with SCSI ID 5 would be referred to as `/dev/dsk/c205d0s0`. See *Setting Up and Managing Your Server* for more information on the SCSI naming scheme.

Tape devices and scanners do not use the new SCSI naming scheme. This means that either they should be given unique SCSI IDs or they should be connected to the external bus on the main logic bus, so that they are scanned before either of the PDS card buses.

Preparing and formatting a tape cartridge

Formatting a tape cartridge prepares it to store data. You can format tapes in A/UX by using the three steps listed in this section. It is a good idea to format all unformatted tape cartridges as soon as you receive them.

Note: If you have Apple tape cartridges, you can skip this section, because Apple tape cartridges are formatted at the factory for you.

If you run out of formatted tape cartridges while you are backing up a disk, you'll have to stop the backup procedure, format a tape, and begin the entire backup procedure again.

The process of formatting a tape cartridge with the 3.5-inch DDS-DC DAT drive usually lasts at least 30 minutes and must not be interrupted.

To format a tape cartridge:

1 Insert the tape cartridge into the internal 3.5-inch DDS-DC DAT drive.

When you insert the tape cartridge, make sure that the record-lock switch is in the unlocked position (to the left).

Push firmly until the tape cartridge is inside the internal drive.

It takes about a minute for the tension to be adjusted.

2 Display a CommandShell window by choosing CommandShell from the Application menu.

If a CommandShell window does not appear, choose New from the File menu.

3 Enter the following command:

```
mt -f /dev/rmt/tcx format
```

where *x* is the SCSI ID number of the tape unit.

Selecting tapes for use with the internal tape drive

Your server comes with one tape for use with the 3.5-inch DDS-DC DAT drive. Make sure all the tapes are of Media Recognition System (MRS) quality. The DDS-DC DAT drive can back up only to MRS-quality tapes. These tapes are the highest-quality digital data storage tapes available. To acquire these tapes, contact your Apple service provider.

Cleaning your tape drive

Your tape drive requires periodic cleaning. The frequency of your cleaning schedule depends on how often you use the tape drive. In general, the drive should be cleaned as follows.

Backup schedule Cleaning schedule

Daily	Weekly
Weekly	Monthly

Please refer to the cleaning instructions provided with the cleaning cartridge tape for more information. To acquire cleaning cartridge tapes, contact your Apple service provider.

5

Managing Disks

Regardless of make or size, any hard disk becomes full over time unless the disk space is managed and cleaned out occasionally. If a disk becomes completely full, the system ceases to function correctly until space is made available. This chapter tells you how to be aware of the space used on your server, as well as other disk information for floppy disks and CD-ROM discs.

This chapter provides information about

- mounting and unmounting file systems
 - changing the name of a disk when it is mounted
 - using CD-ROM discs with the system
 - managing the data on your disks
 - initializing Macintosh-formatted and UNIX data floppy disks
-

Mounting file systems with UNIX commands

If you are familiar with UNIX and prefer to mount file systems with UNIX commands, you can mount UNIX file systems by using the command `fentry`. This command makes an entry in the `/etc/fstab` file so that A/UX 3.0.1 mounts the file system at startup.

To mount a file system with the `fentry` command:

1 Choose CommandShell from the Application menu.

If a CommandShell window does not appear, choose New from the File menu.

2 Enter the `fentry` command with the following options and arguments:

```
fentry -p 2 -t type block-device mount-point
```

In the `fentry` command line, the command-line arguments are as follows:

<code>-t type</code>	The type of file system: Enter 4.2 for UFS, 5.2 for SVFS, or <code>nfs</code> for NFS.
<code>block-device</code>	The file system to be mounted; for example, <code>/dev/dsk/cx0yd0sz</code> , where <i>x</i> specifies the bus (1–4), <i>y</i> is the SCSI ID number (0–6) of the hard disk that contains the file system, and <i>z</i> is the slice number (usually 0–6, inclusive—never use slice 30 or 31).
<code>mount-point</code>	The full pathname of a directory on the local machine that is to be used as a mount point. The <code>fentry</code> command creates this directory if it does not exist.

For example,

```
fentry -p 2 -t 4.2 /dev/dsk/c405d0s3 /usr
```

creates a file-system table entry that automatically mounts the file system located at slice 3 of the SCSI drive on bus 4 at address 5 (`/dev/dsk/c405d0s3`) to the mount point directory `/usr`.

Note: You can make one entry each time you use the `fentry` command. If you have more than one UNIX file system on the disk, use the `fentry` command once for each file system.

Unmounting file systems

To disassociate a file system from its mount point by using the `umount(1M)` command:

- 1 **Choose CommandShell from the Application menu.**

If a CommandShell window does not appear, choose New from the File menu.

- 2 **Enter the `umount` command as follows:**

```
umount mount-point
```

In this command line, *mount-point* is the directory to which this file system was mounted.

Note: The spelling of this command is `umount`, *not* `unmount`.

The error message

```
source: Device busy
```

indicates that you or other users are accessing files or directories on the file system mounted on *source*. If you are in single-user mode or are the only user on the system, simply change your current directory to a directory that is not on this file system and reenter the `umount` command. If other users are accessing this file system, ask them to finish their work and change their current directory to another so that you can unmount the file system.

If you change a disk's bus or SCSI ID

Mounting a file system in Apple HD SC Setup creates an entry in the file-system table, `/etc/fstab`. The file system is then automatically mounted each time the system starts. If you subsequently move the drive to another bus or physically change its SCSI ID, you must change the file system's entry in `fstab`. This allows it to be correctly mounted at startup.

To change the file system's entry in `fstab`:

- 1 **Open the file `/etc/fstab` with your favorite text editor.**
- 2 **Edit the file-system entry for each file system on the disk you are moving.**

Locate the file-system entry for the disk you are moving to a different bus or whose SCSI ID you are changing. The device is identified as

`/dev/dsk/cx0yd0sz` where *x* specifies the bus (1–4), *y* is the SCSI ID number (0–6), and *z* is the slice number.

Select the bus or the SCSI number (or both) that you are changing, and type the new address. If the disk contains more than one UNIX file system, be sure to update each entry.

The entry in `/etc/fstab` for a file system looks like this:

```
/dev/dsk/c401d0s0    /users    4.2    rw    1    2
```

This entry mounts the file system located at slice 0 of the SCSI drive on bus 4 at address 1.

Note: If you are *removing* the hard disk from your system, delete the entry for each file system that is located on the disk, or you will get an error message at startup.

3 Save and close the file.

Choose Save from the File menu, and then choose Close from the File menu.

4 Restart the computer by choosing Restart from the Special menu.

A/UX 3.0.1 mounts the file systems at startup, according to the current entries in the file-system table.

Changing the name of a mounted volume

If you are familiar with the Macintosh OS, you may know that you are able to change the name of an item on your desktop by selecting the text and typing a new name. However, A/UX will not let you change the name of mounted volumes in this way.

To change the name of a mounted volume:

1 Create a new folder in the /Shared Data folder, and give the folder the new name of the volume.

2 Enter this command:

```
umount old-volume-name
```


3 Edit the file `/etc/fstab` to reflect the new name.

Change the name in this file from the old mount name to the new mount name. For example, if you changed the name `applications` to `utilities`, make this change in `/etc/fstab`.

4 Enter this command:

```
mount new-volume-name
```

The volume is now mounted under the new volume name.

Using an Apple CD-ROM drive

An Apple CD-ROM drive is extremely useful for accessing large amounts of information stored on CD-ROM discs. (CD-ROM is an acronym for *compact disc read-only memory*.) The information on a CD-ROM can be read, but it cannot be added to, changed, or deleted. A CD-ROM can hold more than 500 MB of data, supplying six times the storage capacity of a commonly used 80 MB hard disk.

Although you cannot make changes to the information on a CD-ROM, you can read or copy files to your hard disk and make changes to the information there. Because the CD-ROM is an excellent means for storing reference material or delivering large amounts of information or huge programs, more products and Macintosh-related materials are likely to be available on CD-ROM discs in the near future.

Identifying the file system on a CD-ROM disc

Your Apple CD-ROM drive supports CD-ROM discs that use Macintosh, ISO (International Standards Organization), and UNIX file systems. The following three sections discuss how to use each kind of CD-ROM. You need to review only the section that pertains to your particular kind of CD-ROM.

If you're unsure about what kind of CD-ROM you have, and if you're unable to get this information from your supplier, check the disc jacket for information. For example, CD-ROM discs that use ISO file systems are typically identified as such. As a final resort, you can attempt to mount the disc in the Macintosh OS without A/UX running. (See the next section, "Using

CD-ROM Discs Containing Macintosh File Systems.”) If its icon does not appear on the desktop, try mounting it as a UNIX disc. (See “Using CD-ROM Discs Containing UNIX File Systems,” later in this section.)

Using CD-ROM discs containing Macintosh file systems

To work with a CD-ROM containing Macintosh applications and files, insert the disc into the CD-ROM drive. The disc’s icon will appear on your desktop much as the icon of a Macintosh disk appears on your desktop. Open the disc by double-clicking its icon or by selecting the icon and choosing Open from the File menu. With the CD-ROM open, you can work with applications and files on it without any further handling. To eject the CD-ROM, drag its icon to the Trash.

Using CD-ROM discs containing ISO file systems

A/UX 3.0.1 contains support for ISO 9660 and High Sierra file systems on CD-ROM discs. You do not need to add modules in the kernel in order to use discs containing applications and files in an ISO file system when you run A/UX 3.0.1.

Using CD-ROM discs containing UNIX file systems

This section describes the choices you have when setting up an Apple CD-ROM drive for CD-ROM discs containing UNIX file systems. Choose the setup scheme you want and follow the instructions for implementing that scheme.

A/UX and audio CDs: A/UX cannot recognize the sound portion of a disc as the Macintosh can. Therefore, if you try to play an audio-only CD, A/UX will display an alert message and you won’t be able to hear anything from the CD. If you try to access a disc containing sound and data tracks, A/UX will access the data but not the sound. Do not install an audio-CD driver in A/UX, because it will cause a system error.

Installing CD-ROM discs permanently on an A/UX file system

This section describes how to install an Apple CD-ROM drive so the files on a CD-ROM are readable whenever A/UX is in use.

The steps in this section take you through using the `fsentry` script, which makes an entry for the CD-ROM in the file-system table `/etc/fstab`. This file contains a list of all the file systems that A/UX automatically mounts each time it starts.

If you want to access the CD-ROM only temporarily, for example to copy its files to a hard disk, then mount it only when you need it rather than mounting it permanently. See the next section, “Installing CD-ROM Discs Temporarily on an A/UX File System,” for instructions.

To permanently install a CD-ROM:

- 1 Log in as `root` and open CommandShell from the Application menu.**

If a CommandShell window does not appear, choose New from the File menu.

- 2 Enter this command:**

```
mkdir /cdrom
```

This command creates a directory called `cdrom`. In the next steps, you will mount the CD-ROM at this directory.

- 3 Type the `fsentry` command to mount the CD-ROM disc.**

Note: The documentation accompanying your CD-ROM states whether the files on the disc are divided among different slices. Most likely, all the files are contained in one slice. For a disc that keeps all its files in one slice, slice 0 is likely to be the slice used.

- 4 Press RETURN.**

This action runs the command. The files on the CD-ROM are now accessible through the `cdrom` folder and will be mounted automatically each time you restart A/UX.

Installing CD-ROM discs temporarily on an A/UX file system

If you need to refer to the information on the CD-ROM temporarily, meaning from now until you next shut down A/UX or unmount the CD-ROM, then mount its files by using the `mount` command.

Note: If you mount the CD-ROM with the following steps and then restart the system or move the system into single-user mode and back into multi-user mode, you will unmount the CD-ROM.

To temporarily install a CD-ROM:

- 1 **Log in as `root` and open CommandShell from the Application menu.**

If a CommandShell window does not appear, choose New from the File menu.

- 2 **Type this `mount` command to temporarily mount the CD-ROM:**

```
mount -ro /dev/dsk/cx0yd0sz mount-point
```

- 3 **Press RETURN.**

This action runs the command, and the CD-ROM is mounted. To the user, there is no difference between reading the files on the CD-ROM and reading files on a hard disk.

Opening files

Once the CD-ROM is mounted, you can access the files on it just as if they were on a local disk. The only difference is that the access to the files is read-only. If you wish to change the files, you must first copy them to a hard disk.

Opening files from the Finder

To open files on a CD-ROM from the Finder:

- 1 **Switch to the Finder.**

Choose Finder from the Application menu if the Finder isn't already active.

- 2 **Double-click the `/` disk icon, then double-click the folder that contains the files on the CD-ROM.**

For example, if you mounted the CD-ROM on `/mnt`, double-click the `mnt` folder; if you mounted the CD-ROM on `/cdrom`, double-click the `cdrom` folder.

The folder opens and shows the files on the CD-ROM.

3 Double-click a file to open it.

If you want to copy files from the CD-ROM to a hard disk, see “Copying Files to Disk” later in this chapter.

Opening files from CommandShell

To open text files on a CD-ROM from a CommandShell window:

1 Choose CommandShell from the Application menu if a CommandShell window isn’t already displayed.

If a CommandShell window does not appear, choose New from the File menu.

2 Enter this command:

```
cd /directoryname
```

Directoryname is the name of the directory on which you mounted the CD-ROM.

For example, if you mounted the CD-ROM on `/mnt`, enter `cd /mnt`; if you mounted the disc on `/cdrom`, enter `cd /cdrom`.

3 Enter this command:

```
ls -C
```

The files in the current directory are listed in columns.

4 Open a file using your favorite text editor.

Remember that you can read the files, but you cannot make changes (write) to them.

Copying files to disk

By copying files from a CD-ROM to a hard disk, you can then make changes to the files.

Files meant to be copied from a CD-ROM to a hard disk can be in one of several formats:

- a UNIX file system
- an archive created with the `tar` utility
- an archive created with the `cpio` utility

The next three sections describe how to copy files of each of these formats. To determine the format of the files on a CD-ROM, read the information accompanying the disc. The documentation for the disc explains the format of the files and how to list and copy them.

Files in a UNIX file system

To copy files that are in the form of a UNIX file system:

- 1 **Log in as `root` and open CommandShell from the Application menu.**
- 2 **Change the directory to `/usr/local` by entering this command:**

```
cd /usr/local
```

In the next few steps you create a new folder within the `/usr/local` folder to hold the files from the CD-ROM. If you like, you can put the CD-ROM files in another directory. However, putting them in the `/usr/local` folder makes them easily accessible to all user accounts on the system. The `/usr/local` directory is provided in A/UX to hold programs that you add for use with A/UX—for example, public domain software.

- 3 **Make a directory by using this command:**

```
mkdir filename
```

- 4 **Enter this command:**

```
cp -r /mnt /usr/local/filename
```

If you mounted the CD-ROM at a directory other than the `/mnt` directory, enter that directory instead.

The files on the CD-ROM, which you mounted temporarily at the `/mnt` directory, are now in the new folder.

Files in a `tar` archive

To copy files that are in the form of a `tar` archive:

- 1 **Log in as `root` and open CommandShell from the Application menu.**

2 Enter this command:

```
mkdir /usr/local/directoryname
```

Here *directoryname* is the name of the new directory for the files you are copying from the CD-ROM.

3 Enter this command:

```
cd /usr/local/directoryname
```

Your current directory is now */usr/local/directoryname*.

4 If the CD-ROM contains a `tar` archive but does not contain a file system, enter the following command. Otherwise, skip to step 5.

```
tar xvf /dev/dsk/cnd0sz
```

In this command, *n* is the SCSI ID number of the CD-ROM drive and *z* is the number of the slice containing the `tar` archive you want to copy.

For example, if your CD-ROM drive is SCSI ID 5 and the file system is slice 0, the name you want to select is *c5d0s0*. If your CD-ROM drive has SCSI ID 4 and the file system is slice 2, the name you want to select is *c4d0s2*.

The files are copied into */usr/local/directoryname*, and you are finished with this procedure. If an error message appears on your screen, see the manual page for the `tar` command. You may need to specify a block size in units of 512 bytes.

5 If the CD-ROM contains a file system along with a `tar` archive, mount the file system and then copy its files by using the following command.

```
mount -ro device /mnt
```

In this command, *device* refers to the name of the device, in the form */dev/dsk/cnd0sz*, where *n* is the SCSI ID number of the device and *z* is the number of the slice containing the files you want to copy.

For example, if the device is SCSI ID 5 and the file system is on slice 0, type

```
/dev/dsk/c5d0s0
```

After you run this command, the CD-ROM is mounted.

6 To copy the files from the CD-ROM, enter this command:

```
tar xvf /mnt/archivename
```

In this command, *archivename* is the name of the directory on the CD-ROM containing the files you want to copy.

The files are copied to */usr/local/directoryname*.

Files in a `cpio` archive

To copy files that are in the form of a `cpio` archive:

1 Log in as `root` and open CommandShell from the Application menu.

2 Enter this command:

```
mkdir /usr/local/directoryname
```

Replace *directoryname* with the name of the new directory for the files you are copying from the CD-ROM.

3 Enter this command:

```
cd /usr/local/directoryname
```

Your current directory is now */usr/local/directoryname*.

4 Enter this command:

```
cpio -ivd < /mnt/archivename
```

Here *archivename* is the name of the directory on the CD-ROM containing the files you want to copy.

The files on the CD-ROM are now in */usr/local/directoryname*.

Removing the CD-ROM discs

If you are using discs containing Macintosh file systems or ISO file systems, you need only to drag the CD-ROM icon to the Trash, or to shut down A/UX before removing the CD-ROM.

If you are using discs containing UNIX file systems, you need to use one of the following procedures, depending on how you mounted the CD-ROM drive.

If the CD-ROM was mounted permanently

If the CD-ROM was mounted with `fseentry`, use the following steps to remove it from A/UX.

Note: You cannot unmount a file system if any of its files is in use. This means that if you are in a directory in the file system that you are trying to unmount, the `umount` command will fail. It also means that if you share the A/UX system with someone and, while you are attempting to unmount the CD-ROM, that person has a file from the CD-ROM displayed on screen or is using one of its files in some other way, your command to unmount the CD-ROM will fail.

To remove a permanently mounted CD-ROM:

- 1 **Log in as `root` and open CommandShell from the Application menu.**

- 2 **Enter this command:**

```
umount /cdrom
```

Note that the command is spelled `umount` and *not* `unmount` as you might expect. This command unmounts the CD-ROM drive.

- 3 **Edit the `/etc/fstab` file. Delete the entry for the CD-ROM drive.**

It is probably the last line in the `/etc/fstab` file and looks similar to this line:

```
/dev/dsk/c5d0s0 /cdrom 4.2 ro 0 2
```

- 4 **Save the change and close the file.**

If the CD-ROM was mounted temporarily

To unmount a CD-ROM that is mounted temporarily:

- 1 **Log in as `root` and open CommandShell from the Application menu.**

- 2 **Enter this command:**

```
umount mount-point
```

Note that the command is spelled `umount` and *not* `unmount` as you might expect. This command unmounts the CD-ROM.

Reclaiming disk space

You can make more room available on your disk in several ways. You can remove the on-line manual pages from the Apple Workgroup Server 95 and read them from the CD-ROM. You can compress infrequently used files to reduce the space required for storing them on the disk, and expand them again on demand. You can back up and remove data that you haven't used in a while and don't need. It is also prudent to trim files (for example, log files) that tend to grow over time. To be on the safe side, be sure to make a backup copy of anything you decide to remove.

Removing unneeded files

The `find` command can help you locate and remove unneeded files from your system. Some files, such as those named `core`, are produced by isolated system events. Unless your system is crashing repeatedly, they should be removed.

For example, to remove all the files named `core`, enter this command:

```
find / -name core -exec rm '{}' \;
```

Large, regular files, as opposed to block or character devices, in the `/dev` directory have usually been created there by accident (for example, if a user directed output to a nonexistent device).

To remove regular files larger than about 10K from the `/dev` directory:

1 Locate the files you want to remove.

Enter this command:

```
find /dev -type f -size +20 -print
```

Note: In this first step you use the `find` command with the `-print` option instead of the `-exec` option. The `-exec` option actually executes the command that follows it. By printing the files to your screen before removing them, you can verify that you will actually remove the files you *want* to remove.

2 Verify that you actually want to remove all the files located.

If you want to keep any of the files, move them to a directory that is not in the search path of the `find` command. In the command given in step 1, the search

path of the `find` command is `/dev`, so you will want to move them out of `/dev`. A good place to move them while you remove the other files is `/usr/tmp`.

3 Remove the files by entering this command:

```
find /dev -type f -size +20 -exec rm '{}' \;
```

4 Replace any files moved in step 2.

Return the files from `/usr/tmp`, or wherever you moved them, back to `/dev`.

Some additional examples of files that may not be needed are as follows:

- `/lost+found/*`—unlinked files salvaged by `fsck` can be removed when you have made sure that all needed files have been restored
- `/usr/preserve/*`—files from interrupted or damaged `ex` or `vi` sessions
- multiple copies of graphics or text files that can be re-created from a source file

Note: When removing files, be careful not to remove the parent directory of the files being removed. For example, you may remove all the files in the directory `lost+found`, but don't remove the `lost+found` directory itself.

Trimming files that grow

As A/UX runs, it creates assorted log and data files and appends data to them. If no action is taken, these files can eventually grow to a substantial size. Fortunately, it is easy to monitor the growth of these files, and you can automate much of the task of reducing their size. (See “Automating Server Administration With the `cron` Command,” later in this chapter, for information on automating tasks.) Note that some log files are written into only if they exist (that is, they are not re-created if absent). Removing such files causes no error condition, but it will disable the logging activity. You can truncate such files rather than deleting them.

Other files are produced by the system to log events of administrative interest. These files should normally be trimmed of their earlier entries. For instance, to trim the file `/usr/adm/messages` to its last 20 entries, enter these commands:

```
tail -20 /usr/adm/messages > mgs.tmp  
mv mgs.tmp /usr/adm/messages
```

Some log files have nonreadable records in them but can still be trimmed if you know the length of each record. (This information is in the on-line manual pages, under the name of the file.) For example, the files `/etc/wtmp` and `/etc/utmp` both have record lengths of 52 characters. These files can be trimmed to some multiple of their record size. This example shows how to trim the file `/etc/wtmp` to its last 20 entries (52 times 20 equals 1040 characters):

```
tail -1040c /etc/wtmp > w.tmp
mv w.tmp /etc/wtmp
```

Finally, some files must be present, such as log files, but can be truncated to zero length periodically. To truncate a file to zero length, you copy `/dev/null` over the contents of the file. For instance, to truncate the file `/usr/adm/messages`, enter this command:

```
cp /dev/null /usr/adm/messages
```

Another example of a file that can periodically be truncated is `/usr/adm/lpd-errors`, the output from the `lpd` daemon.

Compressing infrequently used files

The tools available on A/UX for compressing files are `compact`, `compress`, and `pack`. These tools reduce the size of files, saving disk space. The amount of disk storage that you gain from such compression can be substantial, but it takes longer and is more difficult to access the compressed versions than to access noncompressed files.

Only large and infrequently used data files are suitable candidates for compression. For example, in A/UX, the manual page files are shipped in compressed form, saving several megabytes of storage. The `man` command has been specially written to deal with compressed files; however, the execution time of the command is longer due to extra decompression time.

All of these compression tools reduce the size of text files by about 50 percent. On binary files, however, compression is lessened. The `compress` utility saves about 40 percent, and the other two programs save only about 25 percent of the original storage. Therefore, `compress` gives you the best compression size when files other than text (ASCII) files are involved.

There are also differences in the time the utilities take to run. On text files, `pack` is slightly faster than `compress`. On binary files, `compress` is slightly faster. In both cases, however, `compact` takes about ten times as long as the faster of the other two. Clearly, `compact` should be used only when compatibility with another machine requires it. Non-A/UX file-compression utilities can be used; these utilities, however, may not preserve file permissions or ownership, unlike the A/UX utilities described in this section. Unless a compression utility is certified to work with A/UX, it should be used with caution.

To search for files larger than about 100K and compress the ones that you select:

1 Locate the files you want to compress using this command:

```
find / -type f -size +200 -print
```

(You can change the options used in this command.)

2 Verify that you actually want to compress all the files located.

If you don't want to compress some of the files, back them up or move them to a directory that is not in the search path of the `find` command. In the command in step 1, the search path of the `find` command is `/`, so you will need to move the files onto another disk.

3 Compress the files by entering this command:

```
find /dev -type f -size +200 -exec compress {} \;
```

4 Restore any files moved or backed up in step 2.

Usage notes

All three compression programs remove their input, or original, files during the compression process. The companion decompression programs (`uncompact`, `uncompress`, and `unpack`) do the same. You should therefore make a copy before compression (or decompression) if the original version is to be retained.

Alternatively, the appropriate “snapshot” decompression program (`ccat`, `zcat`, or `pcat`, respectively) can be used. These tools are analogous to the `cat` command in that they display the uncompressed version of ASCII data on the screen while leaving the compressed file in place. The `man` command thus uses `pcat` for compressed manual pages and `cat` for uncompressed ones.

Extensions to names of compressed files

By default, the compression programs append a two-character suffix to the names of their output files. For example, the `compress` utility converts the file `sample` into the compressed file `sample.z`. If the input filename exceeds 12 characters, an error message tells you that the filename is too long. (This limitation is implemented in order to support System V file systems.)

Compressing an archive of files

You may sometimes need to compress a collection of small files. You can achieve a substantial saving in storage by first producing a `cpio`, `tar`, or `pax` archive, then compressing the archive. Note, however, that the same archiving utility is needed after the decompression process when you decompress the archive.

Automating server administration with the `cron` command

As a server administrator, you must do certain tasks periodically. Some of these tasks, such as making backups onto removable media, may require your direct interaction. Most, however, can be run automatically by the A/UX `cron` process. Management of disk space can easily be automated with `cron`.

There are several benefits to using `cron`. For example, `cron` cannot forget to perform its functions, as a human might; it never goes on vacation or takes sick leave; and it is quite willing to do things at awkward times, such as 2:00 A.M. Also, `cron` can do things very frequently, as often as once a minute. Of course, in order for `cron` to complete its assignments, the system must be left running.

Instructions to `cron` can consist of single commands or shell scripts. The commands, and the definitions of when the commands are to be performed, are located in the file `/usr/spool/cron/crontabs/user`, where `user` is the login name of the user on whose behalf the command will be run. These files are called `crontab` files. For example, the `crontab` file `/usr/spool/cron/crontabs/adm` controls the timing of the various accounting functions.

User access to `cron`

Once an activity has been chosen for automation by means of `cron`, the system administrator must decide who is to perform the action. The `cron` daemon accords a command the same privileges as the user running the command. Since a user logged in as `root` has so few restrictions, `cron` commands should typically be put in the `crontab` file of some less powerful account.

To allow users to use the `cron` facility, add the names of such users to the file `/usr/lib/cron/cron.allow`. You can add either user login names, such as `joe`, or administrative login names, such as `oracleadm`.

If `cron.allow` is empty or absent, then the system checks a file called `cron.deny`. This file lists users who are denied access to `cron`; again, you enter names into the file. It is your choice whether to use `cron.allow` or `cron.deny` to control access to `cron`; you should not use both. If neither file exists, only the `root` account is allowed to use `cron`.

In general, using the `cron.allow` file is more secure than using `cron.deny`, because the administrator must actively approve each user who might wish to use `cron`. If it is more convenient to use `cron.deny` and an administrator feels that a site's security needs are low, this option is available. Lack of both files is appropriate for sites where the `root` account runs `cron` commands on behalf of other users.

The `crontab` file format

The `cron` utility uses `crontab` files to control the execution of programs. Each line of text in this file consists of six fields that together specify a process to be executed at a certain time. The fields are separated by spaces or tabs. The first five fields specify a time. They specify, in order,

<i>minute</i>	Time in minutes (0–59)
<i>hour</i>	Time in hours (0–23)
<i>day-of-month</i>	Day of the month (1–31)
<i>month-of-year</i>	Month of the year (1–12)
<i>day-of-week</i>	Day of the week (0–6, with 0 as Sunday)

Each of these fields may contain either a list of elements separated by commas, which specify specific cases to be activated, or an asterisk (*), which means ignore all cases. An element may be one number or two numbers separated by a hyphen (-), meaning an inclusive range.

You specify days in two fields: *day-of-month* and *day-of-week*. If you specify both, both are used. To specify days by using only one field, set the other to *.

The sixth field is the process to be executed. A percent sign (%) in this field, unless surrounded by quotation marks or preceded by a backslash character (\), is translated as the signal for a new line. The shell executes only the first line (up to the % or the end of the line).

For example, you might specify this line:

```
15 3 1 * * find / -mtime -31 -print > /usr/tmp/month_files
```

This line means that at the 15th minute of the third hour (that is, 3:15 A.M.) on the first day of each month, the system will list all the files that have changed in the last 31 days into the file `/usr/tmp/month_files`. You can then use this information to back up only the files that have changed.

The asterisks in the *month-of-year* and *day-of-week* fields tell the system to match all options—that is, no matter which day of the week the first day of the month is—and print to the file every month.

Adding cron commands

Any user with cron privileges can schedule commands to be run by cron. Before adding a command, it's a good idea to test its action.

To add a cron command:

- 1 Test the desired command, using a fresh login session.**

You can use the substitute user command (`su user`) for the account that cron will emulate. Alternatively, you can log in as *user*.

2 Execute the desired command just as it will appear in the `crontab` file.

This ensures that the permissions, command format, and user environment are the same as those used by `cron`. If the command fails, try running the command again when logged in as `root`. If the command then works, the problem may have been that the user did not have the correct permissions to perform the task. If the command fails when you are logged in as `root`, there is probably something wrong with the command itself.

3 Insert the command in a file with a specification of when `cron` is to run the command.

See “The `crontab` File Format,” earlier in this chapter, for information on how to define the time at which the command is to be run.

4 Install the file into the user’s `crontab` file.

While logged in as `user` (or impersonating `user` by means of `su`), enter this command:

```
crontab file
```

In this command line, *file* is the name of the file you created in step 4. This copies the file into `/usr/spool/cron/crontabs/user` and alerts `cron` to reread the `crontab` file. The value of *user* is determined by your login name or the login name to which you have changed by using the `su` command—whichever is most recent.

Monitor the results of a few executions of the automated activity to make sure the operation is working properly. Also, `cron` will send you mail when it cannot execute something in the `crontab` file.

Changing `cron` commands

To change `cron` commands, you replace the `crontab` file with a new version.

1 Copy the current `crontab` file by entering this command:

```
crontab -l > snapshot
```

A copy of the current `crontab` file is put into the file *snapshot*.

2 Edit the file *snapshot* to reflect the new changes.

3 Reinstall the file into the user's crontab file by entering this command:

```
crontab snapshot
```

In this command line, *snapshot* is the name of the file you created in step 1. This replaces the file `/usr/spool/cron/crontabs/user` with the contents of *snapshot* and alerts `cron` to reread the `crontab` file. The value of *user* is determined by your login name or the login name to which you have changed using the `su` command—whichever is most recent.

Removing `cron` commands

You can completely remove a `crontab` file by using this command:

```
crontab -r
```

This removes the file `/usr/spool/cron/crontabs/user`. The value of *user* is determined by your login name or the login name to which you have changed using the `su` command—whichever is most recent.

Using floppy disks that contain data in a non-Macintosh format

A/UX can accommodate three categories of floppy disks: new, unformatted disks; UNIX data disks; and Macintosh-formatted disks.

If you insert into your server a floppy disk that has never been formatted, or a disk that is formatted in another type of format, you see the usual dialog box that states that the disk is unreadable and provides the option to initialize it. If you initialize such a disk, you will lose all the data on the disk, and the disk will be formatted as a Macintosh disk.



However, if you insert a floppy disk that contains data in a non-Macintosh format, such as a `cpio` archive or a UNIX file system, no dialog box appears and no icon is displayed on the desktop. You can work with this disk only by using CommandShell. You can still use the disk for backing up files with UNIX backup utilities, such as `tar` or `cpio`. (UNIX backup utilities are described in the manual pages under the name of the utility.)

If you insert a Macintosh disk, a dialog box asks you to specify whether you want to use it for the Macintosh file system or for UNIX data. A third choice is to eject the disk (and therefore to cancel the procedure).



- If you click UNIX Data, the disk behaves like the UNIX disk described earlier (that is, no icon appears).
- If you click the Macintosh button, the disk behaves like a Macintosh file-system disk, and the icon appears on the desktop. You must use the Macintosh disk with the Finder.

6

Creating and Managing a TCP/IP Network

This chapter provides instructions for creating a simple two-computer network, to illustrate the procedures for creating and maintaining a larger TCP/IP network. Topics include

- how to gather network information
 - how to modify network configuration files
 - how to configure subnets; routing and forwarding
 - how to determine network status
-

Overview of the TCP/IP networking software

The TCP/IP networking software uses the Transmission Control Protocol and the Internet Protocol (TCP/IP) for communications between computers connected by Ethernet cable. TCP/IP actually represents a suite of protocols, of which two main components are TCP and IP. The A/UX implementation of TCP/IP is derived from the 4.3 BSD release of the UNIX operating system.

TCP/IP networking commands allow users to log in to remote computers, transfer files between computers, and execute commands on remote computers. In addition, TCP/IP is the foundation of UNIX networking upon which additional networking services, such as Network Information Service (NIS) and Network File System (NFS), are built.

Although every effort has been made to make TCP/IP networking secure, adding your computer to a TCP/IP network allows additional users access to your computer, and thereby increases security risks.

If you want to start building a TCP/IP network, first try creating a simple two-computer network as described in “Creating a Simple Network” later in this chapter.

If you want to add a computer running A/UX to an existing TCP/IP network, see Chapter 4, “Setting Up TCP/IP Networking,” in *Setting Up and Managing Your Server*.

Using other network resources

You may also want to consider other resources the network offers and whether you want your computer to use those resources. The following services are distributed on the system software installer CD-ROM and are documented in the *A/UX Network System Administration* manual:

- Network Information Service, which helps you manage administrative files
- Network File System, which lets you share files over a network
- CSL/IP, which lets you connect the computer to the network by using a serial connection
- UUCP, which allows you to communicate with UUCP networks
- network mail, which lets you send mail to users that are not on your system

Configuration parameters

Before enabling TCP/IP networking, you need to determine the values the computers will use for several important networking configuration parameters. These configuration parameters are

- host name
- network number and Internet Protocol (IP) address
- broadcast address
- netmask

The configuration parameters are set when you run the `newconfig` command and respond to its prompts. The `newconfig` command stores your answers in the network files described later in this chapter. The rest of this section defines each configuration parameter and describes the restrictions that apply to it.

Choosing your host name

Every computer on a TCP/IP network must have a host name. By default, the host name is `localhost`. The host name you choose must be no longer than 31 characters; cannot include metacharacters, such as the exclamation point (!), the backslash (\), the question mark (?), or the asterisk (*); should not contain any uppercase letters; and must be unique among all host names used by computers on the network. The host name is used by everyone on the network to refer to the system and to perform tasks such as logging in to the computer and using its files.

The host names of the computers that form the sample network used in this chapter are `hostname1` and `hostname2`.

Establishing your network number and IP address

The TCP/IP networking software allows the connection of a local area network (LAN) to other TCP/IP-network-based LANs to create a larger network called a *wide area network* (WAN). Thousands of interconnected TCP/IP-network-based LANs and WANs throughout the world make up a super network known as the *Internet*. To give each computer on the Internet a unique ID, each LAN is identified by a unique network number. The network number is combined with a host number to generate a unique 4-byte address, called an *IP address*,

for each computer on the LAN. The assignment of network numbers is handled by the Network Information Center (NIC), which acts as a clearinghouse for information about the Internet, including the development of new networking software.

If you do not plan to connect your network to the Internet, you can use the information in this section to create a valid network number, and from that number assign an IP address to each computer on your network. In effect, you create an isolated “mini-Internet” of your own. If you plan to connect your network to the actual Internet, you must obtain an official network number from NIC. See Appendix B, “Network Contacts,” for details.

If you plan to connect the network to the Internet and you need to use your network before you receive your official network number, you can use temporary numbers until you receive your network number. While you are using temporary numbers, do not connect your network to the Internet. The temporary numbers should follow the conventions described in this section.

Table 6-1 shows samples of the three classes of network numbers: A, B, and C. In class A networks, the network number is 3 bytes long, leaving 1 byte for the host part when assigning the IP address. This means that class A network numbers are given to organizations that must support a large number of networks—and therefore a large number of users. In class C networks, the network number is 1 byte long, leaving 3 bytes for the host part. While class C networks can therefore have more hosts on each network, they have a restricted number of networks and are therefore assigned to organizations that support fewer users.

Table 6-1 Sample IP addresses

Network class	Network number in decimal	Host number in decimal	IP address
A	10.45.106	32	10.45.106.32
B	135.2	57.10	135.2.57.10
C	192	33.20.5	192.33.20.5

Table 6-2 summarizes the parts of an IP address. The TCP/IP networking software lets you use subnets to increase the number of IP addresses that a network number can support; see “Subnets,” later in this chapter, for details.

Table 6-2 The parts of an IP address

Network class	Range of numbers in first byte (in decimal)	Number of bytes in network part	Maximum number of networks	Number of bytes in host part
A	1–126	1	126	3
B	128–191	2	16,384	2
C	192–223	3	2,097,152	1

Note: 0 and numbers over 223 are reserved for special purposes and should not be assigned to specific hosts on the network. The number 127 is reserved for the loopback driver.

If you are creating your own network number, select a class that allows you to plan for growth in terms of both the number of networks and the number of computers on each network.

Once you have decided on a network class that suits your individual needs, choose a number from the appropriate range as your first byte. If you are using a class B or class C network number, choose second and third bytes (respectively) to complete your network number. IP address numbers are most commonly referenced in “dot” notation; in this notation, each byte is separated by a period (.).

As an example, this chapter uses the class C network number 192 . 33 . 20. The network number serves no other purpose than to uniquely identify the network.

To form an IP address, append a period to the network number, and then append a unique decimal number, which is the host number. For example, if the network number is 192 . 33 . 20, the first IP address that you assign might be 192 . 33 . 20 . 1. (If there is more than one physical Ethernet cable at your site, see “Routing and Forwarding” later in this chapter.)

Once you have a network number, assign an IP address to each computer on your network. To avoid confusing the IP address with a broadcast address, make sure that the bytes in the host number do not contain all binary 0's (decimal 0) or all binary 1's (decimal 255). For example, do not assign 192.33.20.0 or 192.33.20.255 as an IP address. (Broadcast addresses are discussed in the next section.)

Here are the IP addresses of the two computers on the sample network:

```
192.33.20.1    hostname1
192.33.20.2    hostname2
```

Technically, the IP address is assigned to the Ethernet card. If a computer has more than one Ethernet card, each card has its own IP address. Most computers, however, have just one Ethernet card, so the IP address of the card is commonly referred to as the computer's IP address.

Determining the broadcast address for your network

A broadcast address is an address that is recognized by all computers on a network. The broadcast address is an IP address with a host part of all binary 1's (decimal 255). This method is used by A/UX and by the 4.3 BSD version of the UNIX operating system to broadcast messages to all computers on the network. Previous versions of the BSD operating system used a host part of all binary 0's (decimal 0). The `newconfig` command uses the IP address and netmask that you enter to calculate the broadcast address. Table 6-3 shows a sample broadcast address for each network class.

Table 6-3 Sample broadcast addresses

Network class	Network number in decimal	Broadcast address in decimal
A	10	10.255.255.255
B	135.2	135.2.255.255
C	192.33.20	192.33.20.255

The use of subnets, whereby the netmask is used to divide a single network into smaller parts, may alter the way in which the broadcast address is formed. (Netmasks are described in the next section.) See “Subnets” and “Subnets and Broadcast Addresses,” later in this chapter, for more details.

Determining the netmask for your network

The netmask is a 32-bit value that tells the networking software how to extract the network number from an IP address. The network number is determined by the bits in the netmask that are set to 1, and the host part is determined by the bits in the netmask that are set to 0. Table 6-4 lists the default netmasks for class A, B, and C networks.

Table 6-4 The default netmasks

Class	Number of bytes		Netmask in binary	Netmask in hexadecimal
	Network part	Host part		
A	1	3	11111111000000000000000000000000	ff000000
B	2	2	11111111111111111000000000000000	ffff0000
C	3	1	11111111111111111111111110000000	ffffff00

To implement subnets on a network, you set additional bits in the netmask. See “Subnets,” later in this chapter, for information on forming a netmask that implements subnets.

Creating a simple network

The simplest way in which you can use the TCP/IP networking software is to create a network of two computers. This section gives you step-by-step instructions for creating the simple network shown in Figure 6-1. The steps used to create a simple network are the foundation for building a more complicated network.

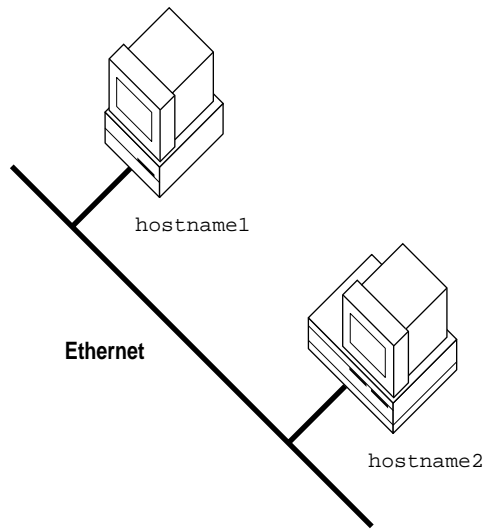


Figure 6-1 A two-computer network

Adding the first computer to a two-computer network

Once you have determined the values of the configuration parameters for your network, you can run `newconfig` to make a kernel that supports the TCP/IP networking software. To run `newconfig`, you must have `root` privileges.

The following procedure provides the steps for making a kernel for a computer that has one Ethernet card. If more than one Ethernet card is installed in the computer, `newconfig` prompts for the information for each card. When `newconfig` asks a question that is specific to one Ethernet card, it does so by prefacing the question with a card identifier, such as `ao0`, to refer to the built-in Ethernet card, or `as1` to refer to a second Ethernet card installed in the first empty NuBus slot.

To add the first computer:

- 1 Log in as `root` and open a CommandShell window.**
- 2 Make a new kernel by entering this command:**

```
newconfig ao
```

The following messages appear:

```
newconfig: Making backup copies of /etc/inittab
/etc/passwd /etc/group.
newconfig: Preparing to build a new kernel.
newconfig: Building a new kernel (this may take several
minutes)
```

The `newconfig` program is now running.

When you make a kernel that supports the TCP/IP networking software for the first time, the `newconfig` program prompts you for the network configuration parameters discussed earlier in this chapter. If you are unsure of the correct responses to these prompts, do not continue. Providing an incorrect response may lead to a problem that is difficult to troubleshoot. At any time, you can interrupt the `newconfig` program (usually by pressing `CONTROL-C`). When interrupted, `newconfig` restores the `/etc/group`, `/etc/inittab`, and `/etc/passwd` files to their original states.

As `newconfig` is running, it displays these messages:

```
newconfig: Restoring kernel parameters.
newconfig: Checking for any installation information.
```

The `newconfig` command then notices your built-in Ethernet interface and displays this message:

```
1 Ethernet card(s) installed
```

Then you see the first prompt:

```
ao0: Would you like to configure this interface for IP?
[y/n, default = y] >>>
```

This message is asking you to configure the Ethernet interface named `ao0`, which is the interface name of the built-in Ethernet interface for your server.

3 Enter `y` to configure the interface for TCP/IP.

If the network information hasn't been entered previously, the system prompts you to enter your IP address and netmask. (If this information has been entered previously, the prompts do not appear.)

If you enter `y`, the following prompt appears:

```
Please enter a host name (it must be unique):
```

4 Enter the computer's host name.

The host name must meet the conditions described in the earlier section “Choosing Your Host Name.” You are not prevented, however, from entering an invalid host name, although you can change your host name later.

The next prompt appears:

```
ao0: Please enter a NIS domain name [none]:
```

5 Enter none for the NIS domain name.

NIS is an optional advanced networking subpackage and is not configured at this time.

The next prompt appears:

```
ao0: Please enter an Internet address:
```

6 Enter the IP address.

When you have entered a valid IP address, the following prompt appears:

```
ao0: Please enter a netmask [none]:
```

Note that you are given a chance to confirm and change your answer later.

7 Enter the netmask, or press RETURN to use the default netmask.

The value of the netmask is discussed earlier in the section “Determining the Netmask for Your Network.”

When you enter a valid netmask, `newconfig` uses it and the IP address you entered to calculate your broadcast address.

If the netmask that you enter is incompatible with the IP address that you have already specified, you will see a message indicating the error and prompting you to enter the netmask again.

After you enter a compatible netmask, you are asked to confirm the IP address and netmask with a display that looks like this:

```
ao0:          IP address 192.33.20.1
              netmask 0xffffffff
              broadcast address 192.33.20.255
OK?          [y/n, default=y]>>>
```

8 Check the numbers to make sure they're correct, and then enter `n` or `y`.

If the IP address or the netmask is incorrect, enter `n`. When you enter `n`, you are prompted to enter the Internet address and the netmask again, and another prompt will appear asking you if the numbers are OK.

If the IP address and the netmask are correct, enter `y`.

When `newconfig` is done, it displays the current settings for your host name, IP address, IP broadcast address, and netmask.

If the host name was entered incorrectly, see “Modifying the `/etc/HOSTNAME` File” later in this chapter.

9 Return to the Finder and choose Restart from the Special menu.

The changes won't be effective until you restart the system.

Note: If you have a need for a broadcast address other than the address calculated by `newconfig`, you can change the broadcast address in the file `/etc/NETADDRS`.

Adding the second computer to the two-computer network

Log in to A/UX on the second computer and follow the steps listed in the preceding section. When `newconfig` prompts for the host name, enter `hostname2`, if you are following the example given earlier in this chapter. For the IP address, enter `192.33.20.2`. For all other prompts, enter the same value that you entered when you were running `newconfig` on `hostname1`.

When you have completed the procedure for adding the second computer to the network, users who have accounts on both computers can run `rlogin` on either computer to log in to the other computer. However, they cannot run `rcp` to copy remote files or `remsh` to execute remote commands. This is because the procedure described earlier imposes a reasonably high level of security. See “TCP/IP Network Security Issues,” later in this chapter, for ways to enable the use of `rcp` and `remsh`. That section also describes ways to increase network security.

When you have restarted the second computer, follow the instructions in the next section, “Testing the Networking Software,” to verify that the network is functioning properly.

Testing the networking software

You can test the TCP/IP networking software in several ways. You should begin by running the tests that involve the least number of variables. That is, you should run tests that exercise just one computer's interface to the network. If those tests succeed, proceed to tests that involve communications between two computers on the same network. If these tests succeed, run tests that involve communications with a computer on another network, if you have created more than one network.

Testing the loopback interface

The simplest test of the networking software is to see whether the computer can use the loopback interface. The loopback interface exercises software interfaces in the kernel, but does not exercise the Ethernet hardware.

To test the loopback interface:

- **Log in to A/UX and enter this command:**

```
rlogin loop
```

If you have given the root account a password, you are prompted to enter it. Otherwise, this prompt is displayed:

```
TERM = (vt100)
```

In response to the prompt, press RETURN. The appearance of the prompt indicates that the test was successful. To stop the `rlogin` command, press CONTROL-D.

If the `login` prompt doesn't appear after several seconds, stop `rlogin` by sending it an interrupt character (by default, CONTROL-C). Rebuild your kernel as described earlier in this chapter. Note that when you rebuild the kernel, `newconfig` does not prompt you to enter the network configuration parameters.

Using the `ping` command

You can use the `ping` command to verify the computer's connection to the network. The computer's IP address is used as an argument to `ping`.

To use the `ping` command:

- 1 If you have followed this two-computer example, and are logged in to `hostname2`, enter this command:

```
ping 192.33.20.1
```

If the network and both hosts are working properly, `ping` displays information similar to this:

```
PING 192.33.20.1: 56 data bytes
64 bytes from 192.33.20.1: icmp_seq=0. time=16. ms
64 bytes from 192.33.20.1: icmp_seq=1. time=16. ms
64 bytes from 192.33.20.1: icmp_seq=2. time=16. ms
64 bytes from 192.33.20.1: icmp_seq=3. time=16. ms
64 bytes from 192.33.20.1: icmp_seq=2. time=16. ms
64 bytes from 192.33.20.1: icmp_seq=3. time=16. ms
```

- 2 Enter the interrupt character (usually by pressing **CONTROL-C**) to stop the output of `ping` once you see that packets are being returned.

When a successful `ping` command is interrupted, it displays summary statistics similar to these:

```
----192.33.20.1 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 16/16/16
```

If TCP/IP networking is not supported in the kernel, `ping` displays this message:
`net down`

If the computer with which you are trying to communicate is connected to another network through a gateway that is not yet working, `ping` displays this error message:

```
Network is unreachable
```

The same error message is displayed when `routed` is not running; you can correct this problem by restarting `routed`. If the computer is connected to your network, but is not working, `ping` displays a message similar to this:

```
PING hostname1: 56 data bytes
```

At this point, `ping` does not produce any further output. Entering the interrupt character causes an unsuccessful `ping` command to display summary statistics such as these:

```
----hostname1 PING Statistics----  
12 packets transmitted, 0 packets received, 100% packet  
loss
```

Logging in to another computer

If you have a login account on both computers, you can log in to one computer from the other. For example, if you are currently logged in to A/UX on `hostname1`, use this `telnet` command to connect to `hostname2`:

```
telnet 192.33.20.2
```

If you are prompted for your login name and password on the remote computer, the test was successful. Enter `CONTROL-D` at the remote-login prompt to end the test.

If you see the message `Trying...` followed by `Connection timed out`, the test failed. Verify that both computers are connected to the network, have kernels that are configured for TCP/IP networking, and have valid IP addresses. See “Troubleshooting Network Problems” in Chapter 9 for information about verifying the integrity of the Ethernet cable and transceivers.

Changing a network configuration parameter

This section discusses the configuration files that the TCP/IP networking software uses to store information about the network. To change a network configuration parameter, edit the file that contains the configuration parameter and restart the computer.

Some of the files are modified when you run `newconfig` to create a kernel that supports TCP/IP networking. These are files that `newconfig` modifies:

```
/etc/HOSTNAME  
/etc/NETADDRS
```

As a network system administrator, you should be familiar with the contents of these files so that you can troubleshoot network problems. In addition, you may want to customize these files to increase or reduce the level of security that your network enforces.

Modifying the `/etc/HOSTNAME` file

The `/etc/HOSTNAME` file contains the computer's host name and NIS domain name in a single entry having the following format:

host-name domain-name

Any change to this file does not take effect until you restart A/UX. You can temporarily change your host name or domain name by running the `hostname` or `domainname` command, respectively, but neither command updates this file.

Modifying the `/etc/NETADDRS` file

The `/etc/NETADDRS` file contains the IP address, broadcast address, and netmask for each Ethernet card in the computer. The format of each entry in the `/etc/NETADDRS` file is as follows:

interface-ID IP-address broadcast-address netmask

If you remove the `/etc/NETADDRS` file, you must either use a text editor to re-create it or run the `newconfig` program again.

Note: If you rerun the `newconfig` program to configure a new `/etc/NETADDRS` file, a new entry for your host name and IP address is added to the end of the `/etc/hosts` file, and the original entry remains in the file. When looking for the IP address of any computer (including the local computer), the TCP/IP networking software uses the first appropriate entry that it finds. If you have more than one entry in your `/etc/hosts` file with the same IP address, you should immediately edit `/etc/hosts` and delete any duplicate or incorrect entries for your computer.

Table 6-5 lists the possible values of *interface-ID*.

Table 6-5 Values of *interface-ID* in the `/etc/NETADDRS` file

Interface-ID	Interface name	Ethernet hardware
ao0	ao	Built-in Ethernet on Apple Workgroup Server 95 (also Macintosh Quadra 700 and 900 computers)
0 through 5	ae6	Apple EtherTalk NB Card
as0 through as5	as	Apple Ethernet NB Card

The A/UX values for *interface-ID* will be ao0 for the first Ethernet card, ao1 for the second Ethernet card, and so on. The values of *IP-address*, *broadcast-address*, and *netmask* are the IP address, the broadcast address, and the netmask, respectively, that are associated with the Ethernet card specified by *interface-ID*. If you decide to change one or more of these values, you can use any text editor to modify `/etc/NETADDRS`, and then restart A/UX. Alternatively, you can remove this file and restart A/UX. As A/UX is restarting, it prompts you for the new information.

Other related files

This section discusses other files that contain important TCP/IP configuration information. Some of these files are modified when you run `newconfig` to configure TCP/IP networking.

Modifying the `/etc/inittab` file

The `newconfig` command edits the `/etc/inittab` file and changes the *action* field of the network daemon entries to the values shown here in boldface:

```
nfs0:2:wait:/etc/portmap
net4:2:wait:/usr/etc/in.routed
...
net8:2:once:/usr/lib/sendmail -bd -q30m
```

Each entry whose action field contains a value other than `off` is enabled. You may not want to run all of the daemons that `newconfig` enables. In particular, you may want to disable the following daemons at first:

- The `portmap` daemon (`/etc/portmap`) is not necessary unless you intend to run software that uses remote procedure calls (RPCs), such as NFS and NIS. See the system file `/etc/rpc` for a list of RPC services.
- The `routed` daemon (`/usr/etc/in.routed`), which manages network routing tables, is not necessary on a simple one-cable network. See “Designing a Complex Network,” later in this chapter, for information about when to enable `routed`.
- The `sendmail` daemon (`/usr/lib/sendmail`) handles the transmission and delivery of UNIX mail over the network.

The `newconfig` command does not enable `rwhod`, whose entry in `/etc/inittab` is shown here:

```
net5:2:off:/usr/etc/in.rwhod
```

The `rwhod` daemon maintains the database used by the `rwho` and `ruptime` commands. The `rwhod` daemon broadcasts frequently and can reduce network response time in large or heterogeneous environments. If you have a small network, you may want to run `rwhod` on each computer on the network. To enable it, change the value of the action field to `once`.

Modifying the `/etc/hosts` file

When you finish running `newconfig` `ao` for the first time, the `/etc/hosts` file contains an entry for the IP address for the loopback interface and an entry for the computer’s IP address and host name. These entries are required for the TCP/IP networking software to work correctly.

You should modify this file to add the IP address and host name of each computer on the network. After you make this modification, users can use host names instead of IP addresses as arguments to commands such as `ping` and `telnet`.

Modifying the `/etc/ftpusers` file

The standard A/UX distribution contains an entry for `root` in the file `/etc/ftpusers`. This entry prevents remote `ftp` users from logging in as `root` on the local computer. If you want to allow users to log in as `root`, edit `/etc/ftpusers` and remove the entry for `root`. You may also want to remove the entry for `root` temporarily until you have completed the setup of your network. However, permanently removing the entry for `root` is not recommended.

Modifying the `/etc/networks` file

This file lists the networks available to the computer. Each network name is used to represent a network number by the `route` and `netstat` commands, in the same way that a host name represents an IP address. For each network, enter a single line with the following information:

```
network-name    network-number  aliases
```

The value of *network-name* can contain any printable character other than a blank, a tab, a newline, or a number sign (`#`). A number sign is interpreted as the beginning of a comment.

Here are two sample entries:

```
arpanet      10    arpa
ucb-ether    46    ucbether
```

The standard A/UX distribution includes a version of this file that is the official network database maintained by the Network Information Center (NIC). Use a text editor to add a line for your own network to `/etc/networks`. For example, if you are logged in to A/UX on `hostname3`, as described in “Routing and Forwarding,” later in this chapter, you would enter a line such as this:

```
development  192.33.20  devnet
```

Modifying the `/etc/printcap` file

This file contains configuration information used by the `lpr` print spooler. If you plan to connect a printer directly to a remote computer on the network, a user on the local computer can print on that printer by using the `-RemoteHost` option to `lpr` if the `/etc/printcap` file on the local computer is set up. Edit the file `/etc/printcap` and find these lines:

```
# Remote UNIX line printer
# Change 'RemoteHost' to actual name of remote UNIX host
remote|remote line printer:\
    :lp=:rp=lp:rm=RemoteHost:sd=/usr/spool/lpd/Remote:
```

Change `RemoteHost` (shown in boldface) to the host name of the computer to which the printer is directly connected.

Modifying the `/etc/protocols` file

This file contains information about protocols used by the network. Modify this file only if you are connecting the network to another network that uses protocols that are not already listed in this file.

Modifying the `/etc/servers` file

This file lists the commands that `inetd` can run. Running `inetd` saves space in memory because the commands that it runs do not have to run continually in the background. Instead, `inetd` runs them only when a request comes in. The `inetd` daemon is enabled by `newconfig` in the `/etc/inittab` file. The `/etc/servers` file contains these entries:

```
ftp      tcp      /usr/etc/in.ftpd
uucp     tcp      /etc/uucpd
telnet   tcp      /usr/etc/in.telnetd
shell    tcp      /etc/in.remshd
login    tcp      /etc/in.rlogind
exec     tcp      /usr/etc/in.rexecd
# tftp   udp      /usr/etc/in.tftpd
```

```

talk    udp    /usr/etc/in.talkd
finger  tcp    /usr/etc/in.fingerd
rpc     udp    /usr/etc/rpc.rstatd    100001  1-2
rpc     udp    /usr/etc/rpc.rwalld    100008  1
rpc     udp    /usr/etc/rpc.mountd    100005  1
rpc     udp    /usr/etc/rpc.rusersd   100002  1-2
rpc     udp    /usr/etc/rpc.sprayd    100012  1
comsat  udp    /etc/comsat

```

If, for security reasons, you want to prevent `inetd` from running a command, precede the command with a number sign (`#`), which causes the entry to be interpreted as a comment. Note that `/usr/etc/in.tftpd` is disabled by default.

Modifying the `/etc/services` file

This file contains information about known services that are called by various network commands. Modify this file if you add a new network service.

Modifying the `/etc/shells` file

This file lists the shells that users are allowed to use as login shells. In the standard A/UX distribution, this file contains `/bin/sh`, `/bin/csh`, and `/bin/ksh`. Several programs, such as `login` and `ftpd`, check this file before allowing certain processes to run. If a user wants to run a different login shell, add its absolute pathname to this file.

TCP/IP network security issues

When you add a computer to a network, you increase the risk that an unauthorized user may gain access to your computer. The procedures described earlier in “Creating a Simple Network” impose a reasonable level of security. This section describes changes you can make to adjust the security of your network.

Making your system more secure from network access

This section discusses some steps you can take to increase the level of security that is imposed when you follow the procedures in “Creating a Simple Network,” earlier in this chapter. The steps consist of disabling certain networking daemons.

You can disable `rlogind` and `remshd`, which, when used with a `$HOME/.rhosts` file (described later in this chapter), do not check passwords to authenticate users. You may also want to disable `tftpd` for the same reason. (The `tftpd` daemon is disabled in the standard A/UX distribution.) You should also disable `rexecd`, unless it is needed to support MacX users, and `fingerd`, which displays information about users from the `/etc/passwd` file.

To disable a daemon, edit the file `/etc/servers` so that the entry for that daemon begins with a number sign (`#`), which causes the entry to be interpreted as a comment. The following list shows the entries from the `/etc/servers` file with the pertinent lines “commented out” and shown in boldface:

```
ftp          tcp          /usr/etc/in.ftpd
uucp        tcp          /etc/uucpd
telnet      tcp          /usr/etc/in.telnetd
#shell      tcp          /etc/in.remshd
#login      tcp          /etc/in.rlogind
#exec       tcp          /usr/etc/in.rexecd
#tftp       udp          /usr/etc/in.tftpd
talk        udp          /usr/etc/in.talkd
#finger     tcp          /usr/etc/in.fingerd
rpc         udp          /usr/etc/rpc.rstatd      100001   1-2
rpc         udp          /usr/etc/rpc.rwalld      100008   1
rpc         udp          /usr/etc/rpc.mountd      100005   1
rpc         udp          /usr/etc/rpc.rusersd     100002   1-2
rpc         udp          /usr/etc/rpc.sprayd      100012   1
comsat      udp          /etc/comsat
```

Making your system easier to access from the network

An open network reduces network security. You can make the network more open by specifying all of the computers on the network in the `/etc/hosts.equiv` file on each computer. The `/etc/hosts.equiv` file establishes a host “equivalence.” This equivalence allows users with entries in the `/etc/passwd` files on two computers to use `rlogin` without having to supply a password. The `/etc/hosts.equiv` file also prevents `rcp` and `remsh` from failing with a `Permission denied` message. Here is the format of an entry in the `/etc/hosts.equiv` file:

remote-host

The value of *remote-host* is the host name of a remote computer. For example, the `/etc/hosts.equiv` file on the system `hostname1` might contain the following:

```
hostname2
```

In this case, all users who are not logged in as `root` on `hostname2` and who have a login account on `hostname1` can log in to `hostname1` without supplying a password. They can also run `rcp` to copy files to `hostname1` and run `remsh` to execute commands on `hostname1`.

If a user has a different login name on another computer, he or she can still use `rlogin` by using the `-l` option. Note that the option is an ell not a one, and that any user on any computer listed in the `/etc/hosts.equiv` file on `hostname1` can log in to any nonadministrative account on `hostname1`. See “The `$HOME/.rhosts` File,” later in this section, for a more secure way of making the computer open.

If you want to allow a user logged in as `root` to use `rlogin` to log in to another computer on the network without having to supply a password, create a `/.rhosts` file that specifically names hosts and `root` on each computer. Here is an example:

```
hostname1 root
```

Because the `/.rhosts` file severely reduces network security, its use is not recommended.

The `$HOME/.rhosts` file

The `$HOME/.rhosts` file is an alternative to the `/etc/hosts.equiv` file. These files, located in the home directory of each user account, provide a level of openness similar to the `/etc/hosts.equiv` file, but at the level of the individual user. Here is the format of an entry in the `$HOME/.rhosts` file:

```
remote-host      login-name
```

The value of *remote-host* is the host name of a remote computer; the value of *login-name* is the name of a user who is logged in to *remote-host*. Whenever the user specified by *login-name* runs `rlogin` from the remote computer specified by *remote-host* to log in to the local computer (that is, the computer on which the `$HOME/.rhosts` file resides), `rlogin` does not prompt for the password. Password checking is also disabled for `rcp` and `remsh`, which cannot work if password checking is required.

Designing a complex network

If you are configuring a large network, you need to consider the physical limitations of the network, such as electrical parameters, the number of computers to connect, the total length of the cable, and the physical location of the computers (that is, whether the geographical distribution allows direct Ethernet connections among all computers on the network). In a large organization that is spread among two or more buildings, a way of communicating between multiple Ethernet cables is needed. To do this, you can use one or more of the following methods:

- Use a unique network number for each cable, and use Internet forwarders to route between networks. Note that if these networks connect to the outside world, the internal routing details are propagated to other computers that have no use for this information, resulting in unnecessarily large Internet routing tables.
- Use a single network number and subdivide the network by assigning subnet numbers to each cable. If you use this method, the internal division is not visible to the outside world.

- Use the Berkeley Internet Name Domain (BIND) software to organize hosts in a hierarchy of domains, which eases the job of resolving host names and IP addresses. The A/UX implementation of BIND is integrated with NIS so that both services can be used on the same network. You can use BIND in conjunction with routers and subnets or to organize a simple network.

Routing and forwarding

Some organizations have more than one Ethernet cable and use a separate network number for each cable. In this situation, a computer is connected to two separate networks and acts as an Internet forwarder by routing packets between the networks. To support the work of the Internet forwarder, each computer on each network runs the `routed` daemon.

Note: Do not confuse the software described in this section with the AppleTalk Internet Router software mentioned in *Setting Up and Managing Your Server*. The AppleTalk Internet Router software routes EtherTalk packets to LocalTalk networks.

Say that you are setting up a Macintosh computer running A/UX as an Internet forwarder between two networks. One network is the Development network and the other network is the Support network. The names of the computers on each network and their IP addresses are shown in Figure 6-2.

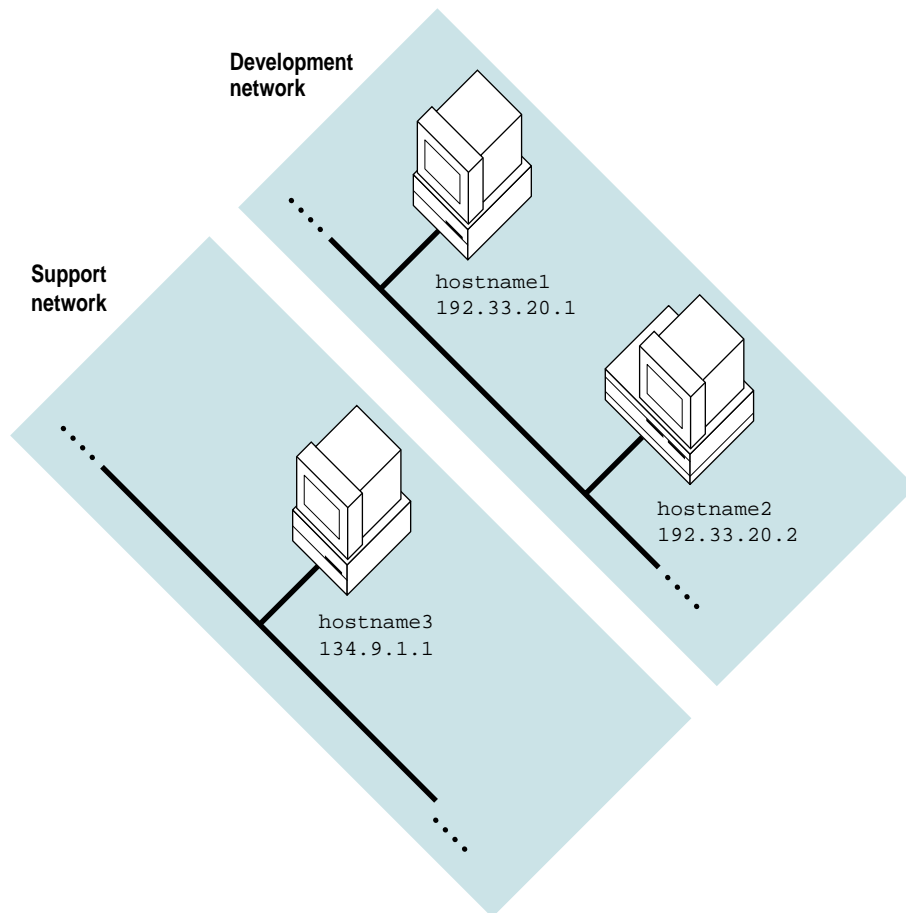


Figure 6-2 Two separate networks

To establish a forwarder between the two networks, choose one computer on one network to forward traffic to the other network. In this example, the forwarder system is `hostname2`, and it is equipped with two Ethernet cards. One card is connected to the Development network and the other card is connected to the Support network; each card has a unique IP address that is consistent with the network to which it is connected. Users on the Development network communicate with users on the Support network by using addresses that begin with `134.9`; these communications are forwarded to the Support network through `hostname2`. Users on the Support network

communicate with users on the Development network by using addresses that begin with 192.33.20; these communications are forwarded to the Development network by `hostname2`. The Development and Support networks are shown in Figure 6-3, connected by `hostname2`.

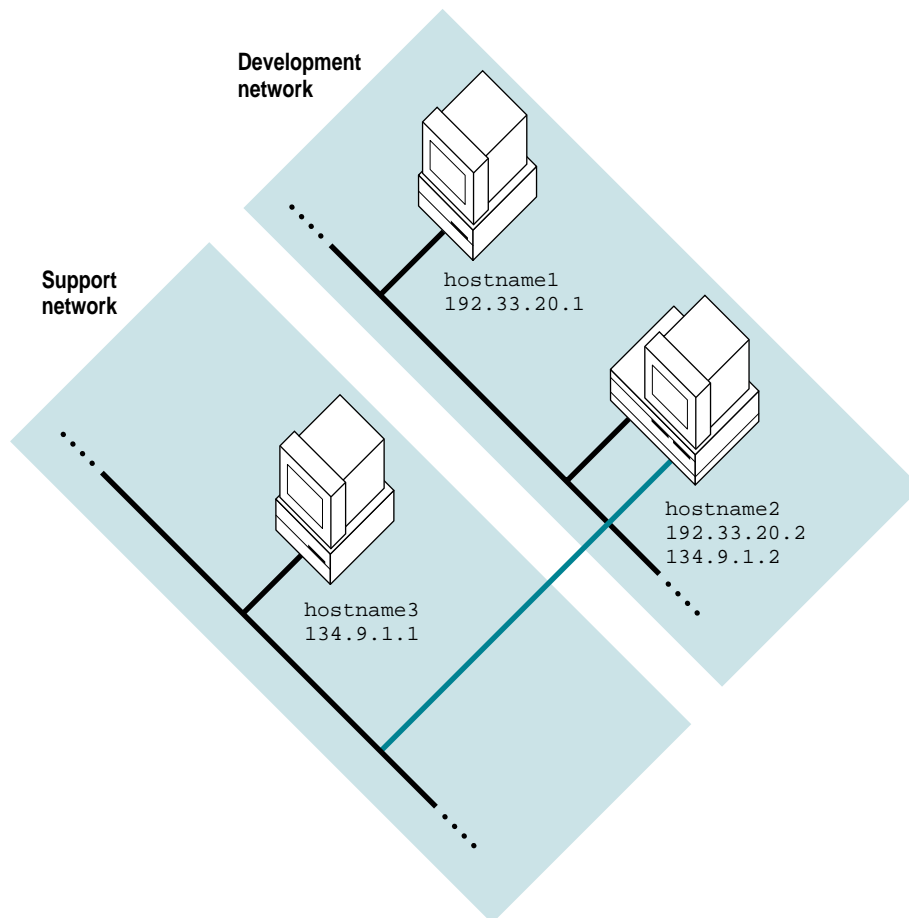


Figure 6-3 Two networks with `hostname2` acting as an Internet forwarder

To set up `hostname2` as an Internet forwarder, install a second Ethernet card, using the directions supplied by the card manufacturer, and connect the cable that came with the card to the second network. Then follow these steps, which use *interface-ID* to represent the value that `newconfig` displays when prompting for information about the second Ethernet card.

To configure `hostname2` as an Internet forwarder:

1 Start up `hostname2` and log in as `root`.

2 Make a new kernel by entering this command in CommandShell:

```
newconfig
```

The following message appears on the screen:

```
2 Ethernet card(s) installed
```

The computer displays this prompt for the interface name of the second Ethernet card:

```
interface-ID: Please enter an Internet address:
```

3 Enter the IP address to be used by the second Ethernet card.

If you setting up the sample forwarder system, enter this address:

```
134.9.1.2
```

Next, the computer displays this prompt:

```
interface-ID: Please enter a netmask [none]:
```

4 Enter the netmask to be used by the second Ethernet card.

Press RETURN to accept the default netmask, which is calculated for you based on the network number part of the IP address. The Support network in this example uses the default netmask. If your network is using a netmask other than the default, enter it, preceded by `0x`. If you don't know the netmask used by that network, log in to a computer on the second network (in this example, `hostname3`) and examine the `/etc/NETADDRS` file for an entry that contains the network number you specified as part of the IP address in step 3. The netmask is the fourth field.

The `newconfig` command uses the IP address and netmask to calculate the broadcast address and then stores all three values in `/etc/NETADDRS`. If the values that `newconfig` calculates are not appropriate for your environment, you can change them later.

5 When `newconfig` is done, restart A/UX.

6 When A/UX is restarted, log in and enable forwarding.

To enable forwarding, run `adb` on the kernel. The following lines show the commands you enter in boldface and the output of `adb` in plain type:

```
adb -w -k /unix /dev/kmem
a.out file = /unix (COFF format)
core file = /dev/kmem
ready
ipforwarding?D
ipforwarding: 0
ipforwarding?W 1
ipforwarding: 0x0 = 0x1
$w
$q
```

Note: To retain forwarding, you will have to repeat this step each time you make a new kernel on the forwarder system.

7 Enable `routed` on each computer on the networks.

You can manually install static (fixed) routes by using the `route` command; see `route(1M)` for details. Alternatively, you can change the action field for the `/etc/in.routed` entry in the `/etc/inittab` file to `wait`. Changing the action field to `wait` causes `in.routed` to start automatically, using dynamic routing tables, each time A/UX starts up. To start `in.routed` manually, enter this command:

```
/etc/in.routed
```

Even though each computer runs the routing software, only the forwarder system broadcasts routing updates, which are sent to each network to inform the computers of the presence of the other network, and routes packets between the two networks.

8 Modify the `/etc/hosts` file on all computers on the Support network.

Add the host name and IP address of all other systems on the local network, including the forwarder system.

9 Modify the /etc/networks file on all computers on the Support network.

You should modify the /etc/networks file on all computers on each network so that the file contains the name and network number of each network to which the local network has access. The /etc/networks file is described in “Modifying the etc/networks File” earlier in this chapter.

Table 6-6 shows the /etc/hosts and /etc/NETADDRS files, respectively, on the three computers that make up the example.

Table 6-6 The configuration files of computers on the sample network

The /etc/hosts file on the sample network computers

```
hostname1 #Development network
          192.33.20.1 hostname1
          192.33.20.2 hostname2
          134.9.1.2 hostname2

hostname2 #Development network
          192.33.20.2 hostname2
          192.33.20.1 hostname1

          #Support network
          134.9.1.1 hostname3
          134.9.1.2 hostname2

hostname3 #Support network
          134.9.1.1 hostname3

          #Development network
          134.9.1.2 hostname2
```

The /etc/NETADDRS file on the sample network computers

```
hostname1 as0 192.33.20.1 192.33.20.255
0xffffffff00

hostname2 ae0 192.33.20.2 192.33.20.255
0xffffffff00

          ae1 134.9.1.2 134.9.255.255
0xffff0000

hostname3 as0 134.9.1.1 134.9.255.255
0xffff0000
```

Subnets

If the network includes more than one physical Ethernet cable, you may still want to use just one network number. To do so, you can treat each physical cable as a subnet while, to the outside world, your network appears to be a single entity. The mechanism for creating subnets is the netmask.

As described in “Determining the Netmask for Your Network,” earlier in this chapter, the netmask is a 32-bit value that is used by the TCP/IP networking software to determine which part of an IP address is the network number and which part is the host number. When interpreted in binary form, the bits of the netmask whose values are set to 1 determine the network number, and the bits of the netmask whose values are set to 0 determine the host number. Each of the three Internet classes has a standard (or default) netmask.

Nonstandard netmasks are used to implement subnets. When you implement a subnet, the bit values in the netmask split the host number into two parts: a subnet number and a host number. On a class B network, a netmask of `0xffff0000` specifies no subnets, because it uses the class B standard netmask of 2 bytes for the network number and 2 bytes for the host number. To divide the network into subnets, some bits in the third and fourth bytes are set to 1 to form the subnet number. The remaining unset bits form the host number.

For example, a netmask of `0xffffff00` (with the additional set bits shown in boldface) for a class B network causes the third byte of an IP address, which is ordinarily the first byte of the host number, to be interpreted as a subnet number. Note that using the netmask to implement subnets does not change the computer’s IP address; the netmask only causes the networking software to interpret the address in a different way.

The procedure for setting up subnets is the same as the procedure described under “Routing and Forwarding,” earlier in this chapter, except that the netmask is different. In addition, on a class C network, the extra bits that you set to form the netmask reduce the number of bits available to form the broadcast address, so the broadcast address is also different in the fourth byte from the standard 255.

Table 6-7 shows the number of bits available for the subnet number and the host number for a sample netmask for each network class.

Table 6-7 Breakdown of a subnet netmask for each network class

Network class	Netmask	Number of bits	
		Subnet number	Host number
A	0xffff0000	8	16
B	0xfffffff0	8	8
C	0xffffffff0	4	4

Table 6-7 shows that, for a class C network number, a netmask of 0xffffffff0 allows four bits for the subnet number and four bits for the host number, which allows a total of 14 subnets with 14 hosts on each. (0 and all 1's have special meaning, so they are not used as network, subnet, and host numbers.) Table 6-8 shows the available IP addresses.

Table 6-8 The range of IP addresses for a class C subnetted network

Subnet 1	Subnet 2	...	Subnet 14
192.11.1.0x11	192.11.1.0x21		192.11.1.0xe1
192.11.1.0x12	192.11.1.0x22		192.11.1.0xe2
.	.		.
.	.		.
.	.		.
192.11.1.0x1e	192.11.1.0x2e		192.11.1.0xee

Note: For the purpose of illustration, Table 6-8 shows the fourth byte of the IP address in hexadecimal, preceded by 0x. The 0x notation is consistent with the A/UX implementation of the TCP/IP networking software, which allows you to enter values in hexadecimal when they are preceded by 0x. Software and hardware from other manufacturers may require the decimal equivalent of the hexadecimal number.

Figure 6-4 shows three computers, on two physical Ethernet cables, using the class C network number 192.33.20. The computer whose host name is `hostname2` contains two Ethernet cards. That computer receives all incoming packet traffic and uses the third byte of the IP address to determine the subnet to which traffic should be routed. All traffic generated by the computers on the network goes through `hostname2` to the outside world. All three computers use IP addresses beginning with 192.33.20 and a netmask of `0xfffffff0`, which defines the third byte as the subnet part of the network number.

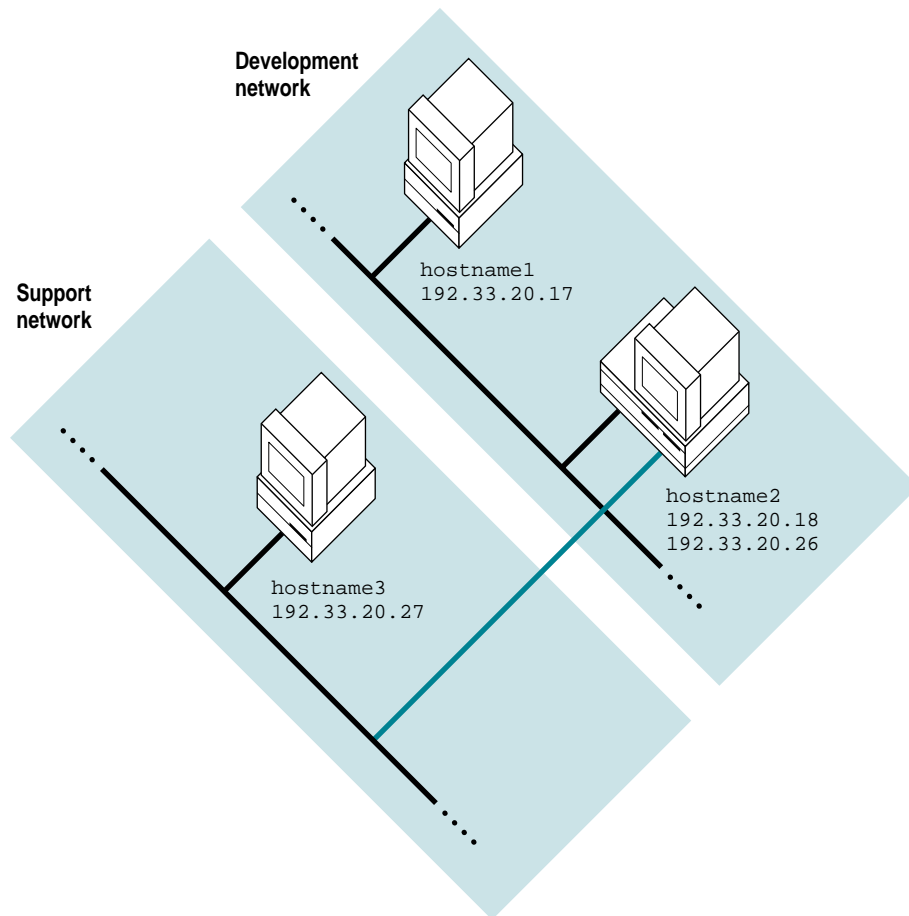


Figure 6-4 Two subnets with a netmask of `0xfffffff0`

Table 6-9 shows the `/etc/hosts` and `/etc/NETADDRS` files for each computer on the two subnets.

Table 6-9 Network configuration files on the sample subnets

The `/etc/hosts` file for each computer on the sample subnets

```
hostname1 #Development network
          192.33.20.17 hostname1
          192.33.20.18 hostname2
          #Support network
          192.33.20.27 hostname3
hostname2 #Development network
          192.33.20.18 hostname2
          192.33.20.17 hostname1
          #Support network
          192.33.20.27 hostname3
          192.33.20.26 hostname2
hostname3 #Support network
          192.33.20.27 hostname3
          #Development network
          192.33.20.17 hostname1
          192.33.20.18 hostname2
```

The `/etc/NETADDRS` file for each computer on the sample subnets

hostname1	as0	192.33.20.17	192.33.20.15	0xffffffff0
hostname2	ae0	192.33.20.18	192.33.20.15	0xffffffff0
	ae1	192.33.20.26	192.33.20.15	0xffffffff0
hostname3	ae0	192.33.20.27	192.33.20.15	0xffffffff0

Subnets and broadcast addresses

Table 6-10 shows the broadcast addresses that correspond to a sample netmask for each network class.

Table 6-10 Netmasks and broadcast addresses for subnets

Network class	Netmask	Broadcast address
A	0xffff0000	<i>n1.n2.255.255</i>
B	0xffffff00	<i>n1.n2.n3.255</i>
C	0xfffffff0	<i>n1.n2.n3.15</i>

Some daemons, such as `rwhod`, broadcast frequently. When many computers receive broadcast packets that do not correspond to their broadcast address, network efficiency is reduced because of the associated high collision and network traffic rates. You can use the same broadcast address on each subnet. If you use such an address, any broadcast would reach every computer on either subnet. If the subnets are large or if there are many subnets, broadcasting to all the computers may noticeably impair network performance. If this situation occurs, you should limit broadcasts to the relevant subnet by assigning a different broadcast address to each subnet.

Administering a TCP/IP network

This section describes commands used to determine network status and describes some responses to common error messages of computers on a TCP/IP network.

Determining network status

You can use several commands to determine the status of the network. This section discusses the `ruptime`, `rwho`, `ping`, `telnet`, and `netstat` commands.

Finding out what computers are running and who is logged in

Use the `uptime` and `rwho` commands to determine what computers are up and who is logged in to them. Here is an example of the output of the `uptime` command:

```
jasmine  down  2+20:58
bun      up    6+23:55,    1 user,    load 0.00, 0.03, 0.12
fin      up    17:13,    1 user,    load 1.00, 1.01, 1.01
```

The computers are listed in alphabetical order by host name. The second column shows whether the computer is running or not. The third column indicates the length of time the computer has been up or down. In this example, the computer `jasmine` has been down for two days, 20 hours, and 58 minutes. The computer `fin` has been up for 17 hours and 13 minutes. The fourth column indicates the number of users who are logged in to each computer that is up, and the remaining columns indicate each computer's load average over the last 1-minute, 6-minute, and 16-minute intervals.

Here is an example of the output of the `rwho` command:

```
lori    fin:ttyC1      Aug  9 10:12 :01
mike    bun:ttyC1      Aug  9 10:10 :28
```

The first column lists the login name of each user who is logged in, and the second column lists the host name and terminal number. The third column indicates the time the user logged in.

The value of the `uptime` and `rwho` commands is somewhat limited because they rely on the daemon `rwhod`, which sends out a broadcast packet once a minute to update its status list. On a large network, the overhead of these broadcasts may slow network response time, so you may want to disable this daemon for the entire network or on individual computers. Therefore, `uptime` and `rwhod` may inaccurately report that no computers are up, or may report only a subset of the computers that are up. If the local computer is not running `rwhod`, running `uptime` displays this message:

```
no hosts!?!
```


Determining if a specific host is up

The `ping` command is the simplest way of determining whether the local or a remote host is really up on the local network. This command uses the mandatory `ECHO_REQUEST` datagram of the Internet Control Message Protocol (ICMP) to elicit a response from a host or gateway. You can then use `ping` on other computers that are farther and farther away, connected through forwarder systems, and so on.

The `ping` command is often the first command to try when you cannot connect to a remote computer by using `rlogin`, `rcp`, `remsh`, `telnet`, or `ftp`. For example, enter this command:

```
ping hostname
```

The value of *hostname* is the host name of the computer on the local network to which you want to connect.

When `ping` cannot communicate with a remote computer because TCP/IP networking is not configured in the local computer, `ping` displays the message `net down`. If the computer with which you are trying to communicate is connected to the network through a gateway, `ping` displays this error message:

```
Network is unreachable
```

If `ping` displays a single line of information such as

```
PING hostname: 56 data bytes
```

and pauses, you must send an interrupt (usually by pressing CONTROL-C) to stop it. In this case, an unsuccessful `ping` communication yields the following possibilities:

- The route daemon, `routed`, may not be running on the local or the remote system. Check to see if `routed` is running by issuing this command:

```
ps -ef | grep routed
```

If the output of the command does not show a `routed` process, start `routed` by issuing this command:

```
in.routed
```

- The remote computer may be down.

- The remote computer may be not be running a kernel that includes the TCP/IP networking software. Verify that the kernel on the remote computer includes the TCP/IP networking software. On computers running A/UX, use the `module_dump` command and look for the name of an Ethernet device driver in the first column. (The name of the Ethernet device driver should be `bnet` combined with either `a0`, `as`, or `ae6`.)
- The physical connection between the computers may be broken. Log in to another computer and run `ping` to the remote computer. Verify that all cable connections and terminators are working.

When `ping` can communicate with a remote computer, it displays information such as this:

```
PING hostname: 56 data bytes
64 bytes from 128.8.1.1: icmp_seq=0. time=16. ms
64 bytes from 128.8.1.1: icmp_seq=1. time=16. ms
(interrupt)

----hostname PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 16/16/16
```

A successful `ping` communication yields the following conclusions:

- The remote computer is up.
- Both the local and remote computers are running the TCP/IP networking software.
- The physical network cable is transmitting packets correctly.

Determining that the IP address is correct

The `/etc/hosts` file, where each host name is associated with its IP address, is a common source of networking problems. Because the `telnet` command accepts both host names and IP addresses as arguments, you can use it to help you verify that an entry in the `/etc/hosts` file is correct.

7

Printing From CommandShell

This chapter discusses printer configuration and print commands used within CommandShell. You print text created by commands, or text files referenced on the command line through CommandShell, by using a command called `lpr`. This command allows you to use your server as a UNIX print server.

This chapter discusses

- printer connections to your server—both direct and through a network
- the definition of a port and a device driver
- the print spooler system and the commands associated with it

If you use the Finder for your printing needs, or you print files only from within Macintosh applications, you do not need to set up the `lpr` print system—refer to *Basic Skills*. If you need instructions on connecting an Apple printer, refer to *Setting Up and Managing Your Server*. For instructions on attaching other printers to your server or the network, refer to the documentation that came with your printer.

To understand some sections in this chapter, you need to be familiar with the file `/etc/inittab`. This file is discussed in Chapter 3, “System Startup,” and in `inittab(4)`. See also “Changing Run Levels: `init`” in Chapter 3, because use of the `init` command may affect other users.

To configure your printer for use by `lpr`, read through this entire chapter. If the print system has already been set up, skip to “Printing Files,” later in this chapter.

Printer connections

You can connect a printer to your Apple Workgroup Server 95 in two different ways—either directly through a serial cable or indirectly through a network. If you print through the Finder, or from within Macintosh applications, and use a network to connect to your printer (the recommended configuration), your use and configuration of the printer will be less complex. If you need to print using CommandShell, you must set up the `lpr` print daemon as described in this chapter.

You can use Table 7-1 to help you decide how to connect your printer to your Apple Workgroup Server 95, depending on the ways you want to be able to print. In Table 7-1, *local* means printing from your server, *remote* means printing from another system on the network, *direct* means the printer is directly connected to the server by means of a serial connection, and *network* means the printer is connected to the server by means of a network. The network can be either LocalTalk or Ethernet. If you are using Ethernet with Apple-compatible printers, you must select EtherTalk in the Chooser.

Table 7-1 Printing with A/UX

	Apple-compatible printer		Non-Apple-compatible printer	
	Serial	Network	Serial	Network
Print from Finder to local system	yes	yes	no	no
Print from Finder to remote system	no	yes	no	no
Print from CommandShell to local system	yes	yes	yes	yes
Print from CommandShell to remote system	yes	yes	yes	yes

Note: Some Apple LaserWriter printers are designed for a network environment only, and are therefore not compatible with the Apple serial interface. This means that in this table, those LaserWriters are considered to be non-Apple-compatible printers when connected through a serial port.

Figure 7-1 shows an example of remote printing over a network.

Before you set up the `lpr` print subsystem, you should understand the concepts of peripheral device drivers and ports.

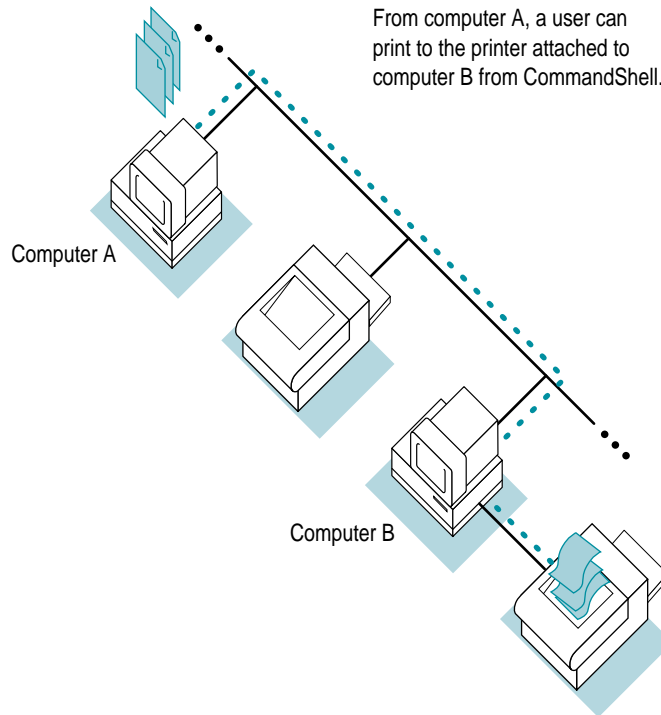


Figure 7-1 Remote printing from CommandShell

Peripheral device drivers

Every A/UX peripheral device requires a device driver. A device driver is the software interface between A/UX and a peripheral device. The peripheral devices that you connect to your machine fall into two categories: those that have device drivers built into A/UX by default, and those that require adding a device driver to the A/UX kernel.

The A/UX kernel has drivers built in for commonly used Macintosh peripheral devices, whereas device drivers for less commonly used or non-Apple devices must be added to the kernel before the device is used. In general, if adding the device involves adding an expansion card, such as the Ethernet NB Card (as opposed to using the built-in Ethernet port), you also need to add a device driver with the `newconfig` command.

If you need to add a device driver to your kernel, refer to “The `newconfig` Command” in Chapter 3, “System Startup.”

Note: You do not need to add a device driver to add terminals, printers, modems, built-in Ethernet port support, tape drives (DAT, nine-track, DCAS, QIC, and Apple Tape Backup 40SC), CD-ROM drives, hard disks, or Apple floppy disk drives to the A/UX system. Support for these devices is already built into the standard A/UX kernel.

Ports

A computer communicates with other equipment through a port, which is a physical connection point on the computer. You can attach peripheral devices to your computer by connecting them (with cables or connectors) to the ports. Refer to *Setting Up and Managing Your Server* for a description of these ports and a diagram that shows their locations.

The ports to which you connect peripheral devices such as printers, modems, and terminals are called *serial ports*. The Apple Workgroup Server 95 has two serial ports; they are located on the back of the computer and are marked with a telephone handset icon (☎) and a printer icon (🖨). Each port takes an 8-pin minicircular (DIN-8) connector. Make sure you distinguish the serial port from the Apple Desktop Bus (ADB) ports, which look similar, but are marked with the ADB icon (⌘) and take 4-pin connectors. The signals on the individual pins of the serial ports are shown in Figure 7-2.

The port identified by the phone icon has the A/UX device name `/dev/tty0`; it is also called `/dev/modem`. The port identified by the printer icon has the device name `/dev/tty1`; it is also called `/dev/printer`. Even though these ports have the name `/dev/modem` and `/dev/printer`, they are both serial ports. Therefore, you can attach a modem or a printer to either port.

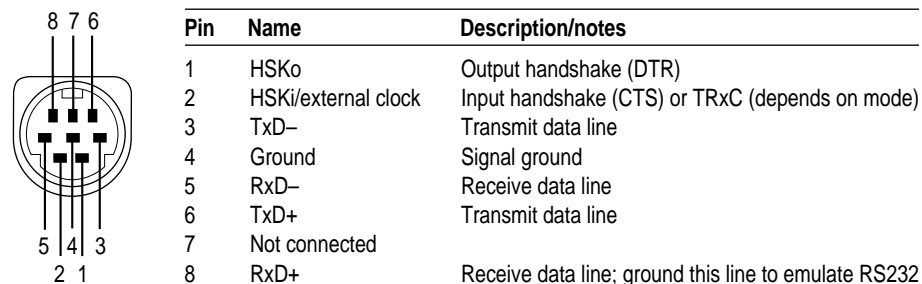


Figure 7-2 Signals on a serial port

Note: This chapter discusses using `/dev/tty1` (`/dev/printer`) as a printer port. This port is also used by LocalTalk to add your system to a LocalTalk network. If you plan to attach your Apple Workgroup Server 95 to a LocalTalk network, you cannot use `/dev/tty1` as a printer port. You can configure the system to use `/dev/tty0` as a printer port.

Managing the `lpr` print spooler

This section introduces the commands necessary to manage the 4.3 BSD `lpr` print-spooler system. The print-spooler system is a collection of programs and files used to manage a printer's operation.

Definitions

Some important terms used in this section are defined as follows:

- A print job submitted to the printer using the `lpr` command is called a *request*.
- The unique name by which the printer system identifies a specific printer is indicated in the command lines by the variable *printer*.
- Output of the printer is normally routed to a specific printer, called a *destination*. The system has a default destination defined in the file `/etc/printcap` unless the user explicitly requests a particular printer or printer class on the `lpr` command line.
- A piece of hardware such as a printer or modem that can be connected to the computer through a port is called a *device*.
- A file in the `/dev` directory that is associated with a particular device is called a *device file*. When the `lpr` system writes to a device file, such as `/dev/tty1`, output is sent to that port. The `lpr` system maintains information necessary to associate each printer with a particular device file.
- The print spooler's scheduler, called `lpd` (line-printer daemon), schedules print requests received from the `lpr` command. The `lpd` scheduler runs continuously in the background and is usually started by the `init` process when A/UX enters multi-user mode. The entry for `lpr` in the file `/etc/inittab` should be set to `once`, when `lpr` is being used, as follows:

```
lpr:2:once:/usr/lib/lpd > /dev/syscon 2>&l
```

- Each printer is controlled by an interface program, which may be shared by more than one printer. Interface programs perform such tasks as setting port speed, selecting printer options, printing header pages, and perhaps filtering certain characters that a particular printer may not know how to handle. The `lpr` system maintains the information necessary to select the proper interface for a given printer.

The `lpr` line-printer system is composed of two subsystems: the administrative system and the spooler. The administrative system is a series of commands you use to configure and maintain the entire line-printer system. The print spooler allows more than one person to use the same printer simultaneously or the same user to enter more than one print job in a queue. The spooler intercepts print requests, schedules them for printing on a specified printer or group of printers, and then selects the appropriate interface for the printer. When configuring the spooler, the server administrator assigns each printer a unique name, by which the spooler will identify the printer. The spooler maintains a list of user print requests organized by printer name. Figure 7-3 illustrates the printing queue.

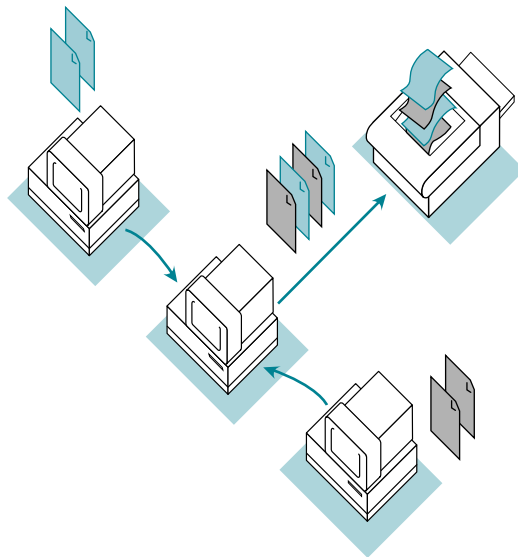


Figure 7-3 The printing queue

This printer system is controlled and managed through a set of commands that

- queue or cancel printer requests
- query the status of printer requests or of the printer system itself
- prevent or allow queuing printer requests to specific printers
- start or stop the printer system
- change the printer configuration

Setting up the print spooler

To be able to spool, or place in a queue, requests to a printer, the printer must be configured within the print-spooler system. By default, A/UX 3.0.1 comes with AppleTalk and ImageWriter printer spools established to print to the printer selected in the Chooser. Therefore, once you have selected a printer in the Chooser, you can use the `lpr` command to print to that printer without further configuration. Selecting a LaserWriter in the Chooser is the easiest way to print to a LaserWriter with `lpr`.

You can configure the print-spooler system to print to non-Apple-compatible printers, or to AppleTalk printers other than the one currently selected in the Chooser, or to print to a printer that is directly connected to a remote printer that also supports `lpd`. You must edit the `/etc/printcap` file as well as provide printer filters for any printers other than the LaserWriter or ImageWriter. This includes both Apple and non-Apple printers. The sections “The `printcap` Database” and “Writing Printer Output Filters,” later in this chapter, tell more about the type of information necessary.

The `printcap` database

The text file `/etc/printcap` contains entries that describe printers used by the print spooler. Each entry is used to describe one printer. You can edit this file with your usual text editor. Because this is an important system file, to save changes to this file you must be logged in as `root`. As with all important files, you should make a copy of this file before you edit it, in case you find that you want to undo your changes.

If you have a printer that does not have an entry in `/etc/printcap`, you can either build a printer definition manually or ask the printer vendor to supply you with a UNIX `/etc/printcap` entry for that printer.

This section describes some of the printer capabilities that can be defined in the `printcap` file. Refer to `printcap(4)` in *A/UX Programmer's Reference* for a list of all the possible options. Refer to `termcap(4)` in *A/UX Programmer's Reference* for further explanation of the format of the file.

The `printcap` file includes entries for AppleTalk and ImageWriter II printers connected locally by means of a serial port. You should need to modify this file only if you are using another type of printer.

Printer naming Each entry in the database begins with a list of names, separated by vertical bars that uniquely identify the printer described by the entry. Any of these names can be used when invoking `lpr` from a CommandShell window. For example, this is the entry for the ImageWriter II:

```
iw|iw2|ImageWriter II:\
:lp=/dev/printer:br#9600:os#0014001:cs#0004060:fd:\
tr=\f:sd=/usr/spool/lpd/ImageWriter:
```

In this entry, the first line contains different names for the printer, and the second and third lines consist of definitions for the printer. Therefore, when designating a printer on the `lpr` command line, either of the following two lines would designate the ImageWriter II entry:

```
lpr -Piw
lpr -Piw2
```

The default printer in the standard configuration is the generic AppleTalk printer, which designates using the printer selected in the Chooser. This entry begins with the following line:

```
lp|at|AppleTalk|postscript|Postscript: \
```

The special name `lp` in the name field is used to designate a printer entry as the system default destination. The default destination is where `lpr` will send any print jobs that do not specify a printer with the `-P` option of `lpr`.

To change the default printer destination, remove `lp` from the AppleTalk entry and place it as one of the names in the entry of the new default destination. The environment variable `PRINTER` can also be used to override the default for a particular user.

The name `lp` is also used as a definition of the default port used by `lpr`. This use of `lp` will be in the definition fields and will always have a colon (`:`) in front of it. Do not confuse `lp` in the name field with the `:lp=/dev/printer` definition.

Printers on serial lines The definitions of the characteristics of the printer follow the printer name (or names) in each entry of the database. These definitions are two-character variables set in fields separated by colons. Each printer connected directly by means of a serial communication port must have its characteristics, such as baud rate, defined in the `printcap` entry. Here is an example of some of these definitions for the ImageWriter:

```
:br#9600:os#0014001:cs#0004060:fd:tr=\f:
```

The `br` entry sets the baud rate for the port—in this case, 9600 baud.

The `os` entry defines the output flag bits that are set for this ImageWriter (that is, the system treatment of output to the printer). The value shown in the example is `0014001`. According to the `termio(7)` manual page, `0014000` for output flag bits tells the system to expand tabs to spaces, and `0000001` tells the system to postprocess output. Therefore, both of these characteristics are in effect when using that ImageWriter.

The `cs` entry sets other control characteristics of the port; in this case, `0004060` tells the system that the connection is a local line (not a dial-up line), and the data should be 8-bit.

The `fd` entry doesn't take a value, but its appearance causes the use of hardware-supported flow control or handshaking as described in `termio(7)`.

The `tr=\f` entry indicates that a form feed (`\f`) should be sent when the queue empties so that the paper can be torn off without taking the printer off line and pressing the form-feed button.

You may also notice backslashes (`\`) at the end of each line of an entry, except at the last line. The backslash indicates that the entry continues on the next line.

The `lp` entry defines the default port used by `lpr`. In the previous example, this is `/dev/printer`. If another port were to be used, it would be specified as follows:

```
:lp=/dev/port:
```

In this entry, *port* is the name of the port that the printer is connected to (as described in “Ports” earlier in this chapter). For example, to connect a printer to the modem port you would specify this entry:

```
:lp=/dev/tty0:
```

Spooling directory A spooling directory is a directory that holds print requests until the printer is available. Each printer that does not have a separate spooling directory has all its requests sent to the default spooling directory `/usr/spool/lpd`. A print request does not keep track of the specific printer it was sent to after it reaches the spooling directory. Therefore, if several printers have the same spooling directory, a request sent to any one of the printers will be printed on whichever printer is available.

The variable `sd` specifies an alternative spooling directory for a printer. In the following example, `/usr/spool/lpd/ImageWriter` is the specified spooling directory.

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

Output filters Filters handle device dependencies and perform accounting functions. The output filter definition, `of`, is used when all text data must be passed through a filter before it is sent to the printer. In the following example, all text sent to the AppleTalk printer will first be processed by the filter `/usr/spool/lpd/AppleTalk/ofilter`.

```
:of=/usr/spool/lpd/AppleTalk/ofilter:
```

For more discussion of filters, see “Writing Printer Output Filters” later in this chapter.

Remote printers It is also possible to send your print requests to printers that are connected to other computers. To accomplish this, you would set the `rm` (remote host) and `rp` (remote printer) variables.

For example, to print files on the printer named `inkwell`, which is directly connected to the remote host `photon`, insert the following entries into the remote entry in `/etc/printcap`:

```
remote| remote line printer:\
        lp=:rp=inkwell:rm=photon:sd=/usr/spool/lpd/Remote
```

Access control The `rg` variable controls local access to printer queues. For example,

```
:rg:lprgroup
```

requires that a user must be in the group `lprgroup` (which must be defined in `/etc/group`) in order to submit jobs to a particular printer. If the `rg` variable is not defined for a printer, all users have access to that printer. Once the requests are in a printer queue, they can be printed locally or forwarded to another system, depending on the configuration. Remote access is controlled by listing the hosts either in the file `/etc/hosts.equiv` or in `/etc/hosts.lpd`. (The format of these files is one remote host name per line.) The `remsh` and `rlogin` commands use `/etc/hosts.equiv` to determine which hosts are specifically designated to allow users to log in without passwords. The file `/etc/hosts.lpd` controls which hosts have line-printer access.

lpr commands

The commands used to administer the line-printer system can be divided into two categories: those that any user can use, and those that only a user logged in as `root` (or a user who is a member of the group `lpr`) can use. This section gives a brief description of what each command does. For examples of how these commands are used, see “Printing Files” later in this chapter.

Commands for general use

<code>lpr</code>	Usually takes a filename as an argument; submits that file as a print request to the <code>lpr</code> system. The request can print on the default system destination or on a specified printer or printer class, depending on the options used. See <code>lpr(1)</code> in <i>A/UX Command Reference</i> .
<code>lpq</code>	Shows the line-printer queue. This program has two forms of output: the short format (the default), which gives a single line of output per queued job; and the long format, which shows the list of files that make up a job, as well as their sizes. See <code>lpq(1)</code> in <i>A/UX Command Reference</i> .

`lprm` Removes one or several jobs from a printer's spool queue. The `lprm` command can cancel requests by printer name, request ID number (supplied by `lpq`), or user name. See `lprm(1)` in *A/UX Command Reference*.

Commands for line-printer administrators

This section discusses the major commands of the `lpc` program, which provides administrative control over line-printer activity. The `lpc` command can be used to

- disable or enable a printer
- disable or enable a printer's spooling queue
- rearrange the order of the jobs in the spooling queue
- find the status of printers

For additional information on the command format and other `lpc` options, refer to `lpc(1M)` in *A/UX System Administrator's Reference*. With each of the following commands, you can specify a particular printer, or you can affect all printers controlled by `lpd`.

```
lpc abort printer_name|all
```

The `abort` option stops printing immediately and disables printing for the specified printers. It does not remove any jobs from the queue. Use the `lprm` command to remove jobs.

```
lpc start printer_name|all
```

The `start` option starts printing to the specified printers.

```
lpc disable printer_name|all
```

The `disable` option turns spooling off in the specified print queue. When spooling is disabled, `lpr` cannot add more requests to the queue. You may want to turn spooling off while testing new line printer filters, since a user logged in as `root` can put jobs in the queue even when the queue is disabled. Another use of the `disable` option is to prevent users from putting jobs in the queue when the printer is expected to be unavailable for a long time.

```
lpc enable printer_name|all
```

The `enable` option turns spooling on in the specified print queue and allows `lpr` to put new printer requests in the queue.

```
lpc restart printer_name|all
```

Occasionally some abnormal condition causes the spooling system daemon to quit unexpectedly. If there is no spooling daemon for a printer, print requests in the queue do not print. You can detect this situation with the `lpc status` command. Enter this command:

```
lpc status
```

The result will look like this:

```
lp:
    queueing is enabled
    printing is enabled
    3 entries in spool area
    no daemon present
```

If you see the line `no daemon present`, use the `lpc restart` command to restart the daemon. This command is useful for restarting `lpd` even when using a network printer.

```
lpc stop printer_name|all
```

The `stop` option halts a spooling daemon after the current job is completed, which disables printing. This is a clean way to shut down a printer for maintenance. Users can still enter jobs in the spool queue while the printer is stopped.

```
lpc up printer_name|all
```

The `up` option enables everything and, if there are any requests in the printer's queue, starts printing.

```
lpc down printer_name|all
```

The `down` option turns off the spooling queue and disables printing. With the `down` option, you can also enter a message that will appear after an `lpc status` command.

Setting up a nondefault printer in the `lpr` system

Follow these steps to add a nondefault ImageWriter printer that is directly connected to your system, to be used through the `lpr` command. For other printers, you may follow these steps, but substitute the proper `printcap` definitions.

1 Log in to A/UX as `root`.

2 Copy the `/etc/printcap` file.

Enter this command:

```
cp /etc/printcap /etc/printcap.original
```

This allows you to restore the original version of the file if something goes wrong while you are editing the file.

3 Edit the `/etc/printcap` file.

The best way to accomplish this for another ImageWriter is to copy the entire ImageWriter default entry and make changes for the new printer. For instance, the original ImageWriter entry looks like this:

```
iw|iw2|ImageWriter II:\
:lp=/dev/printer:br#9600:os#0014001:cs#0004060:fd:\
tr=\f:sd=/usr/spool/lpd/ImageWriter:
```

You may want to change the name, the port, and the spooling directory, so the new entry may look like this:

```
ix|ix2|newiw|New ImageWriter II:\
:lp=/dev/modem:br#9600:os#0014001:cs#0004060:fd:\
tr=\f:sd=/usr/spool/lpd/New-ImageWriter:
```

4 Make the new spooling directory.

For this example it could be `/usr/spool/lpd/New-ImageWriter`. Use the `mkdir` command. Also, use `chmod` to change the permissions to `775` and `chown` to change the owner and group to `daemon`. When you are finished, `ls -l` should print this output:

```
drwxrwxr-x daemon, daemon /usr/spool/lpd/New-ImageWriter
```


5 Enable the printer with the `lpc` command.

Here is the most efficient form of an `lpc` command for this purpose:

```
lpc up newiw
```

6 Test the new printer by printing a file.

Enter this command:

```
lpr -Pnewiw /etc/printcap
```

Writing printer output filters

Filters supplied with A/UX handle printing and accounting for AppleTalk-based printers. For other devices or accounting methods, you may have to create a new filter.

Writing filters requires some understanding of programming and features such as standard input, standard output, and standard error. To see an example of a filter provided with A/UX, look at one of the filter programs in the `/usr/spool/lpd/AppleTalk` directory. The following information is provided for the more advanced system administrator.

Filters are spawned by `lpd` with their standard input being the data to be printed and their standard output the printer. Standard error is attached to the `lf` file for logging errors, or `syslogd` can be used for logging errors. A filter must return one of these exit codes, depending on the circumstance:

- 0 There were no errors.
- 1 The job should be reprinted.
- 2 The job should be thrown away.

When `lprm` sends a terminate signal to the `lpd` process controlling printing, it sends a `SIGINT` signal to all filters and descendants of filters. This signal can be trapped by filters that need to do cleanup operations, such as deleting temporary files.

Arguments passed to a filter depend on the type of the filter. The `of` filter is called with the following arguments:

```
filter -width -length
```

The *width* and *length* values come from the `pw` and `pl` entries in the `printcap` database. The `if` filter is passed the following parameters:

```
filter [-c] -wwidth -llength -iindent -n login -h host accounting-file
```

The `-c` option is optional and is supplied only when control characters are to be passed uninterpreted to the printer (when the `-l` option of `lpr` is used to print the file). The `-w` and `-l` options are the same as for the `of` filter. The `-n` and `-h` options specify the login name and host name of the job owner. The last argument is the name of the accounting file from `printcap`, specified by the *accounting-file* variable.

All other filters are called with the following arguments:

```
filter -xwidth -ylength -n login -h host accounting-file
```

The `-x` and `-y` options specify the horizontal and vertical page size in pixels (from the `px` and `py` entries in the `printcap` file). The rest of the arguments are the same as for the `if` filter. For more information on filters, see the `printcap` on-line manual page.

Printing files

You can use the `lpr` command to print files created with text editors (such as `vi` or `TextEditor`). A/UX spools the files to be printed, printing them in the order in which they are received.

Note: The `lpr` spooler doesn't format documents. You must use a formatting command such as `pr` or `troff` to format documents before you send the job to the printer. For further information, see `troff(1)`.

Sending a file to a printer connected directly or through AppleTalk

To test the `lpr` command on a printer that is connected to your computer directly or through an AppleTalk network:

- 1 Use the Chooser to select the printer.
- 2 Choose CommandShell from the Application menu.

3 Enter this command:

```
lpr /etc/passwd
```

This command sends the file `/etc/passwd` to the printer.

Sending a file to a remote UNIX printer

Once you have set up TCP/IP networking as described in Chapter 6, you can test printing to a printer connected to another UNIX system.

To test printing using the `lpr` command:

1 Be sure that your `/etc/printcap` file is set up.

For information on setting it up, see “The `printcap` Database” earlier in this chapter.

2 Choose CommandShell from the Application menu.

3 Enter this command:

```
lpr -Pprinter /etc/passwd
```

Replace *printer* with the name of the printer that you want to use. This command sends the file `/etc/passwd` to the remote printer specified by *printer*.

Finding out how many files are waiting in the printing queue

A/UX spools printing jobs, placing them in a queue. The printer handles them on a first-come, first-served basis. If you share your printer and there are other jobs ahead of yours, those jobs are printed first. Your file is printed when your turn comes.

■ **To display a list of the print jobs in the printing queue, enter this command:**

```
lpq
```

A list appears that looks something like this:

Rank	Owner	Job	Files	Total size
active	jms	3	<i>filename</i>	720 bytes
1st	pzp	4	<i>filename</i>	1500 bytes
2nd	alp	5	<i>filename</i>	2451 bytes

The first column shows the rank, the second shows the owner of the file sent to the printer, the third shows the identification number of the print job, the

fourth column shows the filename (including its path), and the last column shows the size of the file.

Canceling a printing job

To cancel a printing job before it's completed but after you've sent it to the printer, use the `lprm` command with the ID number of the printing job.

- **Enter this command:**

```
lprm ID-number
```

Substitute the ID number of the printing job (such as `lprm 398`) for *ID-number*. The ID number is in the third column (labeled `Job`) in the list of printer jobs shown in the previous section.

The following message appears as A/UX cancels the job:

```
ID-number dequeued
```

If you are logged in as `root`, you can cancel any print job. Otherwise, you can only cancel jobs that you have submitted to print.

For more information about the `lpr` command, see `lpr(1)` in *A/UX Command Reference*.

8

User and Group Administration for A/UX Users

A number of factors affect how A/UX interprets and executes commands issued by a user. These factors are the user's file permissions, group identification, login shell, and other environment variables typically set during startup. When setting up an A/UX user account, the administrator defines these factors, which make up the user's working environment.

This chapter discusses the user's environment, and includes

- what the A/UX user's environment is, and how to change it
- how A/UX establishes the environment
- administrative logins and groups
- how to add an A/UX user
- what to consider when defining an A/UX user
- simple and advanced features of file permissions

Users and groups defined in the A/UX environment are in no way related to the users and groups defined in AppleShare. For information on AppleShare users and groups, see the *AppleShare Pro Administrator's Guide*.

The A/UX working environment

Each login account can be set up to define the environment in which it works and to provide the following features:

- *A secure place to work:* To log in to A/UX, a user must first enter his or her login name and password. After the user logs in, an alias to the user's home directory folder is on the desktop (unless the user is `root`, or the folder has been put away by means of the Put Away command in the File menu). This arrangement permits a private environment in which each user has control over who has access to his or her work.
- *The ability to share tools and data:* A/UX also provides mechanisms by which users can share their work with other users. This is done mainly through the formation of groups of users who perform common tasks. Through the prudent use of groups, the server administrator can set up environments that permit the appropriate mixture of resource sharing and security.
- *The ability to customize:* Users can modify many features of their working environments by making changes to specific files located in their home directories. This gives them the power to personalize their environments. The server administrator, however, is responsible for the initial setup and for security-related modifications.

An introduction to the working environment

A number of factors determine the interpretation and operation of commands given by a user logged in to an A/UX system. This guide refers to these factors as the *user's working environment*. Some additional terms are used throughout this chapter to describe factors of a user's working environment:

- *Permissions:* Every file in a UNIX file system has UNIX-based permissions that determine who can see, open, or change the file. UNIX permissions treat program files, data files, input and output devices, and even directories merely as "files." Three actions can be performed on files: read, write, and execute. Permissions allowing a user to perform each of these actions can be set for three types of users: the user who created (or owns) the file, users belonging to the same group as the group affiliation of the file, and all other users. Each kind of permission can be specified differently for each kind of user. Users can execute programs or have access to files (including directories) only when specific permissions are granted.

Permissions in the Finder environment (See Files, See Folders, and Make Changes) are conceptually similar, though not identical, to UNIX permissions. AppleShare permissions and A/UX permissions are completely separate, as are the procedures for setting them.

- *Login name and password:* If you configure your system to require a user to log in, the user must enter a valid login name and password before he or she will be able to access the system. Login names and the user's initial shell are defined in the `/etc/passwd` file. (See "The `/etc/passwd` File" later in this chapter.)
- *User ID:* Each user login name is associated with a unique numeric user identification number (UID) that is recorded in the `/etc/passwd` file and identifies the user. The system administrator defines the UID. When a user attempts to use a command or gain access to a file, access is allowed or denied depending on the access permissions that are set on the file relative to the user's UID.
- *Group ID:* Each user login name is associated with at least one group identification number (GID). A user's default, or login, group ID is defined in the `/etc/passwd` file, and the valid groups are defined in `/etc/group`. The group ID indicates the groups to which the user belongs at the time of login. Group membership is a security feature that permits a specific group of users access to files, while denying this access to users who are not members of the group.
- *Home directory:* Each login name must be associated with a directory that serves as a starting point when the user logs in to the system. This directory is defined as the user's home directory. The server administrator creates home directories when adding user accounts to the server. (See "Administering User Accounts" later in this chapter.) Users also return to this directory as the default when they execute the `cd` command with no arguments. The specific location of the user's home directory depends on how his or her system is set up, but a recommended location for all users' home directories is in the `/users` (not `/usr`) directory. Users can customize their working environment by modifying setup files in their home directories.

- *Current directory:* The current directory is the one in which the user is working within the UNIX hierarchical directory structure. Every time the user changes to a new directory, the new directory becomes the current directory. When a user specifies a relative filename (a filename that does not start with a /), the current directory is searched for a file of that name.
- *Setup files:* Applications can have setup files in the user's home directory. Setup files usually begin with a period. (They are also referred to as "dot files.") For instance, you can have a `.mailrc` file to set your mail preferences and a `.xdefaults` file to set your X Window System™ preferences.
- *System Folders:* Users can configure their own version of the Macintosh environment through the use of a personal System Folder. If an account does not have a System Folder in its home directory, the global System Folder in `/mac/sys` is used.
- *Shell:* After logging in, the user communicates with the system through a shell. There are several shells: the C shell, the Bourne shell, and the Korn shell. After a successful login, the shell defined in the `/etc/passwd` file runs and executes a series of setup files that further define the user's working environment. The setup files are discussed in "Command Shells and Setup Files" later in this chapter. `CommandShell` is the application through which the user interacts with a shell.

If the `/etc/passwd` file does not specify a shell, the shell is the Bourne shell. The `adduser` command, which simplifies the process of adding a user to your system, uses the C shell as the shell. (See "Adding a User," later in this chapter, or `adduser(1M)` for more information.)

More restrictive programs can be used as the shell; for example, `/bin/rsh` limits system access. You can also set up an account to execute a single command and then log the user out. See "Changing a User's Shell," later in this chapter, for more information.

- *Environment variables:* The three shells provide a number of shell environment variables that can be assigned different values to alter the user's working environment. The three shells use different variables. Some environment variables are automatically assigned values from the `/etc/passwd` entry for each user. These include `LOGNAME` (login name),

HOME (home directory), and SHELL (login shell). Other variables, such as PATH (search path), TERM (terminal type), and EDITOR (preferred editor program) are assigned values from files such as `.login`, `.cshrc`, `.kshrc`, or `.profile` in each user's home directory and are read when a user logs in.

New environment variables in A/UX 3.0.1

A/UX 3.0.1 includes three new environment variables designed to help you configure your compatibility with both Macintosh and UNIX environments. See `startmac(1M)` in the on-line manual pages for more information on these variables.

- **TBTRANSLATESLASHES** This variable affects how a slash (/) is translated when it is in the name of a file in the Finder. If this variable is not defined in your environment, a slash is translated to the underscore character (_) and stored in the file system that way. If this variable is assigned any value in your environment, A/UX 3.0.1 maintains the slash (/) in the filename by storing it as a colon (:) in the file system. (You will see a slash in the filename when viewed from the Finder, and a colon in the file name when viewed from CommandShell.)
- **TBFMIGNORECASE** This variable affects the case sensitivity of the names of files. Traditionally, `Filename` and `filename` are considered the same file by the Macintosh Finder, but they are considered different by UNIX. When this variable is assigned a value, `Filename` and `filename` are considered to be the same file. If you would like to have case-sensitive filenames, do not define this variable in your user environment.
- **TBTRANSLATEUXONLY** This variable affects how carriage returns (CR) and line feeds (LF) are stored in the system. A/UX 3.0 converts carriage returns to line feeds, and then tries to translate the line feeds back to carriage returns when the file is reopened. A/UX 3.0.1 avoids possible complications that this may cause and allows you to set this variable. If this variable is assigned any value in your environment, carriage returns are translated to line feeds upon a file save of UNIX text files. (Files created in the CommandShell as well as by TextEditor are considered UNIX text files.)

Shells and setup files

The setup files for the C shell (`/etc/cshrc`, `.cshrc`, `.login`, `.logout`) and for the Bourne and Korn shells (`/etc/profile`, `.profile`, and `.kshrc`) are discussed in this section.

The suggested default copies of these setup files are stored in the directory `/usr/lib/skel`. When a new account is created with the `adduser` program, these files are copied to the new user's home directory. These files can be viewed or edited.

Note: When `$HOME` precedes a setup filename, as in `$HOME/.login`, it represents the user's home directory.

If you log in using 32-bit or 24-bit mode (using the Options menu while the Login dialog box is displayed), your commands are processed by the Finder that operates in the A/UX environment.

If you log in specifying Console Emulator mode, a console window appears, and you bypass the Finder. The Console Emulator window displays an A/UX command-line prompt, and you are placed in your home directory.

You can also work directly from the A/UX command-line interface rather than from a Finder interface after you log in by selecting `CommandShell` from the Application menu. This action creates a `CommandShell` window and displays an A/UX command-line prompt. In these windows, you enter traditional UNIX commands and receive responses from the system through one of the command shells.

The command-line prompt

By default, the system host name is displayed in the command-line prompt along with the user login name. The command-line prompt is set by an entry in your default login script, which is `.profile` for Bourne shell or Korn shell users, or `.login` for C shell users.

If your system's host name is `picasso` and you have logged in as `root` (and you have not changed the contents of your login script), you will see the following prompt in your shell:

```
picasso.root#
```

The C shell setup files

When a user logs in and `/bin/csh` is specified as the shell in the user's `/etc/passwd` entry, the system automatically runs several shell scripts. The first of these scripts is `/etc/cshrc`. This is a script of shell commands that typically sets certain shell variables and sets a file creation mask. (See “Setting Default File Permissions With `umask`” later in this chapter.) This file is readable by all users but cannot be modified by users who do not have the permissions of `root`.

The `/etc/cshrc` file is run *before* the `.cshrc` file in the user's home directory. A user can override any definitions set by the execution of `/etc/cshrc` by redefining the variable in his or her own `.cshrc` or `.login` file.

The `.login` script is run after `.cshrc`. It is typically used to set up terminal defaults and environment variables. After the initial login, whenever the user reinvokes the C shell, the `.cshrc` file is run again; `/etc/cshrc` and `$HOME/.login` are run automatically only at login. Therefore, users should place commands that need to be executed only once in the `.login` file in their home directory, and environment variables they want in all their shells in the `.cshrc` file.

Under the C shell, when a user logs out, the commands in his or her `.logout` file are executed.

The Bourne shell setup files

When `/bin/sh` is specified as the user's command shell in `/etc/passwd`, the first of the shell scripts run is the file `/etc/profile`. Similar to the `$HOME/.profile` file, this is a script of shell commands, which typically exports certain shell variables and sets a file creation mask. This file is readable by all users but cannot be modified by users who do not have the permissions of `root`.

The `/etc/profile` file is run *before* the file `.profile` in the user's home directory and thus serves as a default `.profile` file. A user can override any definitions set by the execution of `/etc/profile` by redefining the variable in his or her own `.profile` file.

The Korn shell setup files

When `/bin/ksh` is specified as the user's command shell, the system runs the `/etc/profile` and `$HOME/.profile` files described previously. If one of these files sets the `ENV` variable to any filename (`ENV` equals `$HOME/.kshrc` in the standard distribution), the system also executes the contents of the named file. After the initial login, whenever the user starts a new shell (usually by opening a new CommandShell window), the file named in `ENV` is run again. (Note that `/etc/profile` and `$HOME/.profile` are not rerun.)

Using System Folders to customize environments

Users can create their own System Folders within their home directory folders by using the `systemfolder` program. If a user does not have a System Folder in his or her home directory, the global System Folder `/mac/sys/System Folder` is used. Although individual System Folders consume disk space, a user might choose to create a personal System Folder for the following purposes:

- to maintain the user's own file-sharing settings (export list) and users and groups file
- to customize the user's desktop, including fonts and system extensions (INITs)
- to work in an environment other than that set by the global System Folder

Administrative login accounts and groups

When first installed, A/UX contains two user login accounts. The first is `start`. This login account provides a starting point for users who are new to A/UX and are using the introductory tutorials.

A/UX also provides a `Guest` account, which lets a guest user have quick access to the system to run Macintosh programs. Because both the `start` and `Guest` accounts allow easy access to the system, you may wish either to remove them or to give them a password to provide better system security.

Note: Users and groups defined in this chapter are in no way related to the users and groups defined in AppleShare.

Administrative login accounts

Login accounts that are used only by system administrators or programs that perform specialized system tasks are called administrative login accounts. The `root` account is the only administrative account you will generally use; other administrative accounts are used by programs. If you accidentally remove any of the administrative login accounts, the system may operate strangely or not at all. The default administrative login accounts, found in the `/etc/passwd` file, are shown in the following list:

<code>root</code>	The user on the system that can change any permissions and perform any actions. As shipped, the home directory is <code>/</code> and the shell is <code>/bin/sh</code> . For better security, avoid using nonstandard shell environments, and do not change the home directory of <code>root</code> .
<code>daemon</code>	The owner of certain noninteractive background processes that handle persistent system services, such as network communication and the print spooler.
<code>bin</code>	The owner of most commands and the system directories in which those commands are stored.
<code>sys</code>	The owner of certain system files, such as <code>/etc/zoneinfo</code> .
<code>adm</code>	The owner of most system accounting programs and directories, in particular <code>/usr/adm</code> . System Accounting is an optional package on the system software installer CD-ROM.
<code>uucp</code>	The owner of programs and directories associated with the optional communications package, UUCP.
<code>nuucp</code>	The login account optionally assigned to incoming UUCP requests. UUCP is a custom-installable package and is described in <i>A/UX Network System Administration</i> , a manual in the A/UX 3.0 suite.

lp	The owner of commands and processes associated with the lp line-printer spooling and printing package. The lp system is also a custom-installable package and is documented in the A/UX 3.0 documentation suite in <i>A/UX Local System Administration</i> .
nobody	Assigned as the default login for remote root access under the optional package Network File System (NFS). See <i>A/UX Network System Administration</i> for further information.

Administrative groups

A/UX allows you to combine multiple users who have similar tasks, and therefore similar access requirements, into a group. Using administrative groups helps you perform specialized system tasks by allowing increased access permissions without the full permissions of root. As mentioned earlier, the only administrative login account you want to log in to is root. However, when performing administrative tasks, you don't always want to use an account with all the permissions of root. One way to avoid this is to create a special user login account and make that user a member of an administrative group.

If, for example, user dataadm is a member of the group dba (for database management), and group dba owns database administrative files; then the system administrator can log in to the dataadm account and run database administrative programs. Since the dataadm login is only affiliated with dba administrative permissions, you avoid the risks of making an error with the global permissions of root. See "Administering Groups," later in this chapter, for more information.

Administrative groups are found in the `/etc/group` file and include `sys`, `bin`, `root`, `daemon`, `adm`, `uucp`, `lp`, `utmp`, and `mail`.

Administering user accounts

The A/UX system administrator is responsible for providing adequate security for users as well as providing information. The default security administered by the `adduser` program should prevent other users from reading or writing to a new user's area. Individual users may, of course, override this initial setup, but relaxation of security is an option, not the default. To ensure security, the administrator should understand the entries in the `/etc/passwd` and `/etc/group` files.

The `/etc/passwd` file

Each user's account is specified by a single entry in the `/etc/passwd` file. You must be logged in as `root` to open this file with an editor and modify it. However, regular users can change their default passwords or shells with the `passwd` or `chsh` command.

The `/etc/passwd` file distributed with A/UX has several administrative login accounts and the user login accounts `start` and `Guest`.

Format of `/etc/passwd`

Each entry in `/etc/passwd` consists of one line with seven fields separated by colons. The following entry is the `/etc/passwd` entry for a user named Jenny Winton:

```
jenny::1001:1001:Jenny Winton,Rm 1,x12,:/users/jenny:/bin/csh
```

The form of an entry is as follows:

```
login-name:password:UID:GID:misc:home-directory:login-program
```

In this entry, the fields are interpreted as follows:

login-name The name the user must use when logging in. This name must be unique in the `/etc/passwd` file and should be lowercase and no longer than eight characters. Typically, the name indicates the person who uses it, such as `ronald` or `mary`.

<i>password</i>	An encrypted version of the actual password the user must use when logging in. The encrypting is done automatically when the password is first assigned and whenever it is changed. If the field is empty, as shown in the example, no password is necessary to log in to that user account.
<i>UID</i>	A unique user identification number for each user.
<i>GID</i>	The user's default group identification number. Even if the user is listed in several groups in the <code>/etc/group</code> file (as discussed in "The <code>/etc/group</code> File," later in this chapter), he or she belongs by default to the group whose number appears in this field in the <code>/etc/passwd</code> file.
<i>misc</i>	Miscellaneous information about the user, such as full name, office, and telephone numbers.
<i>home-directory</i>	Whenever the user logs in to the system, this is the directory in which he or she is initially located.
<i>login-program</i>	The name of an executable program, usually one of the shells, that permits the user to communicate with the system.

Any program can serve as *login-program*. For instance, the last field of the `/etc/passwd` file can be a program such as `/bin/who`. If `/bin/who` is the startup program, the account allows a user to log in and view the output of the program `/bin/who`; the user is then logged out automatically.

Although `/bin/who` is not a very useful working environment, other programs, such as the restricted shell, `rsh`, may be. The `rsh` program allows a user the use of a shell within the home directory but allows no directory changes. See `sh(1)` for more information on `rsh`.

For more information on the `/etc/passwd` file, see `passwd(1)` in *A/UX Command Reference* and `passwd(4)` in *A/UX Programmer's Reference*.

Adding a user

Adding a user to your system is a two-step process: First, you plan the user's working environment; and second, you add the user according to your plan. The planning stage is important; neglecting it can cause both security and administrative problems.

After you plan the new user's working environment, you can use the `adduser` command to easily add a user. The `adduser` command creates the entry in `/etc/passwd`, creates or copies the necessary directories and files, and sets access permissions for the new user.

This section presents recommended planning steps and then gives instructions on using the `adduser` command in CommandShell.

Planning a user's working environment

To plan a user's working environment:

1 Keep a hard-copy record of data about the new user.

The information you will need to add a new user's working environment is as follows:

- date you add the user (month/day/year)
- user's real full name
- user's telephone number (office)
- user's telephone number (home)
- user's login name
- user identification number (UID)
- group identification number (GID)
- group name
- full pathname of the user's home directory
- full pathname of the user's shell

2 Record the full name and telephone numbers of the user, along with the date you add the user.

You can choose to use one or both of the home and office numbers. This information becomes available to the `finger` command, which reports this information.

3 Pick a login name for the user.

Login names usually consist of all lowercase alphabetic characters. A maximum of 15 characters is allowed for a local login, but it is recommended that you use only 8 characters for maximum compatibility across networks. Each login name must be unique.

When you decide on an unused name, record it on your hard copy.

4 Select a user identification number.

The `adduser` command will choose a new UID for you, or you can choose one yourself. Pick a number that is not being used and write it on your hard-copy report. By convention, ID numbers under 100 are reserved for special uses, such as for special system functions. The `adduser` command chooses numbers starting at 1000.

If you plan to allow A/UX users access to more than one system on a network through TCP/IP utilities (as opposed to AppleShare), you should consider assigning the same UID for the same user on each system. This will simplify user administration.

5 Select a group identification number.

By default the `adduser` command will put each new user into a newly created group, but you can also specify a group ID to which you assign a user. See “Administering Groups,” later in this chapter, for information about planning, selecting, and specifying group membership.

6 Select a home directory.

You should always use absolute, rather than relative, pathnames when you enter paths into `/etc/passwd`. By default, `adduser` will put a user’s home directory into `/users`. Therefore an account with login name `jenny` would have a home directory of `/users/jenny`.

7 Select a shell.

You may want to ask if the user has a preference. If he or she does not have a preference, or isn't sure of the differences between the C shell, the Bourne shell, and the Korn shell, choose the C shell. In that case, you would enter `/bin/csh` in this part of the record. See "Changing a User's Shell," later in this chapter, for information about using different command interpreters as a user's shell.

Adding users

This section describes how to add a user to your system using the `adduser` command. For more information, see `adduser(1M)`. In the example, a user named Jenny Winton is added to the engineering group. Only the user logged in as `root` can add users to the system.

To add users:

1 Log in to A/UX as `root`.

2 At the prompt, enter this command:

```
adduser
```

3 You are then prompted to enter the login name of the user you wish to add to the system.

```
Enter user's login name (e.g. fred) jenny
```

4 Enter the user's full name when the following prompt appears:

```
Suggested values are shown in brackets.
```

```
To accept, press RETURN; otherwise, type the new value.
```

```
User's full name (e.g. Fred Smith) [] Jenny Winton
```

5 If you want to, you can add information to the password file about the user's address or mail stop and phone numbers. Press RETURN if you don't want to enter the information.

```
Office address (e.g. mail stop) [] Eng-25
```

```
Office telephone (extension) [] x0116
```

```
Home telephone [] 555-1213
```

- 6 Enter the user's initial group when you are asked to designate the user's initial group affiliation, as follows:**

Initial group (create a new group by entering a new name).

Currently available groups are:

Special (system) groups:

root	daemon	bin	sys	adm	uucp
utmp	lp	mail	nuucp		

Regular (user) groups:

staff	perm	temp	contract	guest	project
-------	------	------	----------	-------	---------

Initial group [gp1000] **eng**

- 7 Enter the shell the user will use.**

Shell [/bin/csh] **/bin/ksh**

- 8 Enter the user's home directory. (You can press the RETURN key to accept the default value.)**

Home directory [/users/jenny] **/users/jenny**

- 9 Decide whether you want the user to have a personal System Folder, or you want the user to use the global System Folder in the /mac/sys directory.**

Create a personal System Folder? [yes] **yes**

- 10 Double-check your entries, and enter yes if you made no mistakes.**

The account for jenny will be created with the following attributes:

Login name: jenny (uid 1000) Real name: Jenny Winton

Office address: Eng-25 ext: x0116 Home phone:555-1213

Home directory: /users/jenny Shell: /bin/ksh

Group eng (gid 1000) *This is a new group*

A personal System Folder will be created.

OK to create account? [yes] **yes**

- 11 **Decide if you want to enforce password use, and enter `yes` if you do.**

```
Creating personal System Folder for jenny ...
```

```
Require user jenny to set password on initial login? [yes]
```

```
yes
```

- 12 **Press the RETURN key to quit, or enter the next user's name to add another user.**

```
Enter next user's login name (e.g. fred)
```

```
[enter empty line to quit]
```

You now would have an entry in `/etc/passwd` like this one:

```
jenny:,:1000:1000:Jenny Winton,Eng-25,x0116,555-1213:/users/jenny:/bin/ksh
```

You will also have an entry in `/etc/group` that reflects the new group you have made:

```
eng:*:1000:
```

Removing users

Removing a user from your system can be as simple as opening the `/etc/passwd` file and inserting the word `VOID` in that user's password field. However, if the user has created many files that must be saved, you may need to find all files owned by the user, back them up, examine each of them, determine who else uses the files, change the ownership of shared files, remove links, and finally delete the user's password entry.

This section introduces the simplest form of user removal first and then discusses additional steps to remove all files and directories associated with that user.

Voiding a user account

The first step in removing a user from your system is to deny the user access to it. The best way to do this quickly is to edit the user's `/etc/passwd` entry and enter the word `VOID` between the first and second colons (`:`). This (second) field is the password field. This makes it impossible for anyone to log in as that user, although that user's files remain unaffected.

IMPORTANT Do not leave the password field blank. A blank password field can result in a serious security breach; anyone can log in to the system through an account without a password.

It is not a good idea to delete the whole `/etc/passwd` entry for the user. If you do, you affect the files owned by that user. Commands that use login names as arguments (for example, `chown` and `find`) or that print information relating to login names (for example, `ls -l`) check the `/etc/passwd` file for the user login and UID number. For instance, if you have deleted a user's entry, and then run `ls -l` on files owned by that user, there is no login name listed as a file's owner; instead, it is replaced by the number that is the UID of the (now deleted) user. If you delete a few `/etc/passwd` entries, you may get confused about which files belonged to which former user.

Deleting a user account

In general, it's a good idea to back up a user's files before deleting them, for two reasons:

- These files may contain information that you will need later.
- These files may be used by other users on your system.

To properly delete a user account:

1 Void the user's password.

Edit the user's `/etc/passwd` entry and enter the word `VOID` between the first and second colons (`:`). For example, in the previous example of the user Jenny Winton, you would replace the characters `, . .` with the characters `VOID`.

2 Find all the files belonging to the user, regardless of their location, with this command:

```
find / -user login-name -print > filename
```

In this command line, *login-name* is the user's login name and *filename* is an arbitrary name. This command creates a file, named *filename*, which contains a list of all files on the system that are owned by the user *login-name*. Use this file as a reference while completing the following tasks.

3 Back up the files, using your favorite backup program.

- 4 **Delete the user's files only after finding out if anyone is currently executing any commands or using any data files owned by that user.**

Inquire personally or through `mail` to find out if any others regularly use files created by that user. If they do, change the ownership of those files. If a file is linked to a file owned by that user, remove the link. Then delete the files.

Alternatively, you may wish to use `chmod` and `chgrp` to assign ownership of these files to active users.

Moving a user's home directory within a file system

Sometimes you must move a user's home directory and files. There are several ways of doing this, and the method you choose depends on the characteristics of the move. If you do move a user's home directory, remember to change the home directory field in the user's entry in `/etc/passwd`.

The simplest move is the one that involves moving a user's directory to another place in the same file system. The command line

```
mv old-directory new-directory
```

moves *old-directory* (including all of its files and subdirectories) to *new-directory*.

Changing a user's shell

The last field in the user's entry in the `/etc/passwd` file determines the user's startup program. Typically, the field is `/bin/csh` (the default when using `adduser`), `/bin/sh`, or `/bin/ksh` (for the C shell, Bourne shell, and the Korn shell, respectively). If there is nothing in an `/etc/passwd` entry after the sixth colon (:), the shell `/bin/sh` is used.

To change a user's startup program, you need to change this last field and to make sure that the user has the proper startup files for the new shell. Either `root` or a user can execute the change-shell command, `chsh`, to change the startup program to any legal value, as listed in `/etc/shells`. (A user can only change his or her own shell; a user logged in as `root` can change any user's shell.) Because the `chsh` command actually changes the entry in the `/etc/passwd` file, other modifications to the user's working environment may be necessary, particularly with regard to shell startup files in the user's home directory. (See `chsh(1)` in the on-line manual pages.)

Administering groups

Defining groups of users on a system provides a way of combining user access permissions for people who have the same interests or responsibilities. Before you add a user, you should have a clear idea of what his or her tasks will be, which other users are currently engaged in similar activities, which parts of the system you want the user to have access to, and whether a new group should be created.

Creating groups

The groups you create will generally comprise users whose members have similar tasks, such as legal, engineering, publications, and so on. Examples of group names include `legal`, `eng`, `pubs`, `acctg`, and `mkt`. Group names are arbitrary, but are usually related to the group's activity.

When you create groups, you add an entry to the `/etc/group` file that includes a group ID number. Group ID numbers can be any value you wish, but there are some practical guidelines to consider:

- Create group numbers in round hundreds or thousands, as shown in the following example of `/etc/group` entries:

```
mktg:*:300:ronald,jenny,ng
eng:*:5000:jenny,jim,seth
```

- When you add users to the `passwd` file, you may want to relate their user ID (UID) to the group they are in, as in this example:

```
jenny:aX3eopRx:301:300::/user2/mkt/jenny:/bin/csh
ronald:ZzXpL3eox:302:300::/user2/mkt/ronald:/bin/csh
jim:7IoiudSSe:5012:5000::/user2/eng/jim:/bin/ksh
```

By convention, user ID numbers under 100 are reserved for special uses, such as for special system functions; therefore, the group ID number chosen should follow the same guidelines. The `adduser` command chooses numbers starting at 1000, by default. You may specify to `adduser` which number to use for both a user and group ID.

If you organize your groups and users systematically, your system will be easier to administer as it grows and expands.

The `/etc/group` file

The GID field of a user's entry in the `/etc/passwd` file establishes a single default group for the user. The `/etc/group` file is used to establish multiple group memberships for a user.

The `/etc/group` file contains entries with four fields separated by colons. Here is the form of the entry:

```
group-name:password:GID:list
```

In this entry, the fields are interpreted as follows:

<i>group-name</i>	The group name. Group names are arbitrary, but by convention their meanings should be self-evident (for instance, <code>acctg</code> rather than <code>xyz24</code>).
<i>password</i>	An encrypted version of the password for this group. Traditionally, this field is not used in UNIX. There is no standard software to give groups passwords. It is common practice to disable group password checking by filling the <i>password</i> field with the word <code>VOID</code> or with asterisks (<code>*</code>). Setting proper group permissions on files and directories makes this field superfluous. However, you should not leave this field blank, as that would allow anyone to use the <code>newgrp</code> command to become a member of this group.
<i>GID</i>	A unique number set for each group. For each group ID there is only one group name, and vice versa. The actual group ID numbers that exist in the <code>/etc/group</code> file are the only numbers that should be entered in the <i>GID</i> field of <code>/etc/passwd</code> entries. The group ID entered for each user in <code>/etc/passwd</code> should coincide with the group ownership of that user's home directory.
<i>list</i>	A list of the login names of the members of the group. The login names must be separated by commas. Entering a user's login name in the group's <i>list</i> field is optional for those users who belong to only the group indicated in the <i>GID</i> field of the <code>/etc/passwd</code> file.

Multiple group memberships

The way A/UX handles groups is derived from the method used by the Berkeley version of the UNIX operating system, and this method differs from the way the System V version of the UNIX operating system handles groups. (In the System V version, the user is allowed to be in only one group.) In A/UX, a user can be in a maximum of eight groups. The system administrator enters the groups to which a user belongs in the file `/etc/group`. To list your group memberships, enter the command `groups`.

It is possible for a user to be a member of several groups. When a user who is a member of more than one group creates a file or directory, the group associated with the parent directory becomes the group associated with the new file or subdirectory. You will notice in the earlier example that the user `jenny` is listed in both `mktg` and `eng`. If the group ownership of her home directory is `mktg`, then every file she creates and every directory she makes below her home directory will also belong to `mktg`. For various reasons, she might want some of the files she creates to belong to the group `eng`. There are two ways she can accomplish this:

- She can create a file, which has group ownership `mktg` by default; she can then change the file's group membership to `eng` after the fact.
- She can create a directory that has group ownership `mktg`; she can then change the group membership of the directory to `eng`. Every file created within that directory will then also belong to `eng`. (This is true in UFS file systems only; in SVFS file systems, the new file takes on the group of the creator.)

If a user belongs to eight groups and temporarily needs to be in yet another group, he or she must enter `newgrp groupname`, where `groupname` is an entry in `/etc/group`. In order to use the `newgrp` command, you must be listed as a member of the group in `/etc/group`, or know the password of the group. This causes `groupname` to replace the first group listed in your environment for the duration of the login session. Note that a user's group membership is still restricted to eight groups. The `newgrp` command temporarily substitutes the new group in place of the first group listed in your environment.

Files, directories, folders, and permissions

For an introductory discussion of permissions and information on how they are represented in the Finder, see *Basic Skills*. The following section discusses the UNIX representation of permissions as seen from the CommandShell.

File permissions

UNIX file access is designated by read, write, and execute permissions for owner, group, and others. While similar in concept, these permissions are different and separate from those enforced for AppleShare accounts.

By default, security for accounts set up with the `adduser` command prevents other users from reading or writing to a user's home directory.

When you display permissions by using the `ls -l` command, the output of this command might be

```
-rwxr-xr--  jenny  eng  512  Oct  3 17:51  importantfile
```

The file permissions that appear at the beginning of the line can be interpreted as shown in Figure 8-1. For the file `importantfile`, the user `jenny` has read, write, and execute permissions; members of her group (`eng`) have read and execute permission (but cannot change the file); all other users on the system can only read the file.

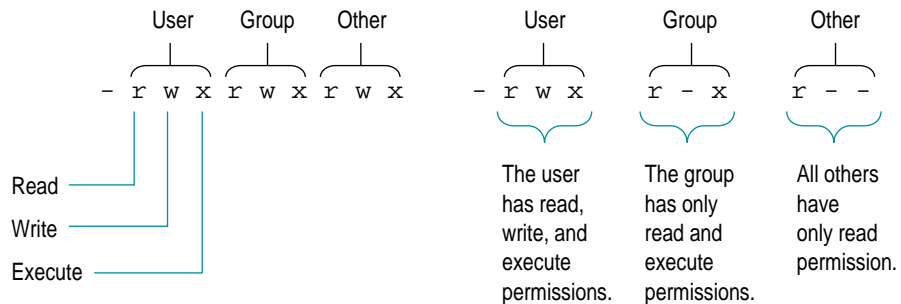


Figure 8-1 Access classes

The initial character, shown as a hyphen (-), represents the file type. Table 8-1 lists the various file types. For more information on different file types, see `chmod(1)`.

Table 8-1 File-type characters

Type	Description
-	A regular file that contains data.
d	The file is a directory.
c	The file represents a character device.
b	The file represents a block device.
p	The file represents a named pipe.
s	The file represents a socket.
l	The file represents a symbolic link to another file.

Directory and folder permissions

Permissions work a little differently when they are assigned to a directory than they do when assigned to regular files. Here is a list of directory and folder permissions and their meanings:

- r Allows you to list the names of the files that are in the directory (`ls`).
- w Allows you to add or delete entries from that directory.
- x Allows you to search the directory, or to make it the current directory (`cd`).

Setting permissions on a directory affects only the directory itself and does not change the permissions settings of any of the directory's files or subdirectories.

Directory permissions are among the most important aspects of the user's environment. For example, *file* permissions that protect against reading or writing by other users are not enough to protect the file from being deleted, if the *directory* permissions allow other users write permission. Similarly, if the directory grants the group read permission, its files can be listed (`ls`) by a group member even if the files themselves deny group read permission and cannot be opened.

Group membership is an important consideration for the administrator setting up directory permissions. The default group membership of a file or directory is the same as the group membership of the directory in which the file is created. This allows for the creation of hierarchies of directories according to their group membership.

Directory permissions can affect the ability to access a file. If a wildcard character (such as `*` or `?`) is used in the path specification, read permission will also be required for the affected directory. This is a result of the wildcard's causing the shell to read the directory (on the user's behalf) to find the requested file. You can remove read permission from a directory in order to prevent snooping.

Changing file permissions: `chmod`

Only the owner of a file or a user logged in as `root` can change a file's permissions using the `chmod` command. Anything that `chmod` can do to a file's permissions it can do to a directory's permissions as well, because UNIX-based systems treat a directory as a file. Note that the `chmod` command does not apply to files and folders on Macintosh file systems.

The `chmod` command can be invoked with either symbolic or numeric terms.

Symbolic terms

Symbolic terms are straightforward: `u` stands for the user (that is, owner) of the file, `g` stands for group, and `o` stands for others; `+` represents granting permission, and `-` represents denying permission. The format for running `chmod` with symbolic terms is as follows:

```
chmod access-class operator permissions filename
```

Note that, on the command line, there are no spaces between the *access-class*, *operator*, and *permissions* fields.

<i>access-class</i>	One or more of the three access classes—user (<code>u</code>), group (<code>g</code>), or other (<code>o</code>). In addition, the access class all (<code>a</code>) lets you grant or deny permissions to all three access classes simultaneously.
---------------------	---

<i>operator</i>	Grants access permission (the + operator) or denies it (the - operator). You can't both grant and deny permissions in a single command. You must grant permissions to one access class in one command, then deny it to another access class in a second command.
<i>permissions</i>	Read permission (r), write permission (w), and execute permission (x). You can grant (or deny) more than one type of permission at the same time. You can also include setuid or setgid (s) and sticky bit (t), discussed in "Setting Command Permissions Using Setuid and Setgid Bits" later in this chapter.
<i>filename</i>	The file or files whose permissions are to be changed. You may use absolute or relative pathnames.

For example, to change the permissions of a file you own, named *filename*, from

```
-rw-rw-rwx
```

to

```
-rwxrw-r--
```

the sequence of commands is as follows:

1 Grant execute permission to the owner:

Enter this command:

```
chmod u+x filename
```

2 Deny write and execute permissions to all others:

Enter this command:

```
chmod o-wx filename
```

3 You can verify that the permissions have been changed:

Enter this command:

```
ls -l filename
```

Numeric terms

Numeric (or absolute) terms are based on octal numbers where, for each access class, the mode of the file is set according to its octal sum. It sounds complex, but it's actually rather simple if you understand basic addition. Figure 8-2 shows how you can easily translate the desired file permissions into numeric terms.

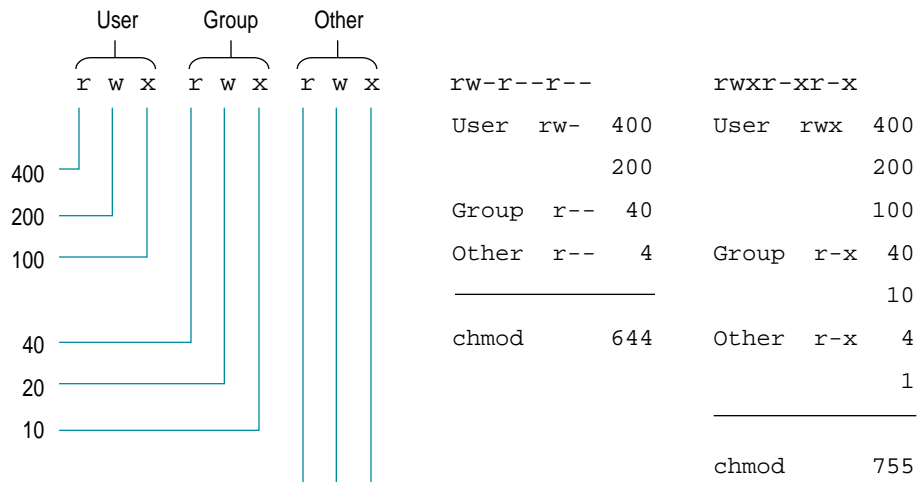


Figure 8-2 Numeric values for `chmod`

As Figure 8-2 shows, by adding up the values associated with each permission (`r`, `w`, `x`), you arrive at the numeric term to use with the `chmod` command.

The format for invoking `chmod` with numeric terms is as follows:

```
chmod permission filename
```

In this command line, *permission* is the calculated numerical representation.

For example, to assign a file read, write, and execute permission for the owner and to deny any access by group users or other users, you would use the following command:

```
chmod 700 filename
```

The permissions of the file are then as follows:

```
-rwx-----
```

Similarly, if you use the command

```
chmod 754 filename
```

the permissions of the file become

```
-rwxr-xr--
```

Setting command permissions using setuid and setgid bits

It is possible to set up commands *to act as if* they were being invoked by a specified user or by a member of a specified group. The mechanism for this is the use of a flag, or bit setting, called *set user identification (setuid)* or *set group identification (setgid)*. The setuid permission on a command allows the command to run with the permissions of its owner (the owner of the command that is being executed) rather than the normal behavior in which the command runs with the permissions of the person running the command. The setgid permission functions similarly but allows a command to run with the permissions of the group affiliation of the command.

For example, the `passwd` command, which a user runs to change his or her password, is a setuid program, `/bin/passwd`. When invoked, `/bin/passwd` takes on the identity of the owner of the `passwd` program, in this case `root`, for the time needed to modify the file `/etc/passwd`.

Of the two, commands using setgid permission tend to be safer, since group membership typically confers less power. Both should be treated with care, however.

It is possible to set up a program that can be run with setuid permission only by a selected set of users. You do this by putting the users into the same group to which the program belongs and denying execute permission to all other users. Then, only group members can run the program, performing the action as if they were the owner of the executable file.

You also use `chmod` to turn the `setuid` bit or `setgid` bit for a file on or off. The last section described using `chmod` with three numeric terms, such as `755`. The `setuid` or `setgid` bit is a fourth optional digit when using `chmod` with numeric terms. When using symbolic terms, use the corresponding characters shown in the following table.

Setting	Numeric term	Symbolic term
Set sticky bit (not used by A/UX)	1	t
Set GID bit on execution	2	s
Set UID bit on execution	4	s

Note: `Setuid` and `setgid` bits are applicable only with user or group permissions, not with permissions governing other users. Also, neither `setuid` nor `setgid` modes are useful when applied to directories or nonexecutable files.

Turning on the `setuid` or `setgid` bit is useful with very specific and restricted files—for example, the `/bin/passwd` program. The `chmod` command that turns on the `setgid` bit on a file with read-only permissions for all (mode `444`) is as follows:

```
chmod 2444 filename
```

The command to turn on the `setuid` bit on a file with read, write, and execute permissions for the owner, read and execute permissions for the group, and no permissions for all others (mode `750`) is as follows:

```
chmod 4750 filename
```

The permissions field in the output of the `ls -l` command in the first case is as follows:

```
-r--r-sr--
```

In this output, the `s` in the group execution field represents the `setgid` bit.

The permissions field in the output of the `ls -l` command in the second case is as follows:

```
-rwsr-x---
```

In this output, the `s` in the owner execution field represents the setuid bit.

You can combine the setting of the setuid bit and the setgid bit, as you can with all other numeric terms, so that

```
chmod 6755 filename
```

results in

```
-rwsr-sr-x
```

The sticky bit

In systems that load an entire file into physical memory, data is swapped in and out of memory as needed. These systems are called *swapping systems*. Paging systems such as A/UX, however, load a page (4K) of the requested data instead of a whole file at a time. This speeds data retrieval.

In swapping systems, the sticky bit indicates that the file should remain in main memory once it has been loaded in; this can shorten initialization time for frequently used programs at the cost of tying up a portion of main memory indefinitely. Because A/UX is a paging system, however, the sticky bit has no effect.

Setting default file permissions with `umask`

The `umask` command defines the default permissions for each file created by a user. You can run this command for all users in the `/etc/profile` or `/etc/cshrc` file, or each user can run it individually in his or her `.profile` or `.login` file. See “How A/UX Establishes the Environment” earlier in this chapter.

The `umask` command, like `chmod`, uses a numeric value of three octal numbers to alter permissions. The value of each specified digit of `umask` is *subtracted* from the corresponding digit specified in a completely open (777) file.

For example, to ensure that all files created by a user have the permissions

```
-rwxr-x---
```

you must set the `umask` value for that user as

```
umask 027
```

Thus, when 027 is subtracted from 777, the files permissions are 750 (-rwxr-x---). The default `umask` in the A/UX standard startup files is 027 for regular users.

The notation

```
umask 27
```

is shorthand for

```
umask 027
```

That is, leading 0's can be eliminated from the notation.

Note that changing a user's `umask` does not affect the permissions on existing files. It affects permissions only on files that are created later.

9

Troubleshooting

This chapter lists some common problems that A/UX users may experience, along with actions that you can take to identify and correct them. Problems are grouped as follows:

- recovering from a system crash
- problems that occur during startup
- user and group problems
- error messages from the `lpr` system
- file-system problems
- kernel problems
- network problems—hardware and software

Error messages from the command `fsck` are discussed in Chapter 10, “Checking Your Data Consistency.”

Troubleshooting guidelines

Here are some basic guidelines for troubleshooting problems:

- When you are troubleshooting a problem, always look for specific symptoms. Find out who the users are, what computers they are working on, what they were doing when the problem occurred, and anything else relevant or unusual. Find out if other users or computers are having similar problems. If only one computer has a problem, you can safely assume that the problem is confined to that computer on either the hardware or the software level. If several users or computers have a similar problem, you can raise the level of troubleshooting to revision levels of software or hardware that the systems have in common.

If error messages appear, read them carefully and write them down in your system log book. Once you have solved the problem, write the solution next to the error message, in case it happens again.

- Check for the simplest kind of failure first. Consider more complex problems after you have eliminated the simple causes of failure.
- Begin by trying the solution that will have the smallest impact on the resources of the computer or the network. For example, a problem may have two solutions: running a program, or restarting the computer. Restarting the computer is often the least desirable solution if the computer is still in use. If you have to restart a computer, be aware of the services it may provide to other computers and warn users that the computer is being restarted.

Recovering from a system crash

Many times a system will freeze—that is, become unresponsive—or you will temporarily lose access to it, without the system actually crashing. Before you restart the system, be sure that you have waited a while to see if the system gives you control again and that you have written down any error messages from the System Console window.

Error messages that begin with the word `panic` mean that the system cannot recover from whatever has happened and you will have to restart. If the system will not fully start up, try to use the Messages button in the startup windows to watch the startup process carefully, and document where the restart process fails. Once you have this information, refer to the next section for troubleshooting advice.

Supplied with this information, your Apple service provider will be able to help you solve the problem more efficiently.

If all else fails, you may have to reinstall your system software from the system software installer CD-ROM and user data files from your backups.

Trouble while starting up A/UX

Starting up A/UX in verbose mode may help you pinpoint the problem area. When verbose mode is enabled, messages that ordinarily are not visible are displayed on the System Console. To enable verbose mode, restart A/UX, and when A/UX Startup presents the first progress window, click the Cancel button. You are now running the A/UX Startup application shell under the Macintosh Operating System.

Choose Booting from the Preferences menu. Locate the AutoLaunch Command text box in the lower-right corner of the Booting dialog box, and add `-v` to the `launch` command. Now restart A/UX.

In a few moments, A/UX displays the output of the scripts that are run as A/UX comes up. The output may help you troubleshoot a problem.

When starting the machine, you press the POWER ON key, but the machine doesn't start.

- Check the power cord to make sure that both ends are tightly plugged into the correct socket. (One end plugs into the power supply at the back of the system and the other into an electrical outlet or power strip.)
- Make sure that the keyboard is plugged into the appropriate port with the appropriate cable.
- The electricity may not be functioning in the outlet; try one that you know works. If you are using a power strip, check to be sure that it is turned on.

A Macintosh icon with an unhappy face appears on the screen, accompanied by a chiming sound.

Turn off your server. Start up from a floppy disk containing system software to rule out any hardware problems. If the system starts up from the floppy disk, reinstall A/UX 3.0.1.

If the system fails to start up from the floppy disk, contact your authorized Apple service provider.

When you start the computer, a floppy disk icon with a blinking question mark appears in the middle of the screen.

- Restart the computer, using the programmer's switch if it is installed. The computer may not have recognized all your disks during startup.
- Examine the SCSI cables for proper configuration.
- Make sure that the SCSI chain has been terminated properly.
- Verify that each SCSI device has a SCSI ID (0 to 6) unique to that SCSI bus (1 to 4).
- Verify that the startup disk has not been hidden by the addition of a new disk having the same SCSI ID, or by an existing disk having its SCSI ID changed. See *Setting Up and Managing Your Server* for more information on hidden disks.
- System software may not be installed. Install, or reinstall, if necessary.
- The internal or external hard disk, or both, may have crashed, causing boot blocks to be damaged. Reinstall system software.
- A system file may be corrupt. Reinstall system software.
- After trying all the other suggestions, contact your authorized Apple service provider.

A/UX is frozen at the Login dialog box, and the keyboard does not respond.

- Make sure that the keyboard is plugged into the computer with the appropriate cable.
- Restart the computer and cancel the startup process, so that you are in A/UX Startup. If you have recently changed the `/etc/inittab` file, use the `cat` command to display the `/etc/inittab` file, and double-check your work.

You have recently installed a new system extension (INIT), and now the system will not start up.

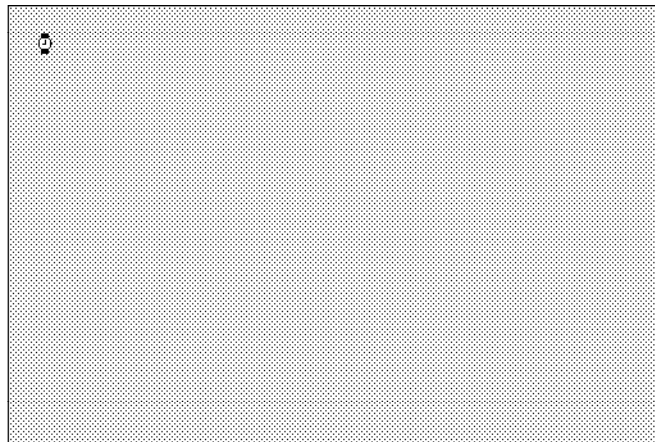
To have the system ignore the system extensions installed in the Extensions folder, you must press a key sequence while the system is starting up. Follow these steps:

1. Watch for the screen saying that the system is starting background (UNIX) processes, shown below.



Next, the screen will go blank.

2. Look for a watch cursor in the upper-left corner of the screen.



3. When you see the watch cursor, hold down the SHIFT key until the desktop appears.
4. Once you have started up, remove the new problem system extensions from the Extensions folder.
5. Try reinstalling the system extensions, one at a time, until you have identified all incompatible system extensions.

Accessing the A/UX file system from A/UX Startup

This procedure is very useful for fixing a multitude of problems, from creating a new `/etc/passwd` file to fixing entries in the `/etc/inittab` file.

Remember that there is a limited set of commands available within A/UX Startup; for instance, you cannot use wildcards, and all editing must be done with the `ed` editor.

1 Enter the A/UX Startup command-line interface.

Double-click the A/UX Startup icon and press `COMMAND-PERIOD`. Enter a password, if needed, to display an A/UX Startup window.

2 Find the SCSI ID number of the disk you want to access.

Most likely this will be the disk containing the root file system of A/UX. To find out where the root file system resides, look at the Root Directory text box in the General dialog box. The General dialog box can be accessed through the A/UX Startup Preferences menu. For more information, see Chapter 3, “System Startup.”

3 Use the `chroot` command to get access to that SCSI device.

If you have not changed the configuration of A/UX from the standard configuration in A/UX Startup, you do not need to use the `chroot` command; you will be in the root directory of A/UX when you enter A/UX Startup. If you want to access the root directory of another disk, use this command:

```
chroot (ID, 0, 0)
```

In this command, *ID* is the SCSI ID of the disk you want to access. Since the format of the `chroot` command does not allow for a SCSI bus number, you cannot access hidden devices with this command. (See *Setting Up and Managing Your Server* for more information on the new SCSI naming scheme and hidden devices.)

4 Use the `fsck` command on the file systems you may alter.

Never edit files on a file system without assuring the consistency of that file system by first running the `fsck` command.

5 Edit the files you wish; save the changes.

Because there is no full-screen text editor available in A/UX Startup, you need to use `ed`—a line editor. If you have accidentally destroyed an important file, such as `/etc/passwd` or `/etc/inittab`, you need to rebuild only enough of the file to get you to single-user mode, where you can replace the file with a backup. It is even possible to use the `echo` command to do this, by redirecting the output of `echo` into a file.

6 Restart or continue the A/UX startup process.

This action depends on what you have changed in A/UX Startup. If you are not sure, go ahead and restart the system. If you use the command `launch -v`, the system gives more information about what is happening during the startup process, and prompts you for any required information the system may require.

Reinitializing a disk with bad sectors

One situation in which you may need to create an A/UX file system is when you need to correct disk errors created by bad sectors on a disk. Apple HD SC Setup can be used to map bad sectors by reinitializing the disk. (If HD SC Setup does not work on your drive, use the partitioning utility provided by the manufacturer.) The following list presents overall steps you take to reinitialize an error-prone disk.

1 Run the file-system consistency check, `fsck`.

Normally, `fsck` can fix most data-storage inconsistencies on the disk, but you may have to run it a few times before it no longer reports errors. Procedures are provided in Chapter 10, “Checking Your Data Consistency.” If errors remain after several attempts to run `fsck`, proceed to step 2.

2 Back up your entire system.

The utilities `cpio` and `dump.bsd` have the ability to resynchronize themselves after encountering a hard disk error, and thus are preferable to use in this situation.

3 Test the disk using HD SC Setup.

This tests the drive for media errors. If you are still experiencing disk errors, you may have no choice but to reinitialize the disk. Proceed to step 4.

4 Reinitialize and partition the disk.

Follow the procedures given in Chapter 5, “Expanding Your System,” in *Setting Up and Managing Your Server*.

5 Restore the original contents of each of the file systems, using the most recent backups available.

Screen locks, power failures, and emergencies

If your system freezes and you cannot enter commands, you can try the following techniques to fix the problem:

- If you have modified `/etc/inittab` to start `getty` on another terminal, you can try to log in as `root` from another terminal.
- If you are connected to a network, attempt to log in over it.
- If you have MacsBug installed, and one of your Macintosh applications freezes, you may be able to use MacsBug troubleshooting techniques to stop the program. MacsBug is a debugging program that can be used with A/UX. Press `COMMAND-CONTROL-I` to enter MacsBug. Two useful commands you can then enter are
 - `es`, which terminates the currently running (or halted) program
 - `rb`, which terminates the Finder
- Pressing `COMMAND-CONTROL-E` terminates all active Macintosh applications, if possible, including the current Finder environment, and logs you out.
- Sometimes the screen unlocks on its own because of timeouts. Wait a few minutes (especially if you are using TCP/IP networking).

As a last resort, press the power switch on the rear of the computer or reset the system by using the Macintosh programmer’s switch.

WARNING Turning off the system power or using the programmer’s switch should be your last resort, because this can cause problems with the file systems and can either damage or lose data stored on the hard disk.

In a normal shutdown process (sometimes called a *graceful shutdown*), any data in the disk write buffer is written to the disk and all disk file information is saved before the file systems are unmounted and the power is turned off. In an emergency, or if the power goes off suddenly, an “ungraceful” shutdown is your only choice.

File-system inconsistencies are detected and repaired, when possible, by the file-system check command, `fsck`. This program runs automatically when the system starts up, and if it detects problems, it displays a message and asks whether to repair the problem:

```
The file system at /mnt needs repairs. Repair
automatically?
```

The best action is always to choose automatic repair. If you wish to use `fsck` manually, you can run the `fsck` command while in the A/UX Startup program and see exactly where the problems are occurring. Chapter 10, “Checking Your Data Consistency,” provides additional information about `fsck`.

Troubleshooting user and group administration

Most user administration problems can be traced to ownership and group membership questions or to erroneous entries in the `/etc/passwd` and `/etc/group` files. See Chapter 8, “User and Group Administration for A/UX Users,” for more information on this set of problems.

The message that the system displays when a problem arises is given first, followed by a description that suggests the cause and an action to take.

Invalid group id [group ID number]. Please contact the system administrator.

The group ID is not defined in the `/etc/group` file. Use a group that is defined in the `/etc/group` file.

Invalid user id [user ID number]. Please contact the system administrator.

The user ID is not defined in the `/etc/passwd` file. Use a valid user ID as defined in the third field of an `/etc/passwd` entry for the user.

Sorry, your password is incorrect. Please reenter it.

The password the user typed is different from what is recorded in the `/etc/passwd` file. Type the password again.

Sorry, that user name is unknown. Please retype the name or contact the system administrator.

The Name text box in the Login dialog box has a name that isn't listed in `/etc/passwd`. Either retype the name in the box or check `/etc/passwd` for the correct name.

The *[kind of session]* session startup program, *[name of startup program]*, does not exist. A console emulator session will be started instead.

The session type `/mac/bin/mac32`, or the chosen session type, is missing. Locate the appropriate session type or recover it from backups.

You don't have permission to execute the *[kind of session]* session startup program, *[name of startup program]*. A console emulator session will be started instead.

The session type `/mac/bin/mac32`, or the chosen session type, is not executable by the user. Edit the user's entry in `/etc/passwd` and change the name of the startup program, or change the permissions on the startup program the user is accessing.

You don't have permission to execute your default shell program, *[name of shell program]*. Please contact the system administrator.

The user doesn't have permission to execute the default shell program: This could happen if the system administrator made a directory in which the name differed from that in the `/etc/passwd` file. Edit the user's entry in `/etc/passwd` and change the name of the shell program, or change the permissions on the shell program the user is accessing.

Your default shell program, *[name of default shell program]*, does not exist. `/bin/sh` will be used instead. (A console emulator session is started.)

The default shell for `root`, as listed in `/etc/passwd`, doesn't exist. Edit `/etc/passwd` and give `root` a valid shell.

Your default shell program, *[name of default shell program]*, does not exist. Please contact the system administrator.

When this error occurs, the Login dialog box reappears. The user's default shell program as listed in `/etc/passwd` can't be found. Edit `/etc/passwd` and give the user a valid shell.

Your home directory, [name of home directory], is inaccessible. Perhaps that directory is on a file system which is not mounted. Please contact the system administrator.

The user's home directory (as listed in `/etc/passwd`) can't be found. Another possibility is that the system administrator made a directory in which the name differed from that in the `/etc/passwd` file. Check that the file system that contains the home directory of the user is mounted.

Your shell program, [name of shell program], is not a standard shell; thus, the session type will be Console Emulator.

There are three standard shells—`/bin/sh`, `/bin/csh`, and `/bin/ksh`. If the user's entry in `/etc/passwd` lists a different shell, this message is displayed in an alert box whenever the user chooses the Every Session or This Session Only button in the Change Session Type dialog box. To define the shell program as a standard one used by your system, add it to `/etc/shells`.

Errors while changing your password

Any of the following messages mean that password requirements are not met. These messages are displayed when the user attempts to click OK in the Change Password dialog box.

Your password must be at least six characters long.

Type in another password that contains more than five characters.

Your password must contain at least two alphabetic characters and one numeric or punctuation character.

Type in a password that has at least two letters of the alphabet and at least one character that is not a letter of the alphabet.

Your password cannot be a circular shift of your login name.

This means that if your login name is `mary`, you cannot have your password be `mary`, `arym`, `ryma`, or `ymar`.

Your new password must differ from your old one by at least three characters.

When you are changing passwords, the new password you choose must be different from the old password by three characters or more.

This password can be changed only by the superuser.

User accounts can be configured such that only the user logged in as `root` can change the password. Contact such a user and ask to have your password changed.

Sorry, your account has password aging restrictions. It has not been long enough since your password was last changed.

User accounts can be configured so that you can change your password only after an interval of time. Contact a user of the `root` account and ask to have your password changed.

This doesn't match your original entry. Please try again.

If you retype your password in the confirmation dialog box incorrectly, this message is displayed.

Another user is modifying the password file. Please try again later.

Only one user can change the password file at a time. Someone else may be editing it with `vipw` or the `passwd` command.

Troubleshooting the `lpr` system

This section explains possible solutions to `lpr` problems and the `lpr` system error messages. Note that the name *printer* refers to the name of the printer in the `/etc/printcap` database.

If files sent to the printer have not printed

- Make sure that the printer is not out of paper.
- Make sure that the cable from the LaserWriter to the Apple Workgroup Server 95 is plugged into the right ports.
- If using the `lpr` spooler, run `lpc status` to verify that the printer is accepting requests.
- If using `lpr` and the default printer destination, make sure a printer has been chosen with the Chooser.

lpr error messages

```
lpr:printer:unknown printer
```

The printer was not found in the `/etc/printcap` file. Verify that an entry for *printer* in the `/etc/printcap` file is present and correct.

```
lpr:printer:jobs queued, but cannot start daemon
```

The `lpd` process on the local machine has quit or is frozen. Check the local socket `/dev/printer.socket` with the `ls` command to make sure that it still exists. (If it doesn't, no `lpd` process will be running.) Logged in as `root`, you can enter the following command to restart:

```
/usr/lib/lpd
```

You can also check the state of the master printer daemon with this command:

```
ps -p `cat /usr/spool/lpd.lock`
```

Another possibility is that the `lpr` program is not setuid to `root` and setgid to the group `daemon`. Check to be sure that the output of `ls -l /usr/ucb/lpr` looks like this:

```
-rwsr-sr-x links root, daemon size date /usr/ucb/lpr
```

The values of *links*, *size*, and *date* are defined in `ls(1)`.

```
lpr:printer:printer queue is disabled
```

This message indicates that the queue was turned off by the system administrator with this command:

```
lpc disable printer
```

This stops `lpr` from putting files in the queue. The system administrator, logged in as `root`, can turn the printer back on by using this command:

```
lpc enable printer
```

lpq error messages

waiting for *printer* to become ready

The daemon cannot open the printer, most often because the printer is off line. The message may also be displayed if the printer is out of paper, or the paper is jammed. The actual reason depends on the meaning of error codes returned by the device driver. Check to see that the printer is ready for printing. If the printer is OK, try the `lpc restart` command. If all else fails, restart the system.

printer is ready and printing

The `lpq` program has checked to see if a daemon process exists for the printer and has printed the file status located in the spooling directory. If this message appears, but the daemon is not working, the user logged in as `root` can use `lpc` to stop the current daemon and start a new one.

waiting for *host* to come up

This message implies that a daemon is trying to connect to the remote computer named *host* to send the files in the local queue. If the remote computer is up, `lpd` on the remote computer has probably terminated or has frozen and should be restarted.

sending to *host*

This message means that the files should be in the process of being transferred to the remote computer named *host*. If they are not, the local daemon should be terminated and restarted with `lpc`.

Warning: *printer* is down

The selected printer has been marked as unavailable with the `lpc down` command. After figuring out why the printer was brought down, use the `lpc up` command to bring the printer back up.

Warning: no daemon present

The `lpd` process overseeing the spooling queue, as specified in the `lock` file in that directory, does not exist. This normally occurs only when the daemon has unexpectedly quit. The error log file for the printer and the System Console window should be checked for a diagnostic message from the former process. To restart `lpd`, enter this command:

```
lpc restart printer
```

`lprm` error message

```
lprm:printer:cannot restart printer daemon
```

This message is the same as the `lpr` message that the daemon cannot be started. This normally occurs only when the daemon has quit unexpectedly. Check the error log file for the printer and the System Console window for a diagnostic message.

`lpd` error messages

The `lpd` program logs every message to the System Console window, by default. Most messages logged by the `lpd` program relate to files that cannot be opened and usually mean that the `printcap` file or the permissions of the files are incorrect. Files may also be inaccessible if users bypass the `lpr` program when printing.

`lpc` error messages

```
couldn't start printer
```

This message is the same as the `lpr` message that the daemon cannot be started. This normally occurs only when the daemon has quit unexpectedly. Check the error log file for the printer and the System Console window for a diagnostic message.

```
cannot examine spool directory
```

Error messages that begin with “cannot” usually result from incorrect ownership or protection mode of the lock file, spooling directory, or `lpc` program.

File-system problems

During the launching of A/UX, `fsck` locates a problem with the file system. You are asked to click the Repair button.

- Let `fsck` automatically repair the file system by clicking Repair.
- Another method is to run `fsck` from A/UX Startup. Restart your system. Click Cancel to enter the A/UX Startup shell window. Enter one of the following two commands, depending on which SCSI bus your drive is on.

```
fsck -p /dev/dsk/cxd0s0
```

In this command, *x* stands for the SCSI number of the device on the internal main logic board.

```
fsck -p /dev/dsk/cb0xd0s0
```

In this command, *b* stands for the number of the SCSI bus of your device (1–4), and *x* stands for the SCSI address number of your device (0–6).

- See Chapter 10, “Checking Your Data Consistency,” for more information on repairing your file system.
- If these suggestions don’t work, contact your local authorized Apple service provider for assistance.

The error message `fserr: filesystem full` appears on the screen periodically.

The file system has run out of space or inodes. Enter `df`, which displays the number of blocks and inodes available for use on the current file system. To free space and inodes, make backups of old files to be removed, either on tape or on floppy disks, and then remove them from the full file system. If you log in as `root`, the messages may stop while you clear out the files. (See “Overview of the A/UX File System Interface” in Chapter 10, “Checking Your Data Consistency,” for a discussion of blocks and inodes.)

While you try to partition a disk for A/UX with Apple HD SC Setup, the drive cannot be found.

- Make sure that the external hard disk is turned on.
- Make sure that the SCSI cables are properly configured.
- Make sure that the SCSI chain has been terminated properly.

- Some non-Apple hard disks cannot be read by Apple HD SC Setup. If the disk is not an Apple product, contact your service provider or manufacturer to get the right partitioning software.
- Two disks on the same SCSI bus with the same SCSI ID may be turned on. Shut the computer down and turn one of the drives off. Change the SCSI ID to make it unique for the bus it is on. Turn on the drive and the computer.
- The drive may be damaged. Contact your authorized Apple service provider.

While trying to unmount a mounted file system, you encounter the error message
/filesystem: Device busy.

The file system *filesystem* is currently in use due to a number of possible reasons:

- A text file or application that is in the file system is in use, perhaps through another CommandShell window.
- Someone else on the network may be accessing that directory. Use the `who` command to see if someone else is logged in who might be using the file system.
- AppleShare may be exporting folders contained within the file system. Check to see which folders AppleShare is exporting.

When you have corrected the problem, change to the root directory by entering

```
cd /
```

Enter the `umount` command again.

While you are creating a file system on a disk that has been initialized and partitioned with A/UX, an error message is displayed indicating that the block limit is too large to fit on that partition.

- Check the slice number to be sure that it coincides with the partition you gave it while using Apple HD SC Setup.
- Make sure that you used the correct SCSI ID number.
- Use HD SC Setup to determine the size of the partition.

Kernel parameter problems

Typically these problems arise after rebuilding a new kernel.

The error message `m_expand returning 0` appears on the screen periodically.

Increase the number of `NMBUFS` with the `kconfig` command. `NMBUFS` allocates buffers for networking; when you install NFS, the number should be increased. Remember that these changes do not take effect until the kernel has been restarted. See `NMBUFS` in `kconfig(1M)` for details.

The error message `file: table is full` appears on the screen periodically.

The system file table is full and needs to be enlarged. The `kconfig` command allows the `NFILE` parameter to enlarge the table. When you increase the `NFILE` parameter, the `NINODE` parameter should be equal to or greater than the `NFILE` parameter. (They are usually kept at the same number.) The total memory configuration of your system should determine the size of your `NFILE` and `NINODE` parameters. See `kconfig(1M)` for details.

The error message `proc: table is full` appears on the screen periodically.

The system has attempted to increase the total number of system processes beyond the default number set in the kernel. The `kconfig` command allows the `NPROC` parameter to reflect a higher number. Increase the `NPROC` parameter in increments of 25 until the message no longer appears. You must restart each time you change the kernel with the `kconfig` command. See `kconfig(1M)` for details.

Other issues

Some of the icons in your folders are blank, or you get the “Application not found” error message with documents for which you do have the appropriate application.

Rebuild the desktop by first removing the file `DB Desktop` from your System Folder. (Remove the file from your personal System Folder if you have one, or from the global System Folder if you do not.) Then log out and log back in. The system rebuilds your desktop automatically.

Error messages are scrolling past on your screen so fast you cannot read them.

If you are in CommandShell and running an A/UX utility, you can press CONTROL-S to stop the output to the screen, and CONTROL-Q to start the output scrolling again. Because of a full input buffer, you may not receive an immediate response to CONTROL-S.

System messages scrolling in the System Console window are not affected by this procedure.

While you are using the `tar` or `cpio` command with a floppy disk, the error message `cannot open /dev/floppy0` is displayed.

The drive that has the floppy disk could be `/dev/floppy1`, or else the disk is write-protected.

When you are using `chgrp` and `chown`, the error message `filename: Not owner` appears.

You do not have the appropriate permissions to change the owner or group of that file. Enter `su` to log in as `root`, and run the command again.

While you are using `tar` or `cpio`, an error message appears indicating that the utility cannot open `/dev/rmt/tcX` where `X` is the SCSI ID of the tape device.

- The tape is write-protected, or you don't have an MRS-quality tape in the DDS-DC DAT drive.
- The device file you selected was assigned an incorrect SCSI ID number. Reselect it with the correct device number.
- The tape drive may be hidden by the SCSI ID you have chosen. Since tape devices don't use the new SCSI naming scheme, you cannot have any devices with the same SCSI ID number in an earlier searched bus. It is preferable to put tape devices on SCSI bus 1 or 2, which are connected to the main logic board. See *Setting Up and Managing Your Server* for additional details on the SCSI naming scheme and bus search order.
- The `tc` driver may have been removed from the kernel. Verify that the `tc` driver is in the kernel by running the `module_dump /unix` command. Look for the `tc` driver in the list. If it isn't there, run `newconfig` to configure the kernel with the tape driver, and then restart the computer.
- When using `tar`, you failed to use `-f /dev/rmt/tcX`, where `X` is the SCSI number.

Troubleshooting network problems

This section describes problems you are most likely to run across when the network fails.

Debugging the network hardware

If you suspect a hardware problem, always check the network connections first. With a medium such as Ethernet, a loose cable tap or misplaced transceiver cable can cause severely deteriorated service. The `-i` option of the `netstat` command can help you track down hardware malfunctions. For example, if you run `netstat -i` on a computer to which packets are being sent, you may see that the input packet count does not increase, which indicates that the computer is not seeing the packets. Such a scenario suggests a hardware failure of some kind.

Checking transceivers

When a computer has been turned on for a while, its transceiver should feel slightly warm to the touch. If a transceiver is cold, it probably isn't receiving power. A transceiver that is cold indicates one of several problems:

- a loose connection on either end of the transceiver cable
- a loose connection between the internal Ethernet cable and the Ethernet card
- a faulty cable, transceiver (less likely), or card (even less likely)

To remedy the situation, check all Ethernet coaxial and transceiver cable connections. Note that hardware devices may be physically but not electrically connected. In this case, disconnect and reconnect them.

Checking terminators

If all the computers on the network suddenly experience network problems, a network-wide problem may be related to the integrity of the Ethernet cable upon which the network depends. A cable terminator may be loose.

Unscrew one of the terminators and use an ohm meter to test resistance across the coaxial connector where you just unscrewed the terminator. (Use the pin “inside” the N-connector for signal and the computer’s housing for ground.) You should measure about 50 ohms. If you get something other than 50 ohms, your cable may be damaged. This check is particularly pertinent if you use a clamp-on (“vampire clamp”) transceiver; such transceivers tend to short-circuit the Ethernet coaxial cable.

Remove one of the terminators and try operating the network. An error message such as this should be displayed on every computer:

```
Ethernet transmission error
```

If a computer does not display this message, it may not be connected correctly to its transceiver. The problem could be loose connections or a faulty connector, cable, transceiver, or Ethernet card.

Swapping hardware

Try swapping transceivers and transceiver cables. If spare Ethernet cards are available, try swapping cards. Even if there are only two computers on the network, learning whether the problem follows the exchanged items or stays with the computer is informative.

Analyzing TCP/IP network software problems

If network response is unusually slow, the problem could be due to either a heavy load on the network or CPU-intensive jobs running on your system. Several tools are available for monitoring both network traffic and computer load. The most useful tool for monitoring network traffic is `netstat`. The most useful tools for monitoring computer load are `lax` and `ps`.

Analyzing network traffic

The `netstat` command is useful for examining the contents of various TCP/IP data structures on the local computer, such as the number of incoming and outgoing packets and any associated errors. For example, to see statistics on active interfaces, enter this command:

```
netstat -i
```

In response, `netstat` displays a line of information per interface. Here is an example:

```
Name Mtu  Network  Address  Ipkts  Ierrs  Opkts  Oerrs  Coll
ao0  1500  128.8    hostname1 40506  0      10158  0      0
lo0  1536  loopback-n loop      1012   0      1012   0      0
```

The display shows the name of the interface, the maximum transmission units of the interface, the network number, the address, the number of input packets, input errors, output packets, output errors, and collisions. You can see the numbers change over time if you also provide an interval argument:

```
netstat -i 5
```

In response, `netstat` first displays a summary of information since the computer was last restarted and then displays the numbers accumulated since that last interval. To see if your computer is dropping packets, use `rcp` to copy several large files to your computer and watch the value in the `Ierrs` and `Oerrs` columns.

To see the contents of the routing tables, enter this command:

```
netstat -r
```

In response, `netstat` displays a line of information per route. Here is an example:

```
Routing tables
Destination  Gateway          Flags  Refs  Use
Interface
loop         loop             UH     0     0    lo0
90.183      hanz.apple.com  UG     0     0    ao0
90.175      yin.apple.com   U      15   2699  ao0
```

Each line in the display represents a destination host or network and a gateway to use in forwarding packets. The third column shows the state of the route (U for up, D for dynamic, G for gateway, H for host, and M for modified).

To see the status of sockets, enter this command:

```
netstat -a
```

In response, `netstat` displays information about each active socket, including the local and remote addresses, the size in bytes of the send and receive queues, and the associated protocol and its state.

To see statistics about the ICMP, IP, TCP, and UDP protocols, enter this command:

```
netstat -s
```

In response, `netstat` displays information that includes summaries for incomplete headers, packets, retransmissions, and checksums.

Analyzing routing errors

If you suspect a malfunction of the `routed` daemon, you can log its actions and all packet transfers. To enable `routed` logging, start up `routed` with a filename argument. Here is an example:

```
/etc/in.routed /etc/routerlog
```

Whenever a route is added, deleted, or modified, `routed` logs the action, as well as a history of the packets previously sent and received, in the log file. To enable full packet tracing, specify the `-t` option as shown here:

```
/etc/in.routed -t /etc/routerlog
```

Note: On a busy network, the `-t` option generates almost constant output. See `routed(1M)` for more detailed information about other options.

Responding to common TCP/IP error messages

This section explains the most common error messages that can appear when users are trying to use the TCP/IP commands (`rarp`, `rlogin`, and `remsh`) or the Internet commands (`ftp` and `telnet`).

```
Connection refused
```

The remote computer is up, but is not running the necessary daemons to handle the remote request.

```
Connection timed out
```

This message usually appears after a long delay (about two minutes) following a `remsh` or `rlogin` command. The remote host is down or is so busy that it could not respond in time. See if the remote host is down or in single-user mode. This message can also appear if the remote computer has a small amount of physical memory or has a small number of network buffers. In the latter case, you may want to use `kconfig` to increase the number of memory buffers. See `NMBUFS` in `kconfig(1M)` for details.

Login incorrect

This message is displayed by `rlogin` in response to several conditions. If the user does not have an account on the remote computer, this message is a legitimate refusal to allow the user to log in. Another possibility is that the user typed the wrong password.

Try typing the password again. If optional packages such as NIS are installed, refer to the *A/UX Network System Administration* manual.

Permission denied

This message is displayed by `rcp` or `remsh` when the remote computer detects a permissions violation. Either you can modify the file `/etc/hosts.equiv` or the user can create a `$HOME/.rhosts` file on the remote computer.

`rcp:filename:` No such file or directory

The `rcp` command cannot find the file or directory to be copied.

`rhost:`Host name for your address unknown

The remote computer does not know the name and address of the local computer. Modify `/etc/hosts` on the remote computer.

`rcmd:`Unknown host "*hostname*"

This message appears when a user tries to run `rcp`, `rlogin`, or `remsh` to communicate with a computer that is not known to the local computer. Add the computer to the `/etc/hosts` file on the local computer.

hostname is an unknown host

This error message comes only from the `talk` command and indicates that the remote computer specified by *hostname* is not known to the local computer. Add the computer to the `/etc/hosts` file on the local system.

panic: out of mbufs

When this message appears, the system has denied a request to allocate memory for use as memory buffers. Restart A/UX and run `kconfig` to increase the number of memory buffers. See `kconfig(1M)` for details.

str_open: out of streams

The system is out of stream heads (used by AppleTalk). The Finder may also freeze when the system is out of stream heads. Stream heads may be exhausted when the system is making heavy use of Macintosh file sharing and TCP/IP networking. Run `kconfig` to increase `NSTREAM`. If you increase `NSTREAM`, you should also increase `NBLK1024`, which is the number of 1-kilobyte stream blocks available for use by a stream head. One-kilobyte blocks are the most frequently requested stream blocks. You may also want to increase `NBLK2048`, and `NBLK4096`, which are used less frequently than 1-kilobyte blocks, by a smaller amount. See `NMBUFS` in `kconfig(1M)` for details.

panic: kmem_alloc failed (N bytes)

When this message appears, the system denied a request to allocate a buffer from the static pool reserved for use by the networking subsystem. Restart A/UX and run `kconfig` to increase the size of this pool. Do this by increasing `MAXCORE`. A good size for a server is `0x60000`. See `NMBUFS` in `kconfig(1M)` for details.

10

Checking Your Data Consistency

A/UX automatically checks your file system for corruption each time the system starts up. This is done with a file-system check (`fsck`) command during the time A/UX Startup controls the system. This chapter discusses `fsck` and includes

- a discussion of how `fsck` works
 - a description of the structure of the UFS file system (the file system on the root partition; also known as *4.2 BSD*)
 - a description of `fsck` phases
 - a description of `fsck` error messages
-

Introduction to `fsck`

In many cases, `fsck` can fix damage to a file system and its data. Therefore, most problems can be solved by properly restarting the system and letting A/UX check your data.

Sometimes, however, `fsck` can report only cryptic messages about the damage that has been done. In these cases, a system administrator can partially repair the file system with the help of `fsck` and restore any unsalvageable files from backups.

If you are currently trying to correct errors with `fsck`, turn to the section “`fsck` Error Messages” later in this chapter. That section contains the suggested responses to various problems. Alternatively, you can consult the index of this guide to locate error messages.

The file-system check program `fsck` locates and resolves inconsistencies within a file system. It is part of the normal startup sequence, as described in Chapter 3, “System Startup.” In the standard A/UX distribution, `fsck` is run automatically on the root file system and file systems listed in `/etc/fstab`. If `fsck` detects errors during the automatic startup procedures, a dialog box appears asking if you want to repair the errors. If you do not click the Repair button, or if a message that says the file system cannot be repaired is displayed, you must run `fsck` through the command-line interface in single-user mode before you can bring the system to multi-user mode. The section “Using `fsck`,” later in this chapter, tells you how to do this.

IMPORTANT File-system problems do not go away with time; they only get worse. Run `fsck` whenever you suspect inconsistencies within the file system—for example, immediately following a power failure or a system crash.

Overview of the A/UX file-system interface

The A/UX operating system supports the Macintosh hierarchical file system and two UNIX file systems, the Berkeley UNIX File System (UFS) and the System V file system (SVFS). A/UX must start up from either a UFS or an SVFS file system. For performance reasons, UFS is the preferred file system and is the file system installed by the system software installer CD-ROM. Although most of this chapter discusses UFS, differences related to SVFS are noted. HFS file systems are not relevant to this chapter.

File systems are periodically updated with modifications made on the data in main memory. Sometimes this updating fails, usually because of a power failure or improper system shutdown; this may result in inconsistencies within the file system. In most cases you can resolve these inconsistencies by using the `fsck` program.

The `fsck` program checks the location of files on disk and uses redundancies and known parameters to resolve inconsistencies. A redundancy is information that the system maintains in more than one place, such as the size of each file and the number of blocks not currently in use. A known parameter is information about the file system that does not change, such as the number of characters per block or the number of blocks in a disk.

It is helpful to understand the organization of UFS and some of the commands that manipulate this organization before you begin to work with `fsck`. This section gives a brief overview of the relevant file and directory information.

Blocks and bytes

A file system is divided into units called *blocks*. A block on the disk is called a *physical block* and is a contiguous sequence of bytes (usually 512 bytes in length). To speed up the disk input/output (I/O) operations, file systems work with more than one physical block at a time; this entity is called a *logical block*. This feature allows the operating system to use more data per disk access. UFS supports the use of from 2 to 16 physical blocks per logical block (called 1K and 8K file systems, respectively).

Inodes

Inodes contain information about files, the most important part of which is a list of logical blocks where a file's contents are to be found. The inode does not point to a single location on the disk, but to several discrete locations (explained in the next section, "Direct and Indirect Blocks").

Unlike files and directory files, inodes are of a fixed size and reside in fixed locations on the disk. For this reason, inodes have inumbers instead of names. The inumber 30 points to the 30th inode in the inode area on the disk. Directory files are simply lists correlating inodes and filenames. To determine a file's inode from its name, you can use this command:

```
ls -i filename
```

To determine a file's name from its inode, you can use this command

```
ncheck -i inode /dev/rdisk/cb0xd0sz
```

In this command, *b0* is the SCSI bus number (if needed) and *x* is the SCSI ID of the hard disk. The value of *z* is the slice number associated with a particular disk partition.

Figure 10-1 illustrates the relationship between the directory `/users/demo`, a file named `letter`, the inumber and inode associated with `letter`, and the disk locations where the contents of `letter` are stored.

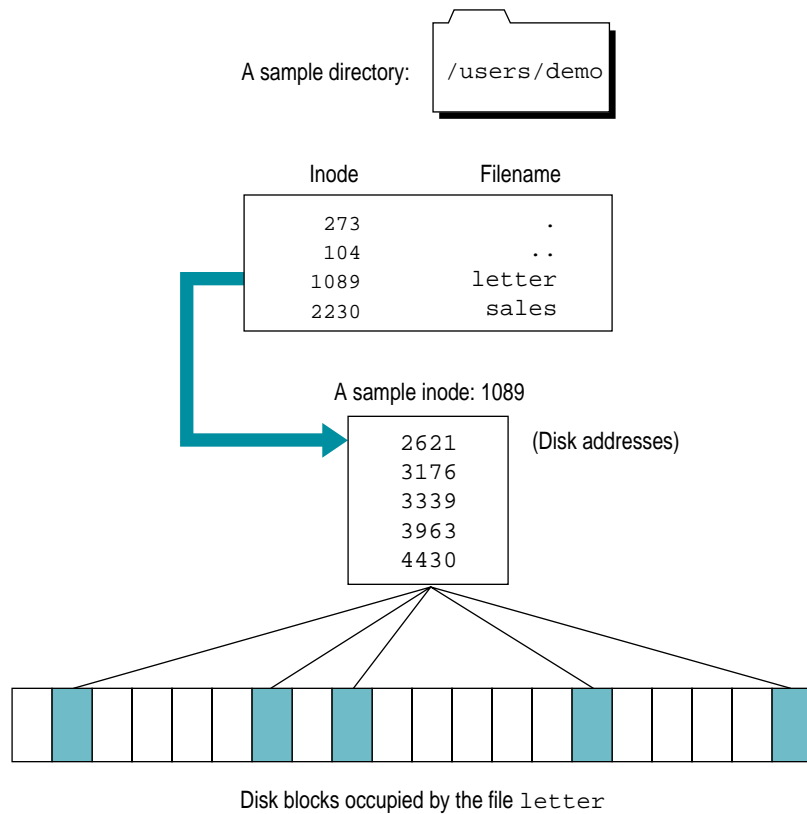


Figure 10-1 Number relationships

Direct and indirect blocks

UFS inodes contain 15 disk addresses. (SVFS inodes contain only 13.) The first 12 addresses point to the first 12 logical blocks of the file. These blocks are called the *direct data blocks*. If a file contains more than one logical block of data, it continues at the second address to which the inode points. If it contains more than two logical blocks of data, it continues at the third address, and so on, until the first 12 addresses have been used.

If a file has used up the direct data blocks, the 13th address given in the inode is then taken into consideration. The 13th disk address points to a block called an *indirect block*, which contains the addresses for the next 1024 logical blocks that the file can use. Figure 10-2 illustrates these connections for a 4K file system.

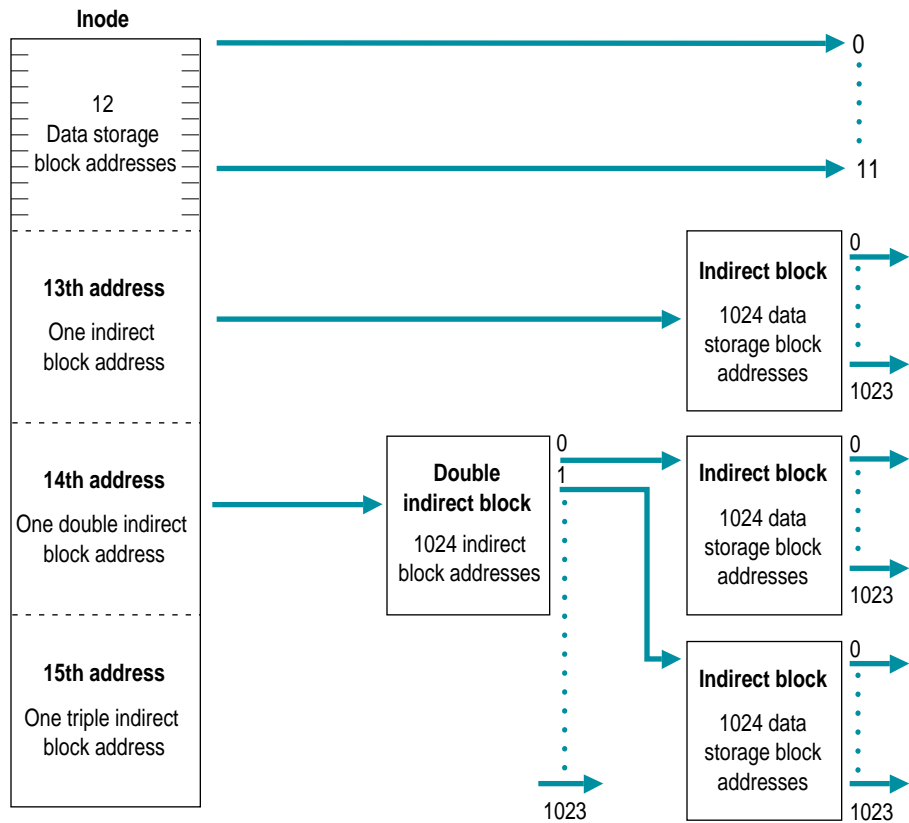


Figure 10-2 Indirect blocks on a 4K file system

If the file is larger than 1036 blocks (12 blocks for the first 12 addresses, 1024 for the 13th), it continues at the 14th address given in the inode. The 14th address points to a block called a *double indirect block*, which refers to up to 1024 indirect blocks that the file can use. This gives a total of 1,049,612 logical blocks of data: 12 from the first 12 addresses, plus 1024 from the indirect block, plus 1,048,576 from the double indirect block. For a 1K file system, a file that uses the double indirect block can hold a little over 1 GB of information. On an 8K file system, the same file can theoretically hold about 8.5 GB. However, because an inode's file size is represented in A/UX by a signed 32-bit quantity, a file can actually never get larger than 2 GB.

The 15th address in the inode points to a block known as a *triple indirect block*, which refers to up to 1024 double indirect blocks, each of which in turn refers to up to 1024 indirect blocks, and so on. In practice, the triple indirect block is not used in UFS file systems.

Additional information contained in inodes

Inodes contain more information than just the location of the data blocks that make up a file. They also contain

- permissions
- owner and group affiliation
- file size
- the time the file was last accessed (read)
- the time the file was last modified (written)
- the time the inode was last modified (written)

Inodes record three different time-related statistics about a file: access time, modification time, and inode modification time. Access time is the last time the file was read, and modification time is the last time the file was written to.

Inode modification time is sometimes referred to as “creation time.” This is a misnomer, because modifying, changing permissions, and changing ownership all update the inode modification time on a file.

You can use the `ls -l` command to see some of this additional information. See `ls(1)` for further information on the available options.

An example of inode and file access

Figure 10-3 illustrates the connection, through multiple inodes, between the root directory and a file called `/usr/tmp/junk` located several levels below the root directory. Initially, the operating system accesses the root directory to get the inode information for the `usr` directory. This inode gives it the disk location of the `usr` directory, which it then accesses to get the inode information of the `tmp` directory. With this inode information the operating system locates and accesses the `tmp` directory, which provides the inode of the file `junk`. The operating system then uses this inode to access the first logical block of the file itself.

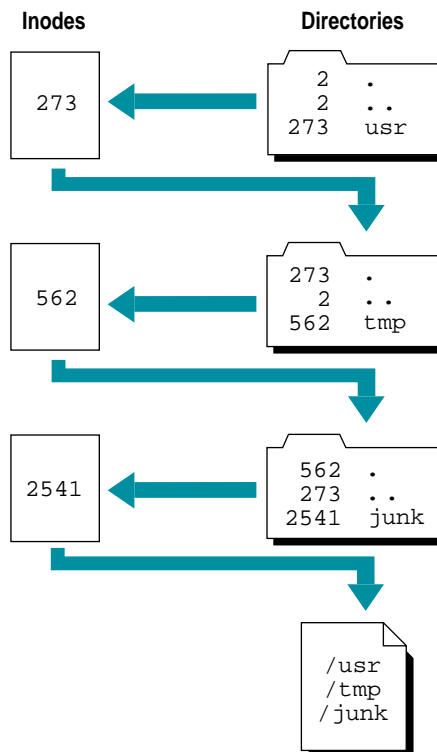


Figure 10-3 File-directory connection through inodes

This process illustrates that the operating system must read information from disk a relatively large number of times to locate a file for the first time. However, because the operating system maintains a cache of active and recently used inode information in an inode table, the next time the file is accessed the process is much quicker.

The superblock

Each UFS file system is described by its superblock, which is located at the beginning of the file system's disk partition with copies elsewhere on the disk. (A System V file system has only one copy of the superblock.) The superblock contains the following critical data about the file system:

- the size of the file system
- the size of the file system's logical blocks
- a "magic number" that identifies the file-system type
- a flag that indicates whether the file system has been mounted as read-only
- a flag that indicates whether the superblock has been modified
- a time stamp that shows when the superblock was last modified
- a flag that indicates whether the file system was unmounted cleanly

The UFS file system is subdivided into cylinder groups, each of which contains its own superblock. This superblock contains space for inodes, a copy of the primary superblock in case it is corrupted, and a bitmap indicating whether data blocks are available or in use.

Block I/O

It would be both risky and expensive to keep all data in main memory (also called *primary memory*, *in-core memory*, or *RAM*). Instead, most files are kept in secondary memory (for example, on a disk), and the system brings them into main memory as necessary. If you modify files, the system eventually writes the modified versions back into secondary memory for future use.

When a program reads data from or writes data to a disk or tape, the system extracts logical blocks and brings them into main memory. It would be impractical and even unsafe to bring data into main memory without imposing limits and some degree of organization on the amount of data transferred. It would also be highly inefficient for the CPU to do physical I/O operations whenever data is transferred from primary to secondary memory and vice versa. For that reason, the system maintains a list of buffers for each device. This list of buffers is said to constitute a buffer cache for all block-oriented I/O.

The kernel buffer cache

When a program asks the system to read data from a file, the system first searches the cache for the desired block.

If the block is found (for instance, when the system opens a file that is already open), the data is made available to the requesting process without a physical I/O read operation. If the data is not found in the cache, the buffer that has been unused for the greatest period of time is renamed, and data is transferred into it from the disk and made available.

When a process writes a file, the operation occurs in reverse order. A write request first writes data to the buffer cache. Data is written to disk only when the cache is full. Therefore, information about “bad write” operations refers generally to unusual bad writes to the buffer, and “bad disk write” operations are generally reported too late to prevent the file system from being corrupted.

The `/dev` directory

There are several types of files in a UFS file system: regular files, directories, special files (device files), sockets, symbolic links, and named pipes. In the beginning days of UNIX, only three types of files existed: regular files, directories, and devices. In this context, device files were special and were given the name *special files*. The addition of other file types makes the name no longer appropriate, but it is still used. In the A/UX books, the term *device file* is commonly used.

When you use the `ls -l` command to list your files, the system response looks something like this:

```
-rw-rw---- 1 groupname 13 Sep 25 11:28 file
drwxrwx--- 2 groupname 512 Sep 25 11:28 directory
```

For regular files, such as the first one listed in the example, the first character in the permissions field is a minus sign (-). In the case of directories, this character is always d. However, suppose that you list `/dev/rdisk/c0d0s0` and `/dev/dsk/c0d0s0` by giving this command:

```
ls -l /dev/rdisk/c0d0s0 /dev/dsk/c0d0s0
```

In response, the system displays these lines:

```
crw----- 2 bin          24, 0 Mar 25 1992 /dev/rdisk/c0d0s0
brw----- 2 bin          24, 0 Mar 25 1992 /dev/dsk/c0d0s0
```

The first character in the permissions field of `/dev/rdisk/c0d0s0` is c (indicating a character device), and the first character in the permissions field of `/dev/dsk/c0d0s0` is b (indicating a block device). Either a b or a c in this position indicates a device file.

Devices (and therefore device drivers and their corresponding device files) come in two forms, block and character (hence the b and c in the `ls -l` listing). These names refer to the method of I/O used with each type of device.

Block devices such as disks use the block I/O buffer cache mentioned previously and are thus written to, and read from, entire blocks at a time. Character devices such as terminals, line printers, and modems are written to, and read from, one character at a time, or use an alternative buffering scheme.

Each hard disk partition is associated with both a character and a block device driver and thus with two device files in the `/dev` directory. For this reason, you can access disk partitions in two ways. They're normally accessed as block devices through the directory hierarchy. But certain programs that access disks, such as `dump.bsd`, `dd`, and `fsck`, run faster when you access the disk as a character device. For example, `fsck` checks the device `/dev/rdisk/c0d0s0` faster than it checks the device `/dev/dsk/c0d0s0`.

Now suppose that while a file is printing, you list `/usr/spool/lp/FIFO` by giving this command:

```
ls -l /usr/spool/lp/FIFO
```

The `p` in the first field tells you that the file is a pipe:

```
prw----- 1 lp lp 0 Oct 21 1991 /usr/spool/lp/FIFO
```

The other two types of file have their own symbols: `l` (symbolic links) and `s` (sockets).

A device driver is a program that controls the actual physical I/O to the devices listed in the `/dev` directory. However, the device driver itself doesn't reside in the `/dev` directory; rather, it is compiled directly into the kernel.

There is a different device driver for each kind of device (disk drives, tape drives, scanners, and so on). The system uses the major number to access the correct device driver. The minor number is passed to the driver, which uses this argument to select the correct physical device, including specific attributes of the device. For example, the rewind and no-rewind devices associated with a tape drive device have the same major number (specifying a tape drive) but different minor numbers (specifying the different behaviors).

The files in the `/dev` directory are all special files that the system uses to select a device driver for performing physical I/O. These files are actually just names and inodes with no associated data on disk (and thus a size of 0 bytes). Instead of storing information about the number of bytes in a file, these inodes contain a major and minor number for each device file. The major and minor numbers are displayed in the output of the `ls -l` command. In the following example, the major number is 24 and the minor number is 0.

```
crw----- 2 bin 24, 0 Mar 25 1992 /dev/rdisk/c0d0s0
```

In summary, the system takes the following steps in response to requests to open device files (such as `fsck` may make):

- It looks in `/dev` directory for a file with the requested name.
- It gets the inumber associated with the filename.
- It finds the inode specified by the inumber.
- It gets the major number stored in the inode.

- It uses this number to select the appropriate device driver.
- It passes the minor number to the device driver.

The driver then uses the minor number to select the correct physical device (the proper partition, in the case of a hard disk device).

How `fsck` works

As you open, create, and modify files, the system keeps track of all pertinent information about them in two places. This file information—including block sizes, their inumbers, active and free inodes in the file system, and total number of active, used, and free blocks—is maintained and updated immediately in main memory. Periodically, different parts of this information are written to the disk. If the system crashes, the various file systems can become inconsistent. This inconsistency arises because some file information in main memory was written to disk before the problem, while other information was not.

The `fsck` program works by comparing one or more items of information to one or more items of equivalent information. For instance, it compares the number of free blocks available to the number of total blocks in the file system minus the number of blocks in use. If the two numbers are not equal, `fsck` generates an error message. This kind of error results from an inconsistent update or one that was performed out of order. To understand the problems `fsck` is designed to solve, it is helpful to understand these updates.

File-system updates

This section describes the various file-system updates the system performs every time you create, modify, or remove a file. There are five types of file-system updates: superblock, inode, indirect blocks, data block, and free list.

Superblock

A mounted file system's superblock is written to disk whenever the file system is unmounted or when a `sync` command is issued. The system periodically issues a `sync` system call to prevent the superblock on disk from getting too out of date. (See `sync(1M)` and `sync(2)`.)

The superblock of a file system is prone to inconsistency because every change to the blocks or inodes of the file system modifies the superblock.

Inode

An inode is written to disk when the file associated with it is closed. In fact, all *in-core blocks* (those pieces of a file that are in main memory) are also written to disk when a `sync` system call is issued. Thus the period of danger when inconsistencies can appear is reduced to that between `sync` calls. Typically, the system issues a `sync` call every 30 seconds.

Indirect blocks

Indirect blocks, as well as the first 12 blocks of a file, are written to disk whenever they have been modified or released by the operating system. More precisely, they are queued in the buffer cache for eventual writing. Physical I/O is deferred until the operating system needs the buffer or a `sync` call is issued.

Inconsistencies in an indirect block directly affect the inode that points to it.

Data block

A data block is written to disk whenever it has been modified. There are two types of data blocks: plain and directory. Plain data blocks contain the information stored in a file. Directory data blocks contain directory entries.

The `fsck` program does not attempt to check the validity of the contents of a plain data block. Instead, `fsck` checks each directory data block for inconsistencies involving the following attributes:

- directory inode numbers pointing to unallocated inodes
- directory inode numbers greater than the number of inodes in the file system
- incorrect directory inode numbers for the dot files (`.` and `..`)

Free list

The system updates the list of free blocks when a file has been deleted or when a file has been enlarged past a block boundary.

The file system maintains bitmaps that represent the state (allocated or unallocated) of each block contained within the file system. Inconsistencies can still arise if the bitmaps fail to be updated after blocks have been allocated or deallocated from an inode.

`fsck` phases

There are five file-system check phases in `fsck` (one of which is generally optional), as well as an initialization phase. Each phase of the `fsck` program passes through the whole file system. If you run `fsck` without a device name in the command line, `fsck` repeats all its phases for all devices listed in `/etc/fstab`.

Phase 1: Check blocks and sizes

The `fsck` program checks the inode list. In this phase, `fsck` may discover error conditions that result from checking inode types, setting up the zero-link-count table, checking inode block numbers for bad or duplicate blocks, checking inode size, and checking inode format. Phase 1B runs only if any duplicate blocks (that is, blocks that belong to multiple inodes) are found.

Phase 2: Check pathnames

The `fsck` program removes directory entries pointing to inodes that have error conditions from Phase 1 and Phase 1B. In this phase, `fsck` may discover error conditions that result from root inode mode and status, directory inode pointers out of range, and directory entries pointing to bad inodes.

Phase 3: Check connectivity

The `fsck` program checks the directory connectivity seen in Phase 2. In this phase, `fsck` may discover error conditions that result from unreferenced directories and missing or full `lost+found` directories.

Phase 4: Check reference counts

The `fsck` program reports messages that result from unreferenced files; a missing or full `lost+found` directory; incorrect link counts for files, directories, or special files; unreferenced files and directories; bad and duplicate blocks in files and directories; and incorrect total free inode counts.

Phase 5: Check cylinder groups

This phase is concerned with the free-block and used-inode maps. This section lists error conditions resulting from allocated blocks in the free-block maps, free blocks missing from free-block maps, and the total free-block count. It also lists error conditions resulting from free inodes in the used-inode maps, allocated inodes missing from used-inode maps, and the total used-inode count.

`lost+found` directory

At the top of each file-system hierarchy, a directory named `lost+found` is created when a new file system is made. If `fsck` detects a file that cannot be found in a directory listing, it places the file in the `lost+found` directory, using the file's inode number as a filename. The file can then be examined and moved back to its proper location.

Using `fsck`

You can use several options with the `fsck` utility, depending on whether you want to check the root file system or auxiliary file systems.

When to use `fsck`

Any file-system inconsistency will be made worse if you continue to use the file system (thus modifying it further) without running `fsck`. Because it is so important to keep your file systems consistent, the `fsck` program is built into the system startup procedure and is automatically run each time you start up A/UX. (See Chapter 3, “System Startup,” for a description of the startup process.)

You are encouraged to run `fsck` whenever you have doubts about the file system's integrity, such as after receiving disk error messages that display “hard read” or “hard write” errors. Typically, `fsck` is run in single-user mode, because in this state all file systems (except the root file system) are automatically unmounted. (See Chapter 3, “System Startup,” for instructions on how to reach single-user mode.)

The general syntax of `fsck` is shown here:

```
fsck [ options ] [ file-system ]
```

You specify *options* to direct `fsck` to run in different ways; see `fsck(1M)` in the manual pages for a complete list of options. You can specify *file-system* to run `fsck` on specific file systems. Without a *file-system* specification, `fsck` runs on the file system names listed in the file `/etc/fstab`; this file is used because it is the list of file systems to be automatically mounted when going to multi-user mode.

The file system on which `fsck` is running should be unmounted, or at least no writes should occur while `fsck` is running. This is important because `fsck` performs more than one pass on the file system. If the system is modified from pass to pass, the results are unpredictable.

When `fsck` finds an inconsistency in a file system, it informs you with a message like this:

```
POSSIBLE FILE SIZE ERROR I=2405
```

The message can also look like this:

```
FREE INODE COUNT WRONG IN SUPERBLK FIX?
```

The second message illustrates one of the interactive error messages produced by `fsck`. The program performs the corrective action only if you enter `y` to confirm that it should do so. If you enter `n`, it will either continue or terminate, depending on the nature of the problem encountered. The section “`fsck` Error Messages,” later in this chapter, contains an exhaustive list of error messages and suggested responses.

`fsck`: A sample interaction

If you bring the system to single-user mode (using the `init 1` command) and enter

```
fsck
```

`fsck` starts running. Because you didn't specify a file system, `fsck` reads the file `/etc/fstab`.

Note: If you are unfamiliar with the `fsck` program, keep in mind that you may exit the program at any point by pressing CONTROL-C.

For each file system checked, the screen displays a message similar to this one:

```
# fsck -n /dev/dsk/c0d0s0
** /dev/dsk/c0d0s0 (NO WRITE)
** Last Mounted on /
** Root file system
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
4511 file, 39695 used, 11695 free (127 frags, 2892 blocks,
0.2% fragmentation)
```

The `fsck` program lets you know which phase and what file system it is in at any given time. Different and separate file systems are discussed in the next section, "Multiple File Systems and `fsck`."

The program gives you a measure of the fragmentation of the file system; this tells you how scattered the files are. As a file system becomes increasingly fragmented, performance decreases because the system must spend more time seeking each successive data block. You can defragment the file system by backing up all the files in the file system, making a new file system, then restoring the files from your backups. (The files should not be removed until you have verified that the data on your backups is good; in this instance, having two copies of the backups is an excellent idea.) As a general rule, you will see modest performance gains after restoring a file system that is 10 percent fragmented, and significant gains after restoring a file system that is 40 percent fragmented.

Note: To defragment your file system by using backups, you must use a backup utility that does not copy the superblock. The `cpio` utility is a good choice, as is Retrospect Remote.

The preceding example illustrates a routine check during which no problems or inconsistencies were found. This next example indicates that problems are found. In this situation, the screen displays a message similar to this one:

```
# fsck -n /dev/dsk/c0d0s0
** /dev/dsk/c0d0s0 (NO WRITE)
** Last Mounted on /
** Root file system
** Phase 1 - Check Blocks and Sizes
POSSIBLE FILE SIZE ERROR I=1147
POSSIBLE FILE SIZE ERROR I=1195
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
4511 file, 39695 used, 11695 free (127 frags, 2892 blocks,
0.2% fragmentation)
```

However, the messages can be more obscure and require some action on your part. See “`fsck` Error Messages,” later in this chapter, for descriptions of error messages.

Multiple file systems and `fsck`

File-system checks occur when the system goes from single-user to multi-user mode. You need to take a few steps to make sure that `fsck` automatically checks file systems other than the root file system during system startup.

Two factors determine whether a file system is checked: options given to the `fsck` command and two fields in the `/etc/fstab` file. As shipped, the system automatically runs `fsck` during startup for the root file system and for files in `fstab`. The determinant fields in the `fstab` file are *type* (of the file system) and *pass-number*. Figure 10-4 shows a sample annotated `/etc/fstab` file. The values it contains cause `fsck` to check the listed file system.

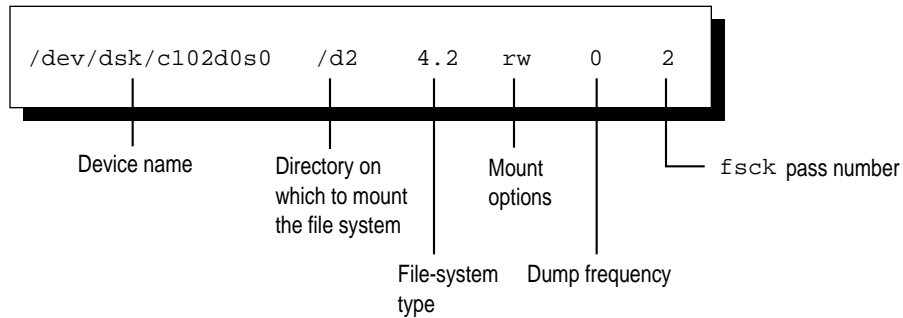


Figure 10-4 A description of sample entries in `/etc/fstab`

Figure 10-5 shows how `fsck` uses its options and the fields of `fstab` to decide whether to check a file system.

A bit in the superblock determines if `fsck` does a full check of the file system. If the file system was shut down correctly (and the superblock was written to disk), `fsck` doesn't have to run. If the file system was shut down without the superblock being written to the disk, `fsck` runs a full check. (This is why `fsck` appears to run quickly at some times and slowly at others.)

You can verify that `fsck` will run automatically at startup on a file system by checking the following details:

- The pass-number for the file system is a number greater than or equal to 2.
- The type of the file system is 4.2 or 5.2. (By default, A/UX makes UFS file systems; their type is 4.2.)

You can have `fsck` check all the file systems having the attributes listed in the `/etc/fstab` file by entering this command:

```
fsck -p
```

Or you can run `fsck` on an individual file system by giving this command:

```
fsck device-name
```

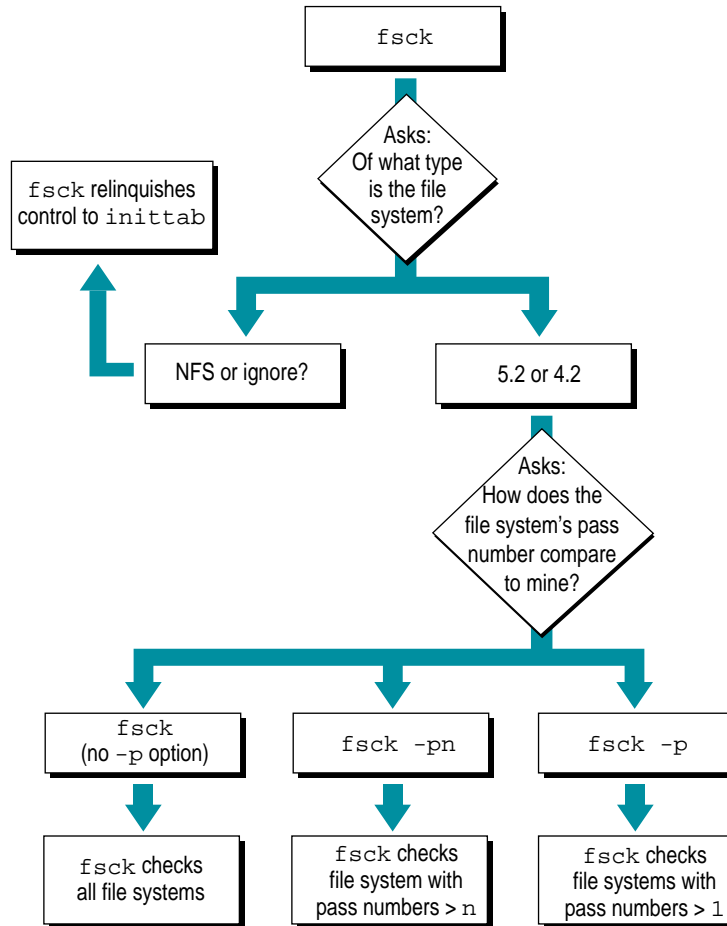


Figure 10-5 How `fsck` decides whether to check a file system

`fsck` error messages

The `fsck` program is a multipass utility. Each file-system pass runs a different phase of the `fsck` program. After the initial setup, `fsck` performs successive phases over each file system, checking blocks and sizes, pathnames, connectivity, reference counts, and the map of free blocks (possibly rebuilding it), and performs some cleanup. However, it is not always possible to correct file-system errors by using `fsck` once, as actions taken in later passes can affect the information acquired in earlier passes. Therefore, if errors are encountered you should run `fsck` repeatedly until the file system is error free.

Normally, `fsck` is run noninteractively to correct the file systems after an abrupt halt. These actions are a subset of those that `fsck` takes when it is running interactively. Note that many errors have more than one response that you can make.

When an inconsistency is detected, `fsck` reports the error condition to you in a message. If a response is required, `fsck` prints a prompt message and waits for a response. This section explains the possible messages in each phase, the meaning of each message, the possible and suggested responses, and the related error conditions. The suggested responses and the action resulting from the suggested responses are indicated in boldface.

WARNING Some suggested responses to `fsck` prompts will remove damaged files from your file system. These files may need to be recovered from your backups. It is therefore important to have current backups of all important files.

Messages and error conditions are organized by the `fsck` phase in which they can occur. Error conditions that may occur in more than one phase are described in the next section, “`fsck` Initialization Error Messages.”

Some messages require a yes or no response to a prompt. A “yes” response can be `Y` or `y`, and “no” can be `N` or `n`; these responses are shown in uppercase in the following sections. Some messages may indicate a serious problem because the file system cannot be completely checked. If the problem is in software, you may need to remake and restore one or more file systems. If the problem is in hardware, you may need to reinitialize your disk and completely restore all your files from backups.

`fsck` initialization error messages

Before a file-system check can be performed, certain files have to be opened. This section discusses error conditions resulting from command-line options, memory requests, opening of files, status of files, superblocks, and file-system checks.

`c` option?

In this error message, `c` stands for any character that is not a legal option to `fsck`. See the on-line manual page for legal options to `fsck`. The `fsck` program terminates on this error condition. Correct the options provided and rerun `fsck`.

```
cannot alloc n bytes for blockmap
cannot alloc n bytes for freemap
cannot alloc n bytes for statemap
cannot alloc n bytes for lncntp
```

These messages mean that a request for memory by `fsck` for its virtual memory tables failed. As a consequence, `fsck` terminates. This indicates a serious problem that usually requires restoring the file system that is being checked. If it is a large file system, you may not have enough virtual space to run `fsck`. Try halting any large processes—the Finder environment takes a lot of space—or adding additional swap space.

```
CANNOT READ: BLK b
CONTINUE?
```

This message indicates that the `fsck` program's request to read a specified block number *b* in the file system failed.

Here are possible responses to the `CONTINUE?` prompt:

Y Attempts to continue to run the file-system check. The `fsck` program attempts the read operation again. If the read operation fails again, `fsck` displays this message:

```
THE FOLLOWING SECTORS COULD NOT BE READ: n
```

In this message, *n* indicates the sectors that could not be read. If `fsck` ever tries to write back one of the blocks on which the read operation failed, it will print this message:

```
WRITING ZERO'ED BLOCK n TO DISK
```

In this message, *n* indicates the sector that was written with 0's. On either kind of file system (UFS or SVFS), if your disk is experiencing hardware problems, the problem will persist. This error condition will not allow a complete check of the file system. Test the disk by using the HD SC Setup utility or the formatting utility provided by the manufacturer of your hard drive. Rerun `fsck` to recheck this file system.

N Stops `fsck`.

```
CANNOT SEEK: BLK b
CONTINUE?
```

This message means that a request to move to a specified block number *b* in the file system failed.

Here are possible responses to the `CONTINUE?` prompt:

Y Attempts to continue to run the file-system check. If the problem persists, run `fsck` a second time to recheck this file system. If the problem still remains, test your hard disk drive using the HD SC Setup utility or the formatting utility provided by the manufacturer of your hard drive.

N Stops `fsck`.

```
CANNOT WRITE: BLK b
CONTINUE?
```

This message indicates that the `fsck` program's request to write a specified block number *b* in the file system failed. The disk is write-protected.

Here are possible responses to the `CONTINUE?` prompt:

Y Attempts to continue to run the file-system check. The write operation will be retried, with the failed blocks indicated by this message:

```
THE FOLLOWING SECTORS COULD NOT BE WRITTEN: n
```

In this message, *n* indicates the sectors that could not be written. If the disk is experiencing a hardware problem, the problem will persist. This error condition will not allow a complete check of the file system. Test the disk by using the HD SC Setup utility or the formatting utility provided by the manufacturer of your hard drive. Run `fsck` a second time to recheck this file system.

N Stops `fsck`.

```
can't open f
```

This message indicates that the file system called *f* cannot be opened for reading. The `fsck` program ignores this file system and continues checking the next file system given. Verify the existence and permissions of *f*, which should be at least 700. If the permissions are correct, try running `fsck` on the file system in single-user mode. Make sure that the drive containing *f* is turned on. If everything else looks good, the file system must be remade and restored.

```
can't open checklist file: f
```

This message means that the default file-system check file *f* (usually `/etc/fstab`) cannot be opened for reading; `fsck` terminates on this error condition. Check the permissions of *f*, which should be at least 400, and modify if necessary. If *f* is missing, it can be restored from backups.

```
can't stat f
can't make sense out of name f
```

This message indicates that the `fsck` program's request for statistics about file system *f* failed. It ignores this file system and continues checking the next file system given. Verify the existence and permissions of *f*, which should be at least 700. If the permissions are correct, try running `fsck` on the file system in single-user mode. Make sure that the drive containing *f* is turned on. If everything else looks good, the file system must be remade and restored.

```
can't stat root
```

This message means that the `fsck` program's request for statistics about the root directory (`/`) failed; `fsck` terminates on this error condition. Rerun `fsck` on the root file system in single-user mode. If this doesn't work, the file system must be remade and restored.

```
IMPOSSIBLE MINFREE=d IN SUPERBLOCK
SET TO DEFAULT?
```

This message indicates that the superblock minimum space percentage is more than 99 percent or less than 0 percent.

Here are possible responses to the `SET TO DEFAULT?` prompt:

Y Sets the `minfree` parameter to 10 percent. (If some other percentage is desired, it can be set; see `tunefs(8)`.)

N Ignores this error condition. One of the following messages appears:

```
MAGIC NUMBER WRONG
NCG OUT OF RANGE
CPG OUT OF RANGE
NCYL DOES NOT JIVE WITH NCG*CPG
SIZE PREPOSTEROUSLY LARGE
TRASHED VALUES IN SUPER BLOCK
```

Any of these messages will be followed by this message:

f: BAD SUPER BLOCK: *b*

USE `-b` OPTION TO `FSCK` TO SPECIFY LOCATION OF AN ALTERNATE SUPER-BLOCK TO SUPPLY NEEDED INFORMATION; SEE `fsck(1M)`.

The superblock has been corrupted. An alternative superblock must be selected from among those listed by `newfs` when the file system was created. For file systems with a block size less than 32K (as are the default A/UX file systems), rerun `fsck`, using this command:

```
fsck -b 32
```

f is not a block or character device; OK?

This message indicates that the `fsck` program has been given a regular filename by mistake. Verify the type of the file specified, or provide the name of a valid file system.

Here are possible responses to the OK? prompt:

Y Ignores this error condition.

N **Ignores this file system and continues checking the next file system given.**

f : (NO WRITE)

This message indicates that either the `-n` option was specified or the attempt by `fsck` to open the file system *f* for writing failed. When `fsck` is running manually, all the diagnostics are printed out, but no modifications are attempted to fix them. If the `-n` option was used, this is not an error. If the option was not specified, remove write protection from the appropriate disk.

UNDEFINED OPTIMIZATION IN SUPERBLOCK
SET TO DEFAULT?

This message means that the superblock optimization parameter is neither `OPT_TIME` nor `OPT_SPACE`.

Here are possible responses to the SET TO DEFAULT? prompt:

Y Sets the superblock to request optimization to minimize running time of the system. (Optimization to minimize disk space utilization can be set; see `tunefs(8)`.)

N Ignores this error condition.

Phase 1: Check blocks and sizes

Phase 1 is concerned with the inode list. This section lists error conditions resulting from checking inode types, setting up the zero-link-count table, examining inode block numbers for bad or duplicate blocks, checking inode size, and checking inode format.

b BAD I=*i*

This message means that inode *i* contains block number *b* with a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system. This error condition may generate the EXCESSIVE BAD BLKS error condition in Phase 1 (described later in this section) if inode *i* has too many block numbers outside the file-system range. This error condition always generates the DUPS/BAD error condition in Phase 2 and Phase 4.

BAD STATE *ddd* TO BLKERR

This message indicates that an internal error has scrambled the `fsck` state map to have the impossible value *ddd*. The `fsck` program exits immediately. This problem can often be solved by rerunning `fsck`. If the problem persists, the file system being checked must be remade and restored.

b DUP I=*i*

This message means that the inode *i* contains block number *b*, which is already claimed by another inode. This error condition may generate the EXCESSIVE DUP BLKS error condition in Phase 1 if inode *i* has too many block numbers claimed by other inodes. This error condition always runs Phase 1B and the DUPS/BAD error condition in Phase 2 and Phase 4.

DUP TABLE OVERFLOW
CONTINUE?

This message indicates that an internal table in `fsck` containing duplicate block numbers cannot allocate any more space.

Here are possible responses to the `CONTINUE?` prompt:

Y Continues with the program. This error condition will not allow a complete check of the file system. A second run of `fsck` should be made to recheck this file system. If too many duplicate blocks are found, this error condition is repeated.

N **Stops the program.** Increase the amount of virtual memory available and rerun `fsck`.

EXCESSIVE BAD BLKS I=*i*
CONTINUE?

This message means that the number of blocks with a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system associated with inode *i* is too high to be acceptable. The usual cutoff point is 10.

Here are possible responses to the `CONTINUE?` prompt:

Y **Ignores the rest of the blocks in this inode and continues checking with the next inode in the file system.** This error condition will not allow a complete check of the file system. Rerun `fsck` to check this file system again.

N Stops the program. Rerun `fsck` to check this file system again.

EXCESSIVE DUP BLKS I=*i*
CONTINUE?

This message indicates that the number of blocks claimed by other inodes is too high to be acceptable. The usual cutoff point is 10.

Here are possible responses to the `CONTINUE?` prompt:

- Y Ignores the rest of the blocks in this inode and continues checking with the next inode in the file system.** This error condition will not allow a complete check of the file system. Rerun `fsck` to recheck this file system.
- N** Stops the program. Rerun `fsck` to recheck this file system.

```
INCORRECT BLOCK COUNT I=i (x should be y)
CORRECT?
```

This message indicates that the block count for inode *i* is *x* blocks, but should be *y* blocks.

Here are possible responses to the `CORRECT?` prompt:

- Y Replaces the block count of inode *i* with *y*.**
- N** Ignores this error condition.

```
LINK COUNT TABLE OVERFLOW
CONTINUE?
```

This message means that an internal table for `fsck` containing allocated inodes with a link count of zero cannot allocate more memory.

Here are possible responses to the `CONTINUE?` prompt:

- Y Continues with the program.** This error condition makes a complete check of the file system impossible. Rerun `fsck` to recheck this file system. If another allocated inode with a link count of 0 is found, this error condition is repeated.
- N** Stops `fsck`.

```
PARTIALLY ALLOCATED INODE I=i
CLEAR?
```

This message means that inode *i* is neither allocated nor unallocated.

Here are possible responses to the `CLEAR?` prompt:

- Y Deallocates inode *i*.** You may need to restore a file from backups.
- N** Ignores this error condition.

```
PARTIALLY TRUNCATED INODE I=i
SALVAGE?
```

This message indicates that the `fsck` program has found inode *i* whose size is shorter than the number of blocks allocated to it. This condition should occur only if the system crashes while truncating a file.

Here are possible responses to the `SALVAGE?` prompt:

Y Completes the truncation to the size specified in the inode.

N Ignores this error condition.

```
UNKNOWN FILE TYPE I=i
CLEAR?
```

This message means that the mode word of the inode *i* indicates that the inode is not a recognized A/UX file type. This problem is usually caused by a corruption of the mode bits.

Here are possible responses to the `CLEAR?` prompt:

Y Deallocates inode *i*. This always generates the `UNALLOCATED` error condition in Phase 2 for each directory entry pointing to this inode. A file may need to be restored from backups.

N Ignores this error condition.

Phase 1B: Rescan for more duplicates

When a duplicate block is found in the file system, the file system is rescanned to find the inode that previously claimed that block. This section lists the error condition. If the duplicate block is found, the following error condition occurs:

```
b DUP I=i
```

This message means that inode *i* contains block number *b*, which is already claimed by another inode. This error condition action always generates the `DUPS/BAD` error condition in Phase 2. You can determine which inodes have overlapping blocks by examining this error condition and the `DUP` error condition in Phase 1.

Phase 2: Check pathnames

This phase is concerned with removing directory entries pointing to inodes that had error conditions from Phase 1 and Phase 1B. This section lists error conditions resulting from root inode mode and status, directory inode pointers in range, directory entries pointing to bad inodes, and directory integrity checks.

```
BAD INODE NUMBER FOR '.' I=i OWNER=o MODE=m SIZE=s MTIME=t
DIR=f
FIX?
```

This message means that a directory *i* has been found whose inode number for itself (dot, or *.*) does not equal *i*.

Here are possible responses to the `FIX?` prompt:

Y Changes the inode number for dot (*.*) to be equal to *i*.

N Leaves the inode number for dot (*.*) unchanged.

```
BAD INODE NUMBER FOR '..' I=i OWNER=o MODE=m SIZE=s MTIME=t
DIR=f
FIX?
```

This message indicates that a directory has been found whose inode number for its parent (dot-dot, or *..*) does not equal the parent of *i*.

Here are possible responses to the `FIX?` prompt:

Y Changes the inode number for dot-dot (*..*) to be equal to the parent of dot-dot (*..*). In the root inode, dot-dot (*..*) points to itself.

N Leaves the inode number for dot-dot (*..*) unchanged.

```
BAD INODE s TO DESCEND
```

This message means that an internal error has caused an impossible state *s* to be passed to the routine that descends the file-system directory structure. The `fsck` program exits. This problem can often be solved by rerunning `fsck`. If the problem persists, the file system being checked must be remade and restored.

BAD RETURN STATE *s* FROM DESCEND

This message means that an internal error has caused an impossible state *s* to be returned from the routine that descends the file-system directory structure. The `fsck` program exits. This problem can often be solved by rerunning `fsck`. If the problem persists, the file system being checked must be remade and restored.

BAD STATE *s* FOR ROOT INODE

This message means that an internal error has caused an impossible state *s* to be assigned to the root inode. The `fsck` program exits. This problem can often be solved by rerunning `fsck`. If the problem persists, the file system being checked must be remade and restored.

DIRECTORY CORRUPTED I=*i* OWNER=*o* MODE=*m* SIZE=*s* MTIME=*t* DIR=*f*
SALVAGE?

This message indicates that a directory with an inconsistent internal state has been found.

Here are possible responses to the SALVAGE? prompt:

Y Throws away all entries up to the next directory boundary (usually 512 bytes). This drastic action can throw away up to 42 entries. You may need to restore several files from backups. The file information provides a clue to the file or directory that may need to be restored.

N Skips to the next directory boundary and resumes reading, but does not modify the directory.

DIRECTORY *f* LENGTH *s* NOT MULTIPLE OF *b*
ADJUST?

This message means that a directory *f* has been found with size *s* that is not a multiple of the directory block size.

Here are possible responses to the ADJUST? prompt:

Y The length is rounded up to the appropriate block size.

N Ignores this error condition.

```
DIRECTORY TOO SHORT I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f
FIX?
```

This message indicates that a directory *f* has been found whose size *s* is less than the minimum size directory. The owner *o*, mode *m*, size *s*, modify time *t*, and directory name *f* are printed.

Here are possible responses to the `FIX?` prompt:

Y Increases the size of the directory to the minimum directory size.

N Ignores this directory.

```
DUPS/BAD I=i OWNER=o MODE=m SIZE=s MTIME=t FILE=f
REMOVE?
```

This message indicates that during Phase 1 or Phase 1B, `fsck` has found duplicate blocks or bad blocks associated with directory entry *f*, inode *i*. The owner *o*, mode *m*, size *s*, modify time *t*, and filename *f* are printed.

Here are possible responses to the `REMOVE?` prompt:

Y Removes the directory entry *f*. You may need to restore a file from backups. The file information provides a clue to the file that may need to be restored.

N Ignores this error condition.

```
DUPS/BAD IN ROOT INODE
REALLOCATE?
```

This message indicates that during Phase 1 or Phase 1B, `fsck` has found duplicate blocks or bad blocks in the root inode (usually inode number 2) for the file system.

Here are possible responses to the `REALLOCATE?` prompt:

Y Clears the existing contents of the root inode and reallocates it. The files and directories usually found in the root inode will be recovered in Phase 3 and put into `lost+found`.

If the attempt to allocate the root inode fails, `fsck` exits with the message `CANNOT ALLOCATE ROOT INODE`. If this error message occurs, you have little choice but to restore the file system being checked from backups.

N The `fsck` program prompts with `CONTINUE?`

Here are possible responses to the `CONTINUE?` prompt:

Y Ignores the `DUPS/BAD` error condition in the root inode and attempts to continue to run the file-system check. If the root inode is not correct, then this action may result in many other error conditions.

N Stops the program.

```
EXTRA '.' ENTRY I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f
FIX?
```

This error message means that a directory *i* has been found that has more than one entry for dot (`.`).

Here are possible responses to the `FIX?` prompt:

Y Removes the extra entry for dot (`.`).

N Leaves the directory unchanged.

```
EXTRA '..' ENTRY I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f
FIX?
```

This message means that a directory *i* has been found that has more than one entry for dot-dot (`..`).

Here are possible responses to the `FIX?` prompt:

Y Removes the extra entry for dot-dot (`..`).

N Leaves the directory unchanged.

```
n IS AN EXTRANEOUS HARD LINK TO A DIRECTORY d
REMOVE?
```

This message indicates that the `fsck` program has found a hard link, *n*, to a directory, *d*.

Here are possible responses to the `REMOVE?` prompt:

Y Deletes the extraneous entry, *n*.

N Ignores the error condition.

```
MISSING '.' I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f
FIX?
```

This message means that a directory *i* has been found whose first entry is unallocated.

Here are possible responses to the `FIX?` prompt:

Y Builds an entry for dot (.) with inode number equal to *i*.

N Leaves the directory unchanged.

```
MISSING '.' I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f
CANNOT FIX, FIRST ENTRY IN DIRECTORY CONTAINS f
```

This message indicates that a directory *i* has been found whose first entry is *f*. The `fsck` program cannot resolve this problem. Mount the file system and move the entry *f* elsewhere. Unmount the file system and run `fsck` again.

```
MISSING '.' I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f
CANNOT FIX, INSUFFICIENT SPACE TO ADD '.'
```

This message means that a directory inode *i* (with name *f*) has been found whose first entry is not dot (.). The `fsck` program cannot resolve this problem. Write down the inode and clear the directory inode using `clri(IM)`. Restore the directory from backups.

```
MISSING '..' I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f
FIX?
```

This message indicates that a directory *i* has been found whose first entry is unallocated.

Here are possible responses to the `FIX?` prompt:

Y Builds an entry for dot-dot (..) with inode number equal to the parent of dot-dot (..). In the root inode, dot-dot (..) points to itself.

N Leaves the directory unchanged.

```
MISSING '..' I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f  
CANNOT FIX, SECOND ENTRY IN DIRECTORY CONTAINS f
```

This message indicates that a directory *i* has been found whose second entry is *f*. The `fsck` program cannot resolve this problem. Mount the file system and move the entry *f* elsewhere. Unmount the file system and run `fsck` again.

```
MISSING '..' I=i OWNER=o MODE=m SIZE=s MTIME=t DIR=f  
CANNOT FIX, INSUFFICIENT SPACE TO ADD '..'
```

This message means that a directory inode *i* (with name *f*) has been found whose second entry is not dot-dot (. .). The `fsck` program cannot resolve this problem. Write down the inumber and clear the directory inode using `clrri(1M)`. Restore the directory from backups.

```
NAME TOO LONG f
```

This message means that an excessively long pathname has been found. This usually indicates loops in the file-system name space, a condition that can occur if a user logged in as `root` has made circular links to directories. The file system must be reinitialized and restored from backups. The circular links can be removed by a technical expert, but the procedure is beyond the scope of this book.

```
i OUT OF RANGE I=i NAME=f  
REMOVE?
```

This message indicates that a directory entry *f* has an inode number that is greater than the end of the inode list.

Here are possible responses to the `REMOVE?` prompt:

Y Removes the directory entry *f*. You may need to restore a file from backups.

N Ignores this error condition.

```
ROOT INODE NOT DIRECTORY
REALLOCATE?
```

This message means that the root inode (usually inode number 2) is not of the directory type.

Here are possible responses to the `REALLOCATE?` prompt:

Y Clears the existing contents of the root inode and reallocates it. The files and directories usually found in the root inode are recovered in Phase 3 and put into the directory `lost+found`.

If the attempt to allocate the root inode fails, `fsck` exits with the message `CANNOT ALLOCATE ROOT INODE`. If this error message occurs, you have little choice but to restore the file system being checked from backups.

N **The `fsck` program prompts with `FIX?`**

Here are possible responses to the `FIX?` prompt:

Y **Makes the root inode's type a directory.** If the root inode's data blocks are not directory blocks, many error conditions are produced.

N Stops the program.

```
ROOT INODE UNALLOCATED
ALLOCATE?
```

This message means that the root inode (usually inode number 2) has no allocate-mode bits.

Here are possible responses to the `ALLOCATE?` prompt:

Y **Allocates inode 2 as the root inode.** The files and directories usually found in the root are recovered in Phase 3 and put into `lost+found`. If the attempt to allocate the root fails, `fsck` exits with the message `CANNOT ALLOCATE ROOT INODE`.

If this error message occurs, you have little choice but to remake and restore the file system being checked from backups.

N Causes `fsck` to exit.

ROOT INODE UNALLOCATED. TERMINATING

This message means that the root inode (always inode number 2) has no allocated mode bits. This error condition indicates a serious problem that you may not be able to solve without remaking and restoring the file system being checked. The program ends.

UNALLOCATED I=*i* OWNER=*o* MODE=*m* SIZE=*s* MTIME=*t* DIR=*f*
REMOVE?

This message indicates that a directory or file entry *f* points to an unallocated inode *i*. The owner *o*, mode *m*, size *s*, modify time *t*, and directory entry *f* are printed.

Here are possible responses to the REMOVE? prompt:

Y Removes the directory entry *f*. You may need to restore a file or directory from backups. The file information provides a clue to the file that may need to be restored.

N Ignores this error condition.

ZERO LENGTH DIRECTORY I=*i* OWNER=*o* MODE=*m* SIZE=*s* MTIME=*t*
DIR=*f*
REMOVE?

This message means that a directory entry *f* has a size *s* that is 0. The owner *o*, mode *m*, size *s*, modify time *t*, and directory name *f* are printed.

Here are possible responses to the REMOVE? prompt:

Y Removes the directory entry *f*. This action always generates the DUPS/BAD error condition in Phase 4. You may need to restore a directory from backups. The file information provides a clue as to what directory may need to be restored.

N Ignores this error condition.

Phase 3: Check connectivity

Phase 3 is concerned with the directory connectivity seen in Phase 2. This section lists error conditions that result from unreferenced directories and missing or full `lost+found` directories.

```
BAD INODE s TO DESCEND
```

This message indicates that an internal error has caused an impossible state *s* to be passed to the routine that descends the file-system directory structure. The `fsck` program exits. This problem can often be solved by rerunning `fsck`. If the problem persists, the file system being checked must be remade and restored.

```
DIR I=i CONNECTED. PARENT WAS I=j
```

This advisory message indicates that a directory inode *i* was successfully connected to the `lost+found` directory. The parent inode *j* of the directory inode *i* is replaced by the inode number of the `lost+found` directory. You can use the command

```
find / -i j -print
```

to determine the former parent directory's name, then move the directory with inode *i* back to its normal location.

```
DIRECTORY f LENGTH s NOT MULTIPLE OF b  
ADJUST?
```

This message means that a directory *f* has been found with size *s* that is not a multiple of the directory block size *b*. (This can recur in Phase 3 if it is not adjusted in Phase 2.)

Here are possible responses to the ADJUST? prompt:

Y **The length is rounded up to the appropriate block size.** A warning is printed and the directory is adjusted.

N Ignores the error condition.

```
lost+found IS NOT A DIRECTORY
REALLOCATE?
```

This message indicates that the entry for `lost+found` is not a directory.

Here are possible responses to the `REALLOCATE?` prompt:

Y Allocates a directory inode and changes `lost+found` to refer to it. The previous inode reference by the `lost+found` name is not cleared. It will either be reclaimed as an unreferenced inode or have its link count adjusted later in this phase. Inability to create a `lost+found` directory generates the message

```
SORRY. CANNOT CREATE lost+found DIRECTORY
```

and stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4.

N Stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4.

```
NO lost+found DIRECTORY
CREATE?
```

This message means that there is no `lost+found` directory in the root directory of the file system; `fsck` can try to create one.

Here are possible responses to the `CREATE?` prompt:

Y Creates a `lost+found` directory in the root directory of the file system. This may generate the following message:

```
NO SPACE LEFT IN / (EXPAND).
```

See the next message for the possible responses. Inability to create a `lost+found` directory generates the message

```
SORRY. CANNOT CREATE lost+found DIRECTORY
```

and stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4.

N Stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4.

```
NO SPACE LEFT IN /lost+found DIRECTORY
EXPAND?
```

This message means there is no space to add another entry to the `lost+found` directory in the root directory of the file system.

Here are possible responses to the `EXPAND?` prompt:

Y Expands the `lost+found` directory to make room for the new entry. If the attempted expansion fails, `fsck` prints the message

```
SORRY. NO SPACE IN lost+found DIRECTORY
```

and stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4. Clean out unnecessary entries in `lost+found`.

N Stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4.

```
UNREF DIR I=i OWNER=o MODE=m SIZE=s MTIME=t
RECONNECT?
```

This message indicates that the directory inode *i* was not connected to a directory entry when the file system was traversed. The owner *o*, mode *m*, size *s*, and modify time *t* of the directory inode are printed. While being checked, the directory is reconnected if its size is nonzero; otherwise, it is cleared.

Here are possible responses to the `RECONNECT?` prompt:

Y Reconnects the directory inode *i* to the file system in the directory for lost files (usually `lost+found`). This action may generate the `lost+found` error condition in Phase 3 if there are problems connecting the directory inode to `lost+found`. This action may also generate the `CONNECTED` error condition in Phase 3 if the link was successful.

N Ignores this error condition. This action always generates the `UNREF` error condition in Phase 4.

Phase 4: Check reference counts

Phase 4 is concerned with the link-count information of Phase 2 and Phase 3. This section lists error conditions resulting from unreferenced files; a missing or full `lost+found` directory; incorrect link counts for files; unreferenced files and directories; and bad or duplicate blocks in files and directories. All errors in this phase except for running out of space in the `lost+found` directory are correctable.

```
DUPS/BAD FILE I=i OWNER=o MODE=m SIZE=s MTIME=t
CLEAR?
```

This message indicates that during Phase 1 or Phase 1B, `fsck` has found duplicate blocks or bad blocks associated with file inode *i*. The owner *o*, mode *m*, size *s*, and modify time *t* of inode *i* are printed.

Here are possible responses to the `CLEAR?` prompt:

Y Deallocates inode *i*. You may need to restore the file from backups. The file information provides a clue as to what file may need to be restored.

N Ignores this error condition.

```
LINK COUNT FILE I=i OWNER=o MODE=m SIZE=s MTIME=t COUNT=x
SHOULD BE y
ADJUST?
```

This message means that the link count for inode *i*, which is a file, is *x* but should be *y*. The owner *o*, mode *m*, size *s*, and modify time *t* are printed. The `fsck` program exits with the message `LINK COUNT INCREASING`.

Here are possible responses to the `ADJUST?` prompt:

Y Replaces the link count of file inode *i* with *y*.

N Ignores this error condition.

```
lost+found IS NOT A DIRECTORY
REALLOCATE?
```

This message means that the entry for `lost+found` is not a directory.

Here are possible responses to the `REALLOCATE?` prompt:

Y Allocates a directory inode and changes `lost+found` to refer to it. The previous inode reference by the `lost+found` name is not cleared. It is either reclaimed as an unreferenced inode or has its link count adjusted later in this phase. Inability to create a `lost+found` directory generates the message

```
SORRY. CANNOT CREATE lost+found DIRECTORY
```

and stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4. One or more files may need to be restored from backups.

N Stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4. One or more files may need to be restored from backups.

```
NO lost+found DIRECTORY
CREATE?
```

This message indicates that there is no `lost+found` directory in the root directory of the file system; `fsck` ignores the request to link a file in `lost+found`. See the entry for `mklost+found(1M)` in the manual pages for additional information.

Here are possible responses to the `CREATE?` prompt:

Y Creates a `lost+found` directory in the root of the file system. This may raise the message: `NO SPACE LEFT IN / (EXPAND)`.

See the following message for the possible responses. Inability to create a `lost+found` directory generates the message

```
SORRY. CANNOT CREATE lost+found DIRECTORY
```

and stops the attempt to link up the lost inode. This action always generates the `UNREF` error condition in Phase 4. One or more files may need to be restored from backups.

N Stops the attempt to link up the lost inode. This action always generates the UNREF error condition in Phase 4. One or more files may need to be restored from backups.

```
NO SPACE LEFT IN lost+found DIRECTORY
EXPAND?
```

This message means that there is no space to add another entry to the `lost+found` directory in the root directory of the file system; `fsck` ignores the request to link the file. Check the size and contents of `lost+found`.

Here are possible responses to the `EXPAND?` prompt:

Y Expands the `lost+found` directory to make room for the new entry. If the attempted expansion fails, `fsck` prints the message

```
SORRY. NO SPACE IN lost+found DIRECTORY
```

and stops the attempt to link up the lost inode. This action always generates the UNREF error condition in Phase 4. Clean out unnecessary entries in `lost+found`. One or more files may have to be restored from backups.

N Stops the attempt to link up the lost inode. This action always generates the UNREF error condition in Phase 4. One or more files may have to be restored from backups.

```
UNREF FILE I=i OWNER=o MODE=m SIZE=s MTIME=t
RECONNECT?
```

This message means that inode *i* was not connected to a directory entry when the file system was traversed. The owner *o*, mode *m*, size *s*, and modify time *t* of inode *i* are printed. If the `-n` option is not set and the file system is not mounted, empty files are not reconnected and are cleared automatically.

Here are possible responses to the `RECONNECT?` prompt:

Y Reconnects inode *i* to the file system in the directory for lost files, usually `lost+found`. This may generate a `CLEAR?` error condition (described in the following section) if there are problems connecting inode *i* to `lost+found`.

N Ignores this error condition. This action always generates the `CLEAR?` error condition (described next).

CLEAR?

The inode mentioned in the immediately preceding error condition cannot be reconnected.

Here are possible responses to the CLEAR? prompt:

Y Deallocates the inode mentioned in the immediately preceding error condition. A file may need to be restored from backups.

N Ignores this error condition.

```
UNREF FILE I=i OWNER=o MODE=m SIZE=s MTIME=t
CLEAR?
```

This message indicates that inode *i*, which is a file, was not connected to a directory entry when the file system was traversed. The owner *o*, mode *m*, size *s*, and modify time *t* of inode *i* are printed. If you don't set the `-n` option and the file system is not mounted, empty files are cleared automatically.

Here are possible responses to CLEAR? prompt:

Y Deallocates inode *i*. You may need to restore a file from backups. The file information provides a clue to the file that may need to be restored.

N Ignores this error condition.

Phase 5: Check cylinder groups

This phase is concerned with the free-block and used-inode maps. This section lists error conditions resulting from allocated blocks in free-block maps, free blocks missing from free-block maps, and the total free-block count. It also lists error conditions resulting from free inodes in used-inode maps, allocated inodes missing from used-inode maps, and incorrect total used-inode counts.

```
BLK(S) MISSING IN BIT MAPS
SALVAGE?
```

This message means that a cylinder-group-block map is missing some free blocks; the maps are reconstructed.

Here are possible responses to the SALVAGE? prompt:

Y Reconstructs the free-block map.

N Ignores this error condition.

```
CG c: BAD MAGIC NUMBER
```

This message indicates that the identifying number (the magic number) of cylinder group *c* is wrong. This usually indicates that the cylinder group maps have been destroyed. When you run the `fsck` command at the A/UX Startup command line, the cylinder group is marked as needing to be reconstructed.

```
FREE BLK COUNT(S) WRONG IN SUPERBLOCK  
SALVAGE?
```

This message indicates that the superblock free-block information is incorrect and asks if it should be recomputed.

Here are possible responses to the SALVAGE? prompt:

Y Reconstructs the superblock free-block information.

N Ignores this error condition.

```
SUMMARY INFORMATION BAD  
SALVAGE?
```

This message means that the summary information was found to be incorrect.

Here are possible responses to the SALVAGE? prompt:

Y Reconstructs the summary information.

N Ignores this error condition.

Cleanup

Once a file system has been checked, a few cleanup functions are performed. The `fsck` program displays messages about the file system and its modify status.

```
v files, w used, x free (y frags, z blocks,  
f % fragmentation)
```

This message indicates that the file system checked contained v files using w fragment-sized blocks, leaving x fragment-sized blocks free in the file system. The numbers in parentheses break the free count down into y free fragments and z free full-sized blocks.

```
***** REBOOT A/UX! *****
```

This message indicates that the root file system has been modified by `fsck`. `A/UX` restarts the system.

```
***** FILE SYSTEM WAS MODIFIED *****
```

This message indicates that the current file system was modified by `fsck`.

Appendix A CommandShell Menu Items

This appendix provides convenient procedures for activating the CommandShell and Finder applications. It also provides a brief description of each menu item that is available when you are using CommandShell.



Activating CommandShell

A convenient way to access the CommandShell application is through the Application menu, at the far-right end of the menu bar.

Note: You should never quit the CommandShell application, as this will interrupt the system's ability to neatly log error messages. To stop using CommandShell, you can switch to another application or choose Hide CommandShell from the Application menu.

When CommandShell is the active application, it displays an untitled CommandShell window like the one in Figure A-1. If you have used a CommandShell window earlier and have not closed it, it is brought in front of other windows and becomes the active window. See Chapter 1, "Using CommandShell," for more information.

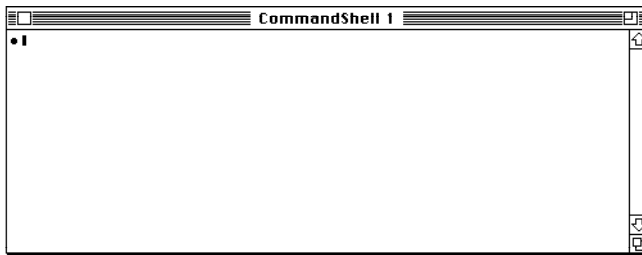


Figure A-1 A CommandShell window

Activating the Finder

Choose Finder from the Application menu to return to the Finder desktop from CommandShell. If A/UX is running a process when you switch to the Finder, the process continues to run in the background.

The CommandShell environment

This section describes the commands to which you have access when you're working with CommandShell.

The CommandShell File menu

The CommandShell File menu contains the commands that manage the CommandShell window environment.

File	
New	⌘N
Open...	⌘O
Close	⌘W
Save Selection...	
Page Setup...	
Print Selection...	
Close All Windows	
Quit	⌘Q

New

This command creates a new CommandShell window. The COMMAND-key equivalent is COMMAND-N.

Open...

This command displays a dialog box that allows you to open a selected file. The COMMAND-key equivalent is COMMAND-O. When used with utilities or applications, Open means “run.”

Close

This command closes the active window. The COMMAND-key equivalent is COMMAND-W.

Save Selection...

This command displays a dialog box that allows you to save, under a name you specify, any text you select in a CommandShell window.

Page Setup...

This command displays a dialog box that allows you to alter printer settings.

Print Selection...

This command displays a dialog box that allows you to send to the printer any data you have selected in a CommandShell window.

Close All Windows

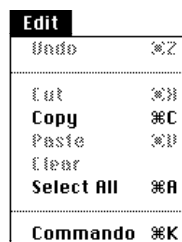
This command closes all CommandShell windows. Holding down the OPTION key and clicking a close box will also close all windows.

Quit

This command turns off CommandShell. The COMMAND-key equivalent is COMMAND-Q. Use this only when it is necessary to turn off all applications in order to load fonts; otherwise, leave CommandShell by reactivating the Finder or whatever other application you need to use. If you choose Quit, CommandShell is no longer available in the Application menu. If you accidentally quit CommandShell, restart it from the Apple menu. (There is an alias of the CommandShell file in the Apple Menu Items folder that is in the System Folder. Do not remove this alias.)

The CommandShell Edit menu

The CommandShell Edit menu contains the commands that you use to edit text in windows.



Edit	
Undo	⌘Z
Cut	⌘H
Copy	⌘C
Paste	⌘V
Clear	
Select All	⌘A
Commando	⌘K

Undo

This command nullifies the last text-editing or formatting change that you made. It does not nullify command execution. The `COMMAND`-key equivalent is `COMMAND-Z`.

Cut

This command copies any selected text to the Clipboard and removes the text from the window. The text is stored in the Clipboard until it's replaced by text sent there as a result of another Cut or Copy command. The `COMMAND`-key equivalent is `COMMAND-X`.

Copy

This command copies any selected text to the Clipboard but does not remove it from the window. The text is stored in the Clipboard until it's replaced by text sent there as a result of another Cut or Copy command. The `COMMAND`-key equivalent is `COMMAND-C`.

Paste

This command inserts the contents of the Clipboard into the window at the insertion point. The `COMMAND`-key equivalent is `COMMAND-V`.

Clear

This command removes any selected text from the window and does not copy it to the Clipboard. Text removed with the Clear command is no longer available. Choosing Clear is equivalent to pressing the `DELETE` key.

Select All

This command selects all of the text in the window. The `COMMAND`-key equivalent is `COMMAND-A`.

Commando

This command helps you build A/UX command lines, including options and arguments.

The CommandShell Window menu

The CommandShell Window menu contains the commands you use to manipulate and access CommandShell windows.

Window	
Tile	⌘T
Tile Horizontal	
Tile Vertical	
Standard Positions	

Standard Size	⌘S
Full Height	⌘F
Zoom Window	⌘/
Hide "CommandShell 2"	⌘H
Show All Windows	
Previous Window	⌘L
Rotate Windows	⌘R

A/UX System Console	⌘0
CommandShell 1	⌘1
CommandShell 2	⌘2

WARNING If a CommandShell window is horizontally reduced in size (because of tiling or resizing) such that its contents cannot be seen, CommandShell erases the obscured screen output. You cannot retrieve the contents by enlarging the window as you would be able to do with a standard Macintosh window.

Tile

This command rearranges all windows on the desktop to provide equal screen space, as shown in Figure A-2. The `COMMAND`-key equivalent is `COMMAND-T`.

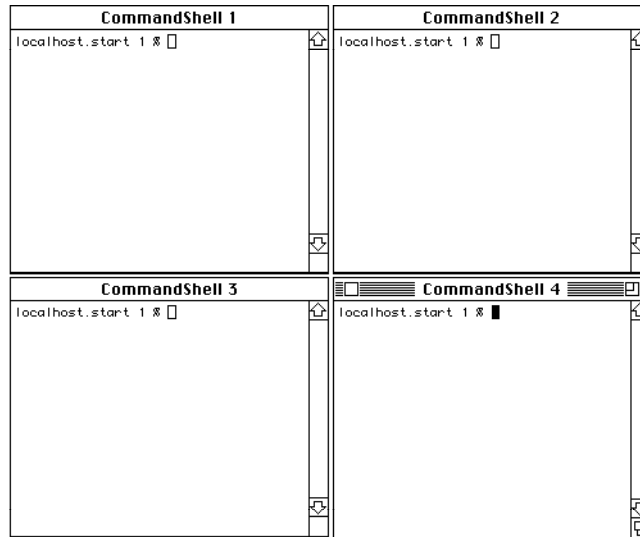


Figure A-2 Tiled windows

Tile Horizontal

This command rearranges all windows on the desktop, providing maximum length to each window and equal space vertically, as shown in Figure A-3.

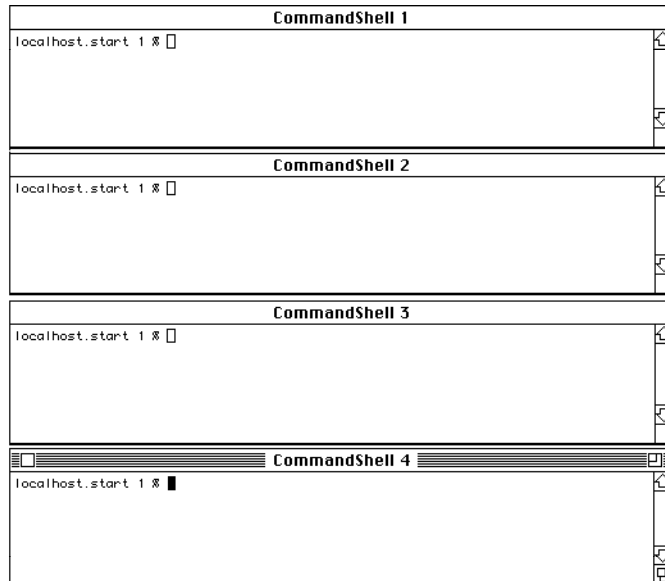


Figure A-3 Horizontally tiled windows

Tile Vertical

This command rearranges all windows on the desktop, providing the maximum number of lines and equal horizontal space, as shown in Figure A-4.

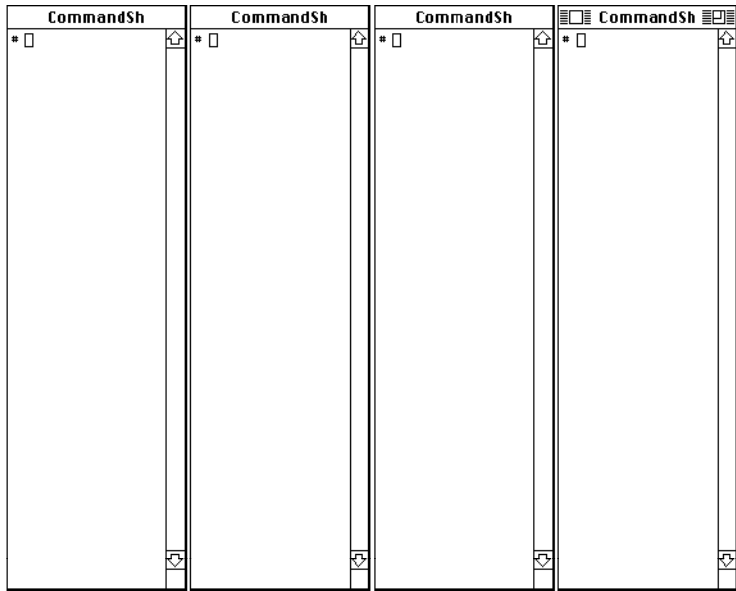


Figure A-4 Vertically tiled windows

Standard Positions

This command rearranges all windows on the desktop as shown in Figure A-5.

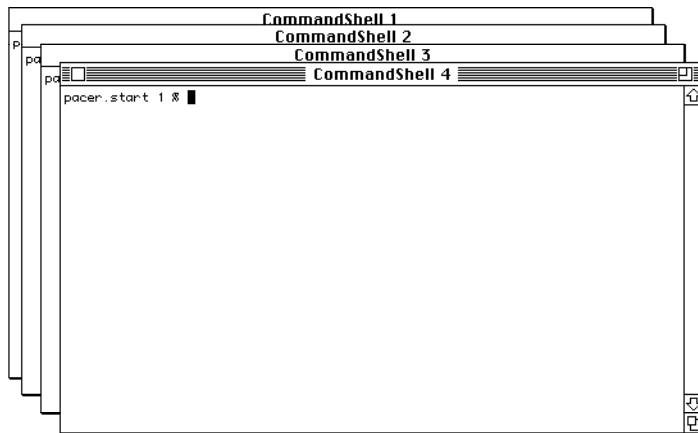


Figure A-5 Standard window positioning

This is the standard stacking arrangement of all CommandShell windows as they are created. If a window has been resized prior to stacking, it retains that size.

Standard Size

This command returns the active window to its standard size, if it has been changed. The COMMAND-key equivalent is COMMAND-S.

Full Height

This command changes the size of the active window so that it is as long as the screen allows. The COMMAND-key equivalent is COMMAND-F.

Zoom Window

This command increases the size of the active window so that it is as large as the screen allows. Choosing Zoom Window is equivalent to clicking the zoom box in the upper-right corner of the window. The COMMAND-key equivalent is COMMAND-/.

Hide “Window Name”

This command hides the active window from view. The actual window name appears in place of the phrase *Window Name*. The window is not closed but merely hidden from view. All accessible CommandShell windows are listed at the bottom of the Window menu. The COMMAND-key equivalent is COMMAND-H.

Show All Windows

This command displays all of the CommandShell windows listed at the bottom of the Window menu, including the A/UX System Console window, which contains system messages about all of the currently running A/UX system processes.

Previous Window

This command redisplay the last window you accessed *before* you activated the current window. The COMMAND-key equivalent is COMMAND-L.

Rotate Windows

This command brings the rear window to the front each time the command is chosen from the menu. The `COMMAND`-key equivalent is `COMMAND-R`.

A/UX System Console

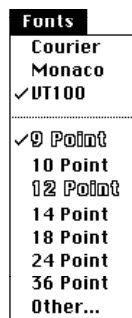
This command activates the A/UX System Console window, which contains system messages about all of the currently running A/UX system processes. The `COMMAND`-key equivalent is `COMMAND-0` (zero).

Names of all accessible CommandShell windows

At the bottom of the menu is a list of all currently accessible CommandShell windows, including any hidden windows. Choosing a window name in the list brings that window to the front as the active window.

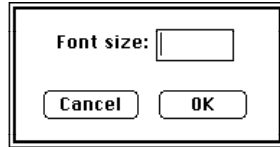
The CommandShell Fonts menu

The CommandShell Fonts menu contains the list of available fonts that you can use in CommandShell windows. Only fixed-width (monospaced) fonts are supported in CommandShell. When you choose a font, all of the text in the window, not just the selected text, is displayed in that font.



The CommandShell Fonts menu allows you to specify which of the available fonts to use in the CommandShell window, as well as the point sizes that determine the size of the text as displayed on the screen and as printed. The smallest preselected point size is 9 point, and the largest is 36 point.

The last item in the menu (Other) allows you to enter any desired type size. When you select Other, a dialog box appears in which you can type the size.



The CommandShell Commands menu

The CommandShell Commands menu allows you to change the environment within CommandShell windows.



Don't Record Lines Off Top

This command marks the window as the barrier for text entry. Any text exceeding the boundaries of the active window is not recorded and thus not accessible. If selected, the menu command changes to Record Lines Off Top.

Clear Lines Off Top

This command erases any information that scrolls off the top of a CommandShell window.

Redraw Screen

This command redraws all of the pixels that make up the screen. You use this command if, at some point, a window is not completely redrawn.

The CommandShell Preferences menu

The CommandShell Preferences menu allows you to edit some of the properties of a CommandShell window.

Preferences
Notification Levels...
New Window Settings...
Active Window Settings... ⌘E
Save Preferences
Restore From Preferences

Notification Levels

This command displays a dialog box that allows you to control how system messages are displayed when CommandShell is *not* the active application.

New Window Settings

This command displays a dialog box that allows you to set some of the default parameters, such as font and font size, for a new CommandShell window.

Active Window Settings

This command displays a dialog box that allows you to alter some of the properties of the active window, such as the name of the window. The COMMAND-key equivalent is COMMAND-E.

Save Preferences

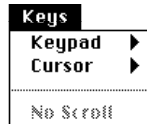
This command saves any changes you have made to a CommandShell window.

Restore From Preferences

This command causes a CommandShell window to revert to the state it was in when you last saved the preferences, if you have made any changes since then. If you have not altered the window, the command is dimmed on the menu.

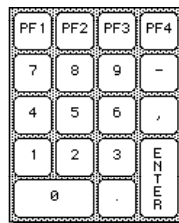
The Keys menu

The Keys menu is a part of the VT102™ terminal emulation, and it allows you to choose from both a keypad and a cursor-movement pad.



Keypad

This command provides an on-screen keypad that you can use with your mouse.



Cursor

This provides an on-screen cursor pad from which you can choose cursor movements with your mouse.



No Scroll

This is a VT102 feature that is not available with A/UX.

Appendix B Network Contacts

This appendix tells you how to contact the organizations that maintain the Internet, CSNET, and BITNET networks.



The Internet

The Internet is managed by the Network Information Center (NIC). You may want to contact NIC to obtain an official network number for your network. Your network needs an official network number if you want to connect the network to the Internet. An official network number is required to prevent duplicate IP addresses, which, when they occur, cause the TCP/IP networking software, upon which the Internet depends, to fail.

If your computer is physically isolated from the Internet and you do not intend to install the telecommunications equipment that is required for connections to the Internet, you do not need an official network number.

In addition, you do not need to get an official network number if another network administrator at your site has already obtained one. Check with the other network administrators to see if you can set your network up as a subnet.

In the past, NIC was managed by SRI International in Menlo Park, California, but it is now managed by Government Systems, Inc. Here is contact information for NIC:

<i>U.S. postal address:</i>	Government Systems, Inc. Attn.: Network Information Center 14200 Park Meadow Drive Suite 200 Chantilly, VA 22021
<i>Help desk telephone numbers:</i>	800-365-3642 (1-800-365-DNIC) 703-802-4535
<i>Help desk hours of operation:</i>	7:00 A.M. to 7:00 P.M. Eastern Standard Time
<i>Fax number:</i>	703-802-8376
<i>Network address:</i>	192.112.36.5 (NIC.DDN.MIL)
<i>Root domain server:</i>	192.112.36.4 (NS.NIC.DDN.MIL)

When you call NIC for a network number, ask to talk to the Hostmaster. Be prepared to tell the Hostmaster the number of computers you plan to have on your network, as this number will influence the class (A, B, or C) that is assigned to your network and how you plan to connect to the Internet.

Sites that are already on the Internet and need information on setting up a domain should contact `HOSTMASTER@NIC.DDN.MIL`. You may also want to be placed on the mailing list for administrators on the Internet who are running the name server. The group discusses future design decisions, operational problems, and other related topics. To request to be placed on this mailing list, contact

`bind-request@ucbarpa.Berkeley.EDU`.

CSNET

A CSNET member organization that has not registered its domain name should contact the CSNET Coordination and Information Center (CIC) for an application and information about setting up a domain.

An organization that already has a registered domain name should keep the CIC informed about how it would like its mail routed. In general, the CSNET relay prefers to send mail by means of CSNET (as opposed to BITNET or the Internet), if possible. For an organization on multiple networks, this may not always be the preferred method. You can reach CIC by electronic mail at `cic@sh.cs.net`, or by phone at 617-497-2777.

BITNET

If you are on the BITNET and need to set up a domain, contact `INFO@BITNIC`.

Appendix C Information on Files in A/UX 3.0.1

This appendix gives information on how to familiarize yourself with the location of A/UX 3.0.1 system files.

Additionally, this appendix lists files and commands that knowledgeable UNIX administrators can review to familiarize themselves with A/UX.



About A/UX files on your disk

A list of the files that make up the A/UX 3.0.1 release can be found in the file `/FILES`, which gives the full pathname of A/UX system files along with a short description. The list is useful when you want to find a short description of a system file.

If you have not installed all the packages from the A/UX release, there will be files listed in `/FILES` that are not present on your system.

A/UX files compared with UNIX files

Here is a list of those commands and setup files found in A/UX that are not part of standard UNIX distributions.

- `/ Applications`
a directory containing applications shipped with the system
- `/ Applications/Retrospect`
a directory containing the Retrospect Remote backup application from Dantz Development Corporation
- `/.aux_prefs`
preferred session type information
- `/bin/diskformat`
binary executable for `diskformat(1M)`—dependent format operation
- `/bin/dp`
binary executable for `dp(1M)`—perform disk partitioning
- `/bin/eject`
binary executable for `eject(1)`—eject floppy disk from drive
- `/bin/xterr`
binary executable for `xterr(1M)`—turn on/off the reporting of extended errors
- `/bin/pname`
binary executable for `pname(1M)`—associate named partitions with device nodes

`/bin/tp`
binary executable for `tp(1)`—manipulate tape archive

`/.cmdshellprefs`
CommandShell preferences data

`/.desk`
directory of Macintosh desktop files, by host name

`/.desk/localhost`
Macintosh desktop file for host named `localhost`

`/dev/appletalk`
directory of devices used for AppleTalk

`/dev/snd`
directory of devices providing A/UX toolbox interface to sound drivers

`/dev/snd/note`
sound driver interface to note synthesizer

`/dev/snd/raw`
raw interface for access to sound driver from CommandShell

`/dev/snd/reset`
reset “switch” for sound manager

`/dev/snd/samp`
sound driver interface to sampled synthesizer

`/dev/snd/wave1`
sound driver interface to wave synthesizer (voice 1)

`/dev/snd/wave2`
sound driver interface to wave synthesizer (voice 1)

`/dev/snd/wave3`
sound driver interface to wave synthesizer (voice 1)

`/dev/snd/wave4`
sound driver interface to wave synthesizer (voice 1)

`/dev/uinter0`
user interface device used by the A/UX Toolbox

`/etc/adduser`
executable shell script for `adduser(1M)`—add a user account

`/etc/apm_getty`
binary executable for `apm_getty(1M)`—set terminal type, modes, speed, and line discipline

`/etc/appleping`
binary executable for `appleping(1M)`—send test packets over the AppleTalk network

`/etc/appletalk`
binary executable for `appletalk(1M)`—configure and view AppleTalk network interfaces

`/etc/appletalkrc`
AppleTalk network configuration file (`appletalkrc(4)`)

`/etc/autoconfig`
binary executable for `autoconfig(1M)`—build a new up-to-date kernel

`/etc/boot.d`
directory of driver object modules used by `autoconfig(1M)` to build new kernels, copied from `/etc/install.d/boot.d`

`/etc/checkinstall`
executable shell script for `checkinstall(1)`—check installation of boards

`/etc/chgnod`
binary executable for `chgnod(1M)`—change current A/UX system nodename

`/etc/cml/descriptions`
description database for all files as shipped (`cml(4)`)

`/etc/cml/masterlist.Z`
database of important information on all files as shipped (`cml(4)`)—compressed

`/etc/dev_kill`
binary executable for `dev_kill(1M)`—removes special device files from directories

`/etc/etheraddr`
binary executable for `etheraddr(1M)`—gets the physical address of an Ethernet address

`/etc/ethers`
NIS data base (`ethers(4)`)

`/etc/ioctl.syscon`
console terminal settings (`ioctl.syscon(4)`)

`/etc/iop`
directory of serial I/O processor code resources

`/etc/kconfig`
binary executable for `kconfig(1M)`—change kernel parameters for tuning

`/etc/keyset`
binary executable for `keyset(1M)`—set console keyboard mapping

`/etc/line_sane`
binary executable for `line_sane(1M)`—push streams line disciplines

`/etc/loginrc`
executable shell script for `loginrc` (start either `/etc/Login` or `/etc/getty`)

`/etc/macgetty`
binary executable for `macgetty(1M)`—put up a Login dialog box on the console

`/etc/macquery`
binary executable for `macquery(1M)`—post a Macintosh alert to query the user

`/etc/macsysinit`
binary executable for `macsysinit` (start the Macintosh environment during startup)

`/etc/macsysinitrc`
executable shell script for `macsysinitrc(1M)`—system initialization shell scripts

`/etc/newconfig`
executable shell script for `newconfig(1M)`—prepare and configure a new kernel

`/etc/newfs`
symbolic link to `/etc/fs/ufs/newfs` (binary executable for `newfs(1M)`)—make a new UFS file system

`/etc/newunix`
executable shell script for `newunix(1M)`—prepare for new kernel configuration

`/etc/powerdown`
binary executable for `powerdown(1M)`—power down the system

`/etc/query`
binary executable for `query(1)`—query the user for input

`/etc/RELEASE_ID`
A/UX release ID number and date

`/etc/sethost`
set the host and domain names (shell commands called from `/etc/sysinitrc`)

`/etc/setport`
binary executable for `setport(1M)`—set a serial port

`/etc/settimezone`
binary executable for `settimezone(1M)`—set the local time zone

`/etc/startmsg`
binary executable for `startmsg(1M)`—send messages to `StartMonitor` during the A/UX startup process

`/etc/tty_add`
executable shell script for `tty_add(1M)`—modify the `/etc/inittab` file

`/etc/tty_kill`
executable shell script for `tty_kill(1M)`—modify the `/etc/inittab` file

`/etc/uninstall.d`
directory of driver removal procedures

/FILES
description database for all files as shipped (cm1(4))

/mac/bin/Apple HD SC Setup
disk-formatting and partitioning application

/mac/bin/Commando
Macintosh Toolbox file for Commando (command dialog-box generator)

/mac/bin/CommandShell
binary executable for CommandShell(1)—A/UX Toolbox application for managing command-interpretation windows and mediating between the desktop and the A/UX environment

/mac/bin/CommandShell24
binary executable for CommandShell24 (24-bit version of CommandShell(1))

/mac/bin/TeachText
Apple TeachText application

/mac/bin/TextEditor
binary executable for TextEditor(1)—Macintosh-style text editor

/mac/bin/changesize
Macintosh Toolbox file for changesize(1)—change the fields of the SIZE resource of a file

/mac/bin/cmdo
binary executable for cmdo(1)—build commands interactively

/mac/bin/derez
binary executable for derez(1) MPW tool to decompile a resource file

/mac/bin/fcnvt
binary executable for fcnvt(1)—convert a resource file to another format

/mac/bin/launch
binary executable for launch(1)—execute a Macintosh binary application

/mac/bin/mac24
executable shell script for mac24 (launch the 24-bit Macintosh environment)

`/mac/bin/mac32`
executable shell script for `mac32` (launch the 32-bit Macintosh environment)

`/mac/bin/rez`
binary executable for `rez(1)` MPW resource compiler—compile resources

`/mac/bin/setfile`
binary executable for `setfile(1)`—set file creator and type

`/mac/bin/startmac`
binary executable for `startmac(1)`—start the Finder environment in A/UX

`/mac/bin/startmac24`
binary executable for `startmac24` (24-bit version of `startmac(1)`)

`/mac/lib`
directory of specialized Macintosh files

`/mac/lib/cmdo`
directory of the Commando dialog-box database

`/mac/lib/SystemFiles/private`
directory of user-writable Macintosh system files

`/mac/lib/SystemFiles/shared`
directory of shared Macintosh system files

`/.mac/localhost/Desktop Folder`
directory where desktop items are put

`/.mac/localhost/Trash`
directory where thrown-away items are put

`/mac/src`
directory of sample Macintosh application sources

`/mac/sys/Login System Folder`
directory of Macintosh initialization files related to logging in

`/mac/sys/Login System Folder/Preferences/Autologin`
a file to configure which user's Finder environment to run upon startup

`/mac/sys/Startup System Folder`
directory of Macintosh initialization files related to system startup

`/mac/sys/System Folder`
directory containing the Macintosh System file, system extensions (INITs), control panels (CDEVs), and so on

`/newunix`
small copy of the kernel, used by `autoconfig(1M)`

`/nextunix`
file containing the name of the preferential kernel to launch

`/root`
alternative home directory for user of `root` account

`/Shared Data`
a directory where file systems are mounted

`/shlib/libmac1_s`
Macintosh shared libraries for code linked under A/UX 2.0.1 and later (run time)

`/shlib/libmac_s`
Macintosh shared libraries for code linked under A/UX 2.0 (run time)

`/System Folder`
an alias to `/mac/sys/System Folder`

`/users`
directory of user accounts

`/users/Guest`
the home directory for the `Guest` account

`/users/start`
the home directory for `start`, a sample account

`/usr/bin/at_cho_prn`
binary executable for `at_cho_prn(1)`—choose a default printer on the AppleTalk network

`/usr/bin/at_printer`
executable shell script for `at_printer(1)`—for backward compatibility with A/UX 1.1

`/usr/bin/atlookup`
binary executable for `atlookup(1)`—look up NVEs registered on the AppleTalk internet

`/usr/bin/atprint`
binary executable for `atprint(1)`—copy data to a remote PAP server

`/usr/bin/atstatus`
binary executable for `atstatus(1)`—get status from a PAP server

`/usr/bin/imagewr`
binary executable for `imagewr(1)`—handle international characters for ImageWriter II

`/usr/bin/isotomac`
binary executable for `isotomac(1)`—convert from Macintosh encoding to ISO encoding

`/usr/bin/iw2`
binary executable for `iw2(1)`—Apple ImageWriter print filter

`/usr/bin/mactois`
binary executable for `mactois(1)`—convert from Macintosh encoding to ISO encoding

`/usr/bin/module_dump`
binary executable for `module_dump(1M)`—dump out information from A/UX kernels

`/usr/bin/sumdir`
binary executable for `sumdir(1)`—sum and count characters in the files in the given directories

`/usr/bin/systemfolder`
executable shell script for `systemfolder(1)`—create a personal 32-bit System Folder

`/usr/bin/systemfolder24`
executable shell script for `systemfolder24(1)`—create a personal 24-bit System Folder

`/usr/lib/terminfo/m/mac2`
terminal capabilities for a Macintosh II

`/usr/lib/terminfo/m/mac2cs`
terminal capabilities for a Macintosh IIcs

`/usr/spool/lp/model/imagewriter2`
lp(1) interface for an ImageWriter II

`/usr/spool/lp/model/iw`
lp(1) interface for an ImageWriter I

`/usr/spool/lp/model/psinterface`
lp(1) interface for Adobe PostScript™/TranScript printer

`/usr/spool/lp/request/iw2`
ImageWriter II output spool file

`/usr/spool/lpd/AppleTalk`
directory of spooled files sent to printer “AppleTalk”

`/usr/spool/lpd/AppleTalk/ifilter`
Berkeley print-spooler filter

`/usr/spool/lpd/AppleTalk/lock`
Berkeley print-spooler lock file

`/usr/spool/lpd/AppleTalk/nfilter`
Berkeley print spooler nroff filter

`/usr/spool/lpd/AppleTalk/ofilter`
AppleTalk printing output filter

`/usr/spool/lpd/AppleTalk/pipe`
named pipe, used for AppleTalk spooler filter

`/usr/spool/lpd/ImageWriter`
directory of spooled files sent to printer “ImageWriter”

`/usr/spool/lpd/ImageWriter/lock`
Berkeley print-spooler lock file

`/usr/spool/lpd/LaserWriter`
directory of spooled files sent to printer “LaserWriter”

`/usr/spool/lpd/LaserWriter/lock`
Berkeley print-spooler lock file

Glossary

AppleTalk A set of networking protocols that works across LocalTalk (serial) and EtherTalk (Ethernet) networks.

argument A piece of information included on the command line in addition to the command; it specifies what the command should act upon. The shell passes this information to the command, which then modifies its execution in some particular way. Filenames, for example, are often supplied as arguments to commands, so that a command operates on the named file.

ARPA Acronym for *Advanced Projects Research Agency*, formerly known as the *Defense Advanced Projects Research Agency (DARPA)*, which is the agency that funded the development of TCP/IP. See also *TCP/IP*.

ASCII Acronym for *American Standard Code for Information Interchange*. A set of numbers representing alphanumeric characters used in text representation.

A/UX Startup The Macintosh application that launches A/UX. Under most circumstances, if you cancel the automatic launch of A/UX, you are in the A/UX Startup shell, from which you can run some system administration commands that modify files located in A/UX file systems.

baud rate The number of bits per second that can be sent over a serial line.

Berkeley Software Distribution See *BSD*.

block devices Devices that access data blocks; the ability to access data blocks allows the device to contain mountable file systems. Reading from and writing to block devices are handled through a cache of buffers that minimize physical access to the device.

Bourne shell The standard UNIX System V command interpreter, developed by Steven Bourne at AT&T. See also *shell*.

break A sequence containing all null characters sent for a specified period of time over a serial line. Used as a signal.

broadcast address An address understood by all hosts on a local network as a special address for broadcast packets. See also *host*, *Internet*.

BSD (Berkeley Software Distribution) Releases of the UNIX operating system made by the Computer Systems Research Group at the University of California at Berkeley.

BSD file system The format of the UNIX file system developed by the Computer Systems Research Group at the University of California at Berkeley. This format supports long filenames (255 characters) and symbolic links. The Berkeley UNIX file system is also known as *UFS*. See also *SVFS*, *UFS*.

cache A region of memory used to store information that is likely to be accessed soon.

case sensitive Able to distinguish between uppercase and lowercase characters. Programming languages are case sensitive if they require all uppercase letters, all lowercase letters, or proper use of uppercase and lowercase letters.

character devices In general, devices that perform I/O asynchronously for a variable number of bytes.

checksum An integer value computed from the data to be transmitted and appended to the data when it is transmitted. When the data is received, the checksum is recomputed and matched against the transmitted checksum to detect a transmission error.

client Depending on the context, a system or process that uses the resources provided to the network by a server.

command option An argument included on the command line that instructs a program to alter its output or to change its mode of execution. A command option is usually a hyphen followed by one or more characters. For example, the `-l` option to the `ls` command makes this utility display extra information, such as the date a file was last saved. Command options are sometimes referred to as *keyletters*, *flag options*, or *options*.

CommandShell An A/UX application that displays a window for using the command-line interface. You can manipulate CommandShell windows just as you do Macintosh windows.

Compressed Serial Line/Internet Protocol See *CSL/IP*.

C shell The standard BSD command interpreter and one of three shells included in A/UX. See also *shell*.

CSL/IP Acronym for *Compressed Serial Line/Internet Protocol*. The Internet protocol that allows a serial line to connect a computer to a TCP/IP network. This version of the protocol supports data compression, which was not supported by its precursor, SL/IP. See also *TCP/IP*.

daemon A process that, when started, runs continuously in the background. A daemon process typically provides a service, the need for which is unpredictable and intermittent.

DARPA See *ARPA*.

desktop The screen display provided by Macintosh graphical user interface. The display represents a surface on which objects such as volumes and applications appear as icons, and workspaces are represented by windows.

device driver A module that is included in the kernel to provide the interface between software (such as a UNIX command or a Macintosh application) and a device (such as a serial port). See also *kernel*.

directory hierarchy The collection of all files currently available to the system; that is, all files on currently mounted file systems.

driver See *device driver*.

Ethernet A hardware medium designed to provide high-speed interconnections between computers. The original Ethernet cable was a 1/2-inch coaxial cable with heavy shielding. Thinner coaxial Ethernet cables and shielded twisted-pair Ethernet cables are now available.

EtherTalk Apple Computer's data-link product that allows an AppleTalk network to be connected by Ethernet cables. See also *AppleTalk*, *Ethernet*.

file system The logical organization of data in a storage medium, such as a partition on a hard disk drive. See also *BSD file system*, *HFS*, *SVFS*.

GID Acronym for *group ID*. A number in the `/etc/group` file that represents a user's membership in a group of users.

HFS Acronym for *hierarchical file system*. A method of organization in the Macintosh Operating System in which disk files are grouped together within directories and subdirectories (folders within folders). HFS is used on hard disks and on floppy disks. In a

hierarchical file system, a file is uniquely specified by its pathname rather than by a single filename.

host A computer on a TCP/IP network; often called a *local host* or a *remote host*.

host name The name of a computer on a TCP/IP network.

host number The portion of an IP address that uniquely identifies a computer on a TCP/IP network.

inode A file that contains pertinent information about a UNIX file, including its owner, permissions, modification times, and the disk location of the first few data blocks of the file.

internet A group of interconnected networks that use protocols that allow the networks to function as a single large network. Compare *Internet*.

Internet A group of networks, including the ARPANET, MILNET, and NSFnet, interconnected by computers called *Internet routers*. Because the Internet spans several networks, the routing of data might involve several intermediate nodes on the way to the destination node. See also *Internet router*.

Internet domain A group of computers, organized in a hierarchy, on the Internet. See also *Internet*.

Internet forwarder A computer that has two or more Ethernet cards, each connected to a different network, and that routes packets between the networks. See also *Internet*.

Internet network number See *network number*.

Internet Protocol The Internet protocol that defines the Internet datagram as the unit of information that is sent over the network. The Internet datagram provides the basis of the Internet connectionless, best-effort packet-delivery service. See also *Internet*.

Internet Protocol address See *IP address*.

Internet router A computer that maintains routing tables used to route network traffic in the most efficient way. See also *Internet*.

inumber Contraction of *index-number*. A number that acts as an index to the superblock that locates an inode. An inumber is effectively the filename of an inode.

IP address A 4-byte number, consisting of a network number and a unique host number, that identifies each computer on a network. See also *host number*, *network number*.

kernel That portion of an operating system that, when running, always resides in memory.

Korn shell A command interpreter that combines many of the best features found in the standard System V shell (the Bourne shell) and the standard BSD shell (the C shell). See also *shell*.

LocalTalk A form of serial connection between computers and devices, such as printers.

loopback interface A module in the kernel that, when running, always allows you to test the TCP/IP networking software locally without actually accessing the network. See also *kernel*, *TCP/IP*.

major number A number that indicates to a UNIX kernel what device driver to use in communicating with a device file.

minor number A number that indicates to a UNIX kernel what subdevice or set of device features to use in communicating with a device file.

mount To make a file system available for use. A local file system is mounted if its device file (such as `/dev/dsk/c0d0s2`) is associated in the kernel with a directory (such as `/usr`). A remote file system is mounted through the NFS software. See also *kernel*.

multi-user mode The state of the UNIX operating system in which more than one user can log in at the same time. When you restart A/UX, the state that A/UX automatically enters is determined by the value of the *rstate* field in the *is* entry in the `/etc/inittab` file. A value of 2 causes A/UX to restart to multi-user mode. See also *single-user mode*.

netmask A 32-bit value. The bits in the netmask that are set to 1 define the portion of the IP address that is the network number, and the bits that are set to 0 define the portion of the IP address that is the host number. See also *host number*, *IP address*, *network number*.

Network Information Center See *NIC*.

network number A 1-byte, 2-byte, or 3-byte number that identifies a network. All hosts on that network use the same network number. To connect to the Internet, you need an official network number. You can obtain an official network number from the Network Information Center; contact information is in Appendix B. See also *host*, *Internet*, *NIC*.

NIC Acronym for *Network Information Center*. An organization that coordinates the development of many of the network applications that work on TCP/IP networks. See also *TCP/IP*.

option See *command option*.

permissions Authorization to read, write, or execute a file or directory. On implementations of the UNIX operating system, each capability is assigned on an individual, a group, and a systemwide (“others”) basis. Also called the *file mode*.

print spooler A utility that writes a representation of a document’s printed image to disk or to memory, schedules it to print in a queue of other jobs, and then prints it.

protocol A well-known set of conventions (for representing data, checking it, transmitting it, and so forth) that must be implemented at both ends of a connection before communications can take place.

root (1) The top directory in a UNIX directory hierarchy. Represented by a slash (/), it is the first element in every absolute pathname. (2) The user with unlimited system privileges. Also called the *superuser*.

root directory See *root* (1).

router A computer that is physically connected to two or more networks and directs internetwork traffic to the appropriate network so that it arrives at its destination by the most efficient route.

serial Describes a hardware medium that provides slow-speed interconnections between computers and devices.

server Depending on the context, a system or process that provides resources to a network.

shell A utility that accepts your commands, interprets them, and passes them on to the appropriate programs for execution. A/UX provides three shells: Bourne, C, and Korn. Each can be used as an interpreted programming language. Through shell variables and environment variables, you can tailor the environment of a shell to your own needs.

single-user mode The state of the UNIX operating system in which only one user at a time has access to the system. That access is normally available only at the System Console window. When you restart A/UX, the state that A/UX automatically enters is determined by the value of the *rstate* field in the *is* entry in the */etc/inittab* file. A value of *s* causes A/UX to restart to single-user mode. See also *multi-user mode*.

spool To place a job in a queue; placing a printing job in the print spooler is an example.

subnet An extension of the Internet addressing scheme that allows a site to use a single network number for multiple physical networks. See also *Internet*, *network number*, *NIC*.

superblock A software structure that maintains information about the status of a UNIX file system. This includes, but is not limited to, the allocation of free and used data blocks, inode allocation, and the last time the file-system state was copied to disk.

SVFS Acronym for *System V file system*. A kind of UNIX file system, characterized by having a single superblock for each disk partition. See also *HFS*, *superblock*, *UFS*.

System Console window A CommandShell window designated to receive system status messages. Equivalent to `/dev/console` in UNIX implementations other than A/UX.

TCP/IP Acronym for *Transmission Control Protocol/Internet Protocol*. A suite of networking protocols developed for the U.S. Department of Defense that specifies the details of how computers communicate. Also called the *Internet protocols*.

TCP/IP networking software Networking software based on the Transmission Control Protocol/Internet Protocol (TCP/IP). See also *TCP/IP*.

UFS A kind of UNIX file system, characterized by having multiple superblocks for each disk partition. See also *HFS*, *superblock*, *SVFS*.

UID Acronym for *user ID*. A number in the `/etc/passwd` file that uniquely identifies the user on a computer.

Index

- * (asterisk) 94
 - \ (backslash) 94, 145
 - [] (brackets) xxiii
 - : (colon) 159
 - . (dot). *See* dot files
 - ... (ellipsis) xxiii
 - \f (form feed) 145
 - (hyphen) 30, 94, 223
 - # (number sign) 119
 - % (percent) 94
 - | (pipe) 37–39
 - / (slash) 22–24, 159
 - _ (underscore) 159
- A**
- abort option 148
 - absolute pathname 22–24
 - access
 - control of, to printer queues 146–147
 - to `crontab` command 93
 - permissions 164, 177–185
 - `access-class` field 179
 - access time 219
 - accounts
 - administrative 162–164
 - deleting 172–173
 - user 165–173
 - voiding 171
 - active area of window 6
 - Active Window Settings menu command 12, 273
 - ADB port. *See* Apple Desktop Bus port
 - addresses, disk 217–219
 - `adduser` command 2, 169–171
 - adm account 163
 - Apple CD-ROM drives 79–87
 - Apple-compatible printers 138
 - Apple Desktop Bus (ADB) port 140
 - AppleShare network server
 - A/UX and 53
 - users and groups 155
 - AppleTalk Internet Router software 122
 - AppleTalk network, printing and 152–153
 - AppleTalk printers 143–144
 - Apple tape cartridges 73
 - archives, compressing 92
 - arguments 31
 - “Arguments” section 36
 - audio CDs 80
 - `autoconfig` command 51–52, 60
 - AutoLaunch command 47–48
 - Autologin file 56–57
 - AutoRecovery command 47–48
 - A/UX Startup
 - A/UX System Console window 43
 - canceling 43
 - commands in 45
 - definition 42
 - entering 43–52
 - initial processes 57–59
 - menus 46–49
 - shell window 44
 - single-user mode 56
-

A/UX Startup (*continued*)
 startup sequence 50–52
 trouble with 189–191
A/UX System Console menu command 10, 271
A/UX System Console window 43
 clearing 10
 closing 10
 displaying 10
A/UX working environment 156–162

B

backing up
 definition 67
 hard disk errors and 193–194
 media for 69–70
 schedule for 74
backslash (\) 145
bad sectors 193–194
Balloon Help program xxi
baud rate 145
b character 223
Berkeley UNIX File System (UFS) 215
 superblock and 221
Berkeley Internet Name Domain (BIND) software
 122
binary files 90–91
 /bin/csh 161
BIND. *See* Berkeley Internet Name Domain
bind-request@ucbarpa.Berkeley.EDU 277
 /bin/ksh 162
 /bin/sh 161
BITNET 277
bit setting 182
block-device argument 76
block devices 71, 223
block I/O 221–225
blocks 215, 217–219
 checking 227, 240–243
 duplicate 240–243, 246–247
 error messages and 240–243
 maps of free 258–259
boot command 47
Booting dialog box 46

Boot menu command 47
Bourne shell 2, 16, 158
 setup files and 161
brackets [] xxiii
br entry 145
broadcast address
 determining 104–105
 /etc/NETADDRS file and 113–114
 subnets and 132
buffer cache 222
 changing size of 62–63
 new kernel and 62
bus, changing 77–78
bytes 215

C

Cancel button 50
 A/UX Startup and 43
carriage returns 159
case sensitivity 30, 159
ccat program 91
c character 223
cd command 27–28
CD-ROM 79–87
 backing up and 68
 installing discs 81
 permanently mounted discs 81, 87
 removing discs 86–87
 temporarily mounted discs 82, 87
Change Password dialog box 197
character devices 71, 223
Checking root file system startup screen 51
Checking startup screen 50
chgrp command 205
child processes 57
chmod command 179–185
chown command 205
chroot command 192
chsh command 16, 173
CIC. *See* Coordination and Information Center
 cic@sh.cs.net 277
cleanup of file-system check 260
clear command 6

Clear Lines Off Top menu command 7, 10, 272
 Clear menu command 265
 Clipboard 17, 265
 Close All Windows menu command 8, 264
 Close menu command 7, 263
 colon in filename (:) 159
 command-line prompt 44, 160
 command notation xxiii–xxiv
 Commando menu command 265
 commands
 in A/UX Startup 45
 cron 95–96
 halting 47
 linking with pipe 37–39
 lpr 147–149
 UNIX 26–28
 CommandShell application
 accessing 262
 activating 2–3
 defined 1–2
 leaving 19
 opening file on CD-ROM and 83
 quitting 262
 switching to Finder 3
 CommandShell menu command 10
 CommandShell windows 3, 262
 alerts 8, 11
 arranging 4–5
 closing 7
 customizing 12–15
 displaying 4–5
 printing contents of 9
 recording 5–7
 saving contents 18
 saving layout of 14–15
 Commands menu 273
 compact disc read-only memory. *See* CD-ROM
 compact program 90–91
 compression programs 90–91
 compress program 90–91
 configuration files 127
 configuration parameters, network 112–114
 connectivity, checking 227, 252–254
 Console Emulator window 160
 controller, device name and 72
 Coordination and Information Center (CIC) 277
 -c option 30, 152
 copying text 17
 Copy menu command 17, 265
 core files 88
 Courier font xxi–xxii
 cp command 84
 cpio
 compressing archive 92
 defragmenting and 231
 error messages 205
 files 86
 reinitializing and 193
 CR character 159
 cron command 92–96
 crontab file 93–94
 cs entry 145
 C shell 2, 16, 158
 setup files and 161
 .cshrc file 161
 CSNET 277
 current directory 24, 158
 finding 27
 Cursor menu command 268
 customizing
 environments 156, 162
 system 53–54
 Cut menu command 259
 cylinder groups 221
 checking 227, 258–259

D

daemon account 163
 daemons, disabling 119
 data blocks, fsck program and 226
 DAT drive. *See* DDS-DC DAT tape drive
 date command 54
 day-of-month field 93
 day-of-week field 93
 d character 223

- DDS-DC DAT tape drive 70
 - formatting 73–74
- decompression programs 91
- defaults
 - 1000-line 5
 - file permissions 184–185
 - port 145
 - printer 144
 - print spooler 143
 - run levels 57
 - setup files 160
 - window 12–13
- defragmenting 231–232
- DELETE key 265
- “Description” section 36
- destination, printer 141
- /dev directory 72–73, 222–225
- device drivers 139, 224
 - A/UX Startup and 51
- device files 71–73, 222
 - definition 141
- device names 140
 - partition numbers and 72
- diamond-shaped symbol 11
- direct data block 217–219
- directories 22
 - changing 27–28, 48
 - of CommandShell window 3
 - current 24, 27, 158
 - error messages and 244–251
 - home. *See* home directory
 - listing contents 28–31
 - ownership of 176
 - permissions 178–179
 - reconnecting 254
 - spooling 146
 - unreferenced 252–254
- directory data blocks 226
- direct printing 138
- “dirty file system” 44
- disable option 148
- disk addresses 217–219
- disk errors 193–194
- disk partitions 202–203, 223
 - backing up and 69
- disk space 88–92
- documentation, on-line xxi
- domain name 52–53
 - changing 113
 - simple network and 108
- Don't Record Lines Off Top menu command 7, 272
- dot files (. and ..) 158, 226
 - error messages and 244–251
- double indirect block 219
- down option 149
- dragging 4–5
 - backing up and 70
- drive, device name and 72
- dump.bsd utility, reinitializing and 193

E

- ed editor 192
- editing, with Macintosh tools 16–18
- Edit menu 264–265
- ellipsis points (...) xxiii
- emergencies 194
- enable option 148
- environment variables 158–159
- error messages 188–189
 - fsck program 234–259
 - group administration 196–197
 - kernel parameters and 63–64
 - ping command and 134
 - quitting and 19
 - in TCP/IP 111–112, 209–211
 - unmounting and 77
 - user administration 196–197
 - /etc/bcheckrc script 58
 - /etc/brc script 52, 58
 - /etc/fstab file 77–79
 - backing up 69
 - fields in 232–233
 - fsck program and 231
 - /etc/ftpusers file 116
 - /etc/getty script 58

- /etc/group file 175
 - backing up 69
- /etc/HOSTNAME file 52
 - modifying 113
 - system naming and 53
- /etc/hosts.equiv file 120–121
- /etc/hosts file
 - backing up 69
 - complex networks and 126–127
 - modifying 115
 - subnet configuration and 131
- /etc/inittab file 52, 57
 - backing up 69
 - modifying 114–115
- /etc/NETADDRS file
 - modifying 113
 - subnet configuration and 131
- /etc/networks file
 - complex networks and 127
 - modifying 116
- /etc/passwd file 165–166
 - backing up 69
- /etc/printcap file
 - modifying 117
 - nondefault printers and 150–151
 - printing and 143
- /etc/protocols file 117
- /etc/rc script 52, 58
- /etc/servers file 117
- /etc/services file 118
- /etc/shells file 118
- /etc/sysinitrc script 51, 57
- Ethernet
 - complex network and 121–125
 - multiple cables 127–132
 - simple network and 106–107
- “Examples” section 36
- exec option 88
- executable file 26
- execute permission 177–185
 - error messages and 196–197
- Extensions folder 191

F

- fd entry 145
- file: table is full error message 204
- file access, inode and 219–220
- File menu 263–264
- filename field 180
- file modification time 219
- files
 - binary 90–91
 - compressing 90–92
 - configuration 127
 - copying from CD-ROM 83
 - copying from cpio archive 86
 - copying from tar archive 84–86
 - crontab 93–94
 - device. *See* device files
 - dot. *See* dot files
 - executable 26
 - information and fsck 225
 - log 89–90
 - opening on CD-ROM 82–83
 - permissions 177–185
 - printing 152–154, 198
 - removing unneeded 88–89
 - setup. *See* setup files
 - special 222
 - trimming 89–90
 - in UNIX file system 84
 - unreferenced 255–258
- /FILES file 281
- “Files” section 36
- file systems
 - accessing from A/UX Startup 192–193
 - check program. *See* fsck program
 - “dirty” 44
 - fragmentation of 231–232
 - identifying on CD-ROM 79–80
 - interface 215
 - mounting with UNIX commands 76
 - multiple, fsck program and 231–233
 - reclaiming disk space 88–92
 - troubleshooting 202–203
 - UNIX 22

- file systems (*continued*)
 - unmounting 77
 - updates of 225–227
 - file type characters 178
 - find command 88
 - Finder program
 - activating 262
 - multi-user mode and 56–57
 - opening files on CD-ROM and 82–83
 - printing and 137
 - single-user mode and 55
 - switching to CommandShell 3
 - finger command 168
 - fingerd daemon 119
 - flag setting 182
 - floppy disks
 - A/UX Startup and 47
 - backing up and 69–70
 - non-Macintosh format 96–97
 - flow control 145
 - folders 3, 24
 - permissions 178–179
 - fonts, changing 17–18
 - Fonts menu 17–18, 271–272
 - form feed 145
 - forwarding 122–127
 - fragmentation 230–231
 - free-block maps 258–259
 - free list 227
 - fsck program
 - A/UX Startup and 44, 50
 - command syntax 229
 - error messages 234–259
 - file system problems and 202
 - how it works 225–228
 - introduction to 214
 - multiple file systems and 231–233
 - phases of 227–228
 - reinitializing and 193
 - sample 230–231
 - using 227–233
 - fsentry command 76
 - ftp users 116
 - Full Height menu command 270
- G**
- General Controls control panel 54
 - General menu command 48
 - GID. *See* group identification number
 - GID field 166, 175
 - Government Systems, Inc. 276
 - graceful shutdown 195
 - grep command 32
 - group identification number (GIO) 157, 166, 174
 - error messages and 195
 - selecting 168
 - group memberships
 - multiple 176
 - permissions and 179
 - group-name field 175
 - group names 174
 - groups
 - administering 174–176, 195–197
 - administrative 164
 - creating 174–176
 - initial 170
 - Guest account 162
- H**
- handshaking 145
 - hard disks
 - bad sectors 193–194
 - partitions 202–203, 223
 - removing 78
 - starting A/UX from 48
 - “hard read” errors 229
 - hardware
 - A/UX Startup and 51
 - debugging 206–207
 - kernel configuration and 60
 - swapping 207
 - “hard write” errors 229
 - HD SC Setup 193, 202–203
 - help facility xxi, 45

- hidden devices, accessing 42
- Hide CommandShell menu command 19
- Hide “Window Name” menu command 270
- hierarchical file systems 215. *See also* Macintosh file systems
- home directory 157
 - error messages and 197
 - moving 173
 - selecting 168
- home-directory* field 166
- \$HOME/.profile file 161–162
- \$HOME/.rhosts file 121
- HOME variable 48
- h option 152
- host equivalence 120
- Hostmaster 277
- host name 52–53
 - changing 113
 - and computers running 133
 - remote 120–121
 - simple network and 108
- host number 103
 - subnet configuration and 128
- hour* field 93
- hyphen (-) 30, 94, 223

I, J

- icons
 - blank 204
 - flashing 11
- if filter 151–152
- ImageWriter II printers 143
- in-core blocks 226
- in-core memory. *See* memory, main
- indirect blocks 217–219
 - fsck program and 226
- inetd program 117–118
- INFO@BITNIC 277
- INIT file, trouble with new 191
- initialization 235–240
- Initializing device drivers startup screen 52
- init process 57–59

- inode list 240–243
- inode modification time 219
- inodes 216
 - fsck program and 226
 - information contained in 219–221
 - maps of used 258–259
- inode table 221
- interface-ID* value 113–114
- interface program, definition 141
- International Standards Organization (ISO) file systems 79–80
- Internet address. *See* IP address
- Internet network 101–102, 276–277
 - complex network and 121–125
- IP address
 - determining if correct 135
 - establishing 101–104
 - /etc/NETADDRS file and 113–114
 - NIC and 276–277
 - simple network and 108
 - subnet configuration and 128–130
- ISO. *See* International Standards Organization
- Italic font xxii

K

- kconfig command 62
 - kernel parameters and 64–65
 - kernel problems and 204
- kernels
 - A/UX Startup and 50–52
 - buffer cache 222
 - changing parameters of 63–65
 - complex network and 125–126
 - configuring 59–65
 - device drivers and 139
 - error messages 204
 - simple network and 106–107
- keyboard, nonresponsive 190
- Keypad menu command 274
- Keys menu 274
- kill command 59
- Kill menu command 47

known parameter 215
Korn shell 2, 16, 158
 setup files and 162

L

LAN. *See* local area network
LaserWriter printers 143
launch command 47–48, 56
Launching startup screen 51
length argument 151–152
line feeds (LF) 159
link counts 255–258
list field 175
Loading startup screen 51
local area network (LAN) 101
localdomain name 52
localhost name 52–53, 101
local printing 138
LocalTalk network 140
log files 89–90
logging in, on another computer 112
logging out 65
logical block 215
login accounts 163–164
Login dialog box, frozen 190
login name 156–157
 and current users 133
 selecting 168
login-name field 121, 165
login-program field 166
login scripts 160–162
login shells 118
loopback interface 110
-l option 30, 152
/lost+found/* files 89
lost+found directory 227, 252–254
 missing or full 255–258
lp account 164
lpc program 148–149, 201
lpd program 201
lp entry 145
lpq command 147, 153
 error messages 200–201

lpr command 137, 147, 153
 error messages 199
lprm command 147, 154
 error messages 201
lpr print spooler 117
 managing 141–152
 setting up 143–147
 troubleshooting 198–201
ls command 28–31

M

Macintosh applications, printing and 137
Macintosh file systems 79–80, 215
MacPartition 45
MacsBug program 194
magic number 221, 259
major number 224
man command 34, 90
man pages. *See* on-line manual pages
manually arranged CommandShell windows 4
manual positions of windows 4
Media Recognition System (MRS) tapes 74
memory
 checking parity 49
 fsck program and 225, 235
 main 221–222
 new kernel and 62–63
messages 11. *See also* error messages
 system 9–10
Messages button 43, 50
metacharacters 101
m_expand returning 0 error message 204
minor number 224
minus sign (–) 30, 94, 223
minute field 93
misc field 166
modification time 219
module_dump command 59, 61
modules 59–61
month-of-year field 93
mount command 78
mounting file systems 76
mount-point argument 76–77

MRS (Media Recognition System) tapes 74
multi-user mode 50, 56–57
mv command 173

N

names
 changing, of mounted volume 78–79
 choosing host 101
 default 52
 device 140
 device file 71
 error messages and 196
 extensions on compressed files 92
 group 174
 pin 140
 printer 144
 system 53
“Name” section 35
naming scheme, SCSI 42, 73
NBUF variable 62
netmask 105
 complex network and 125
 /etc/NETADDRS file and 113–114
 simple network and 108
 subnet configuration and 127–130
netstat command
 hardware and 206
 TCP/IP and 207–209
Network File System (NFS) 100
Network Information Center (NIC) 102, 276–277
Network Information Service (NIS) 100
network-name field 116
network number
 complex network and 121
 establishing 101–104
 NIC and 276–277
 subnet configuration and 128–130
network printing 138
networks
 analyzing traffic on 207–209
 changing configuration parameters 112–114
 classes of 102–103, 128–129
 creating simple 105–112

 designing complex 121–132
 determining status 132–135
 LocalTalk 140
 multiple Ethernet cables and 127–132
 resources 100
 security issues 118–121
 troubleshooting 206–211
newconfig command 53
 broadcast address and 104
 complex network and 125
 device drivers and 139
 kernel configuration and 60–61
 simple network and 106–107
 TCP/IP and 101
newgrp command 176
New menu command 4, 263
newunix program 60
New Window Settings menu command 13, 273
NFILE parameter 204
NFS. *See* Network File System
NIC. *See* Network Information Center
NIS. *See* Network Information Service
nobody account 164
-n option 32, 152
No Scroll menu command 274
Notification Levels menu command 11, 273
NSTREAM parameter 211
NuBus cards, kernel configuration and 59–60
number sign (#) 119
numeric terms, chmod command 181–182
nuucp account 163

O

oF filter 146, 151–152
on-line documentation xxi
on-line manual pages 21
 displaying 34
 quitting 37
Open menu command 263
operator field 180
OPTION key 264
options 28–31
 for ls command 35

- os entry 145
- output 27
 - redirecting 32–33
- output filters 146
 - writing 151–152
- output flag bits 145

P

- pack program 90–91
- Page Setup menu command 264
- paging systems 184
- panic error messages 189
- paper records 69
- parity memory 49
- partition number, device name and 72
- partitions, disk. *See* disk partitions
- pass-number* field 232
- passwd command 182
- password 156–157
 - A/UX Startup and 42, 49
 - blank field 172
 - error messages and 195, 197–199
 - remote users and 120–121
- password* field 166, 175
- Paste menu command 17, 265
- pathnames
 - checking 227, 244–251
 - UNIX 22–25
- pax archive, compressing 92
- pcat program 91
- peripheral devices
 - drivers for 139
 - filenames of 71
 - kernel configuration and 60
- permissions 156–157, 177–185
 - access 164
 - b character 223
 - backing up and 70
 - c character 223
 - command 182–184
 - fsck program and 237
 - setuid 182
- permissions* field 180, 223

- physical block 215
- ping command 110–112, 134–135
- pin names 140
- pipe (|) 37–39
- plain data blocks 226
- portmap daemon 115
- ports 140
 - default 145
- power failure 194
- power switch 44
- Preferences menu 273
- Previous Window menu command 270
- primary memory. *See* memory, main
- printcap database 143–147
- Print dialog box 9
- printer connections 138
- printer icon 140
- printers
 - Apple-compatible 138
 - characteristics 144–145
 - names 144
 - setting up nondefault 150–151
- printer* variable 141
- printing 152–154
 - canceling 154
 - direct 138
 - Finder and 137
 - local and remote 138
 - troubleshooting 198
 - window contents 9
- printing queue 142, 153–154
 - access control and 146–147
- print option 88
- Print Selection menu command 9, 264
- proc: table is full error message 204
- programmer's switch 44
- prompt, command-line 44, 160
- pwd command 26–27

Q

- question mark icon, blinking 190
- Quit menu command 19, 264

R

RAM. *See* memory, main
raw devices. *See* character devices
r`cp` command 109
read permission 177–185
reboot switch 44
reconnection 254
recorded area of window 6
redirection 27, 32–33
Redraw Screen menu command 272
redundancy in file systems 215
reference counts, checking 227, 255–258
reference manuals xxiv–xxv, 21, 34–37
reinitializing 193–194
relative pathname 24–25
remote computers
 host names of 120
 ping command and 134–135
 printers 146
 TCP/IP and 100
remote-host field 120–121
remote printing 138
remsh command 109
remshd daemon 119
Repair button 202
request, printer 141
resizing 4–5
responses to error messages 234–235
restarting 65
Restart menu command 65
restart option 149
Restore From Preferences menu command 15, 273
Retrospect Remote 67
 limitations of 68–69
RETURN key 35
rexecd daemon 119
rg variable 146
rlogin command 109–110, 120–121
rlogind daemon 119
rm variable 146
root account xix, 162–163
 A/UX Startup and 42, 49
root directory 48, 238

root file system, checking 51
root inode, error messages and 246, 250–251
root privileges, single-user mode and 55
ROOT variable 48
Rotate Windows menu command 271
routed daemon 115, 122
 malfunction of 209
routing 122–127
 errors in 209
rp variable 146
run levels 55–57
 changing 58–59
 default 57
ruptime command 133
rwho command 133
rwhod daemon 115, 133

S

Save Preferences menu command 14–15, 273
Save Selection menu command 18, 263
scanners 73
scheduler, print spooler 141
screen lock 194
screens, setting appearance of 14
scrolling, stopping 205
SCSI buses 42, 73
 naming scheme 42
 number of 73
SCSI identification number
 A/UX Startup and 48
 changing 77–78
sd variable 146
searching 32–33
security 118–121, 156
“See Also” section 36
Select All menu command 265
selecting text 16
sendmail daemon 115
serial ports 140
server administration, automating 92–96
server administrator xix
set group identification (setgid) 182
settimezone command 54

- setup files 158, 281–290
 - shells and 160–162
- set user identification (setuid) 182
- /Shared Data folder 78, 287
- sharing 156
- shells 2, 158
 - changing 16, 173
 - error messages and 196–197
 - selecting 169
 - setup files and 160–162
- Show All Windows menu command 270
- Shut Down menu command 65
- shutting down 195
- SIMMs, 8-bit 49
- single-user mode 55–56
- slash (/) 22–24, 159
- slices 81
- sockets, status of 209
- software
 - kernel configuration and 60–61
 - networking 100
 - testing network 110
- SPACE bar 35
- spaces
 - in command notation xxiv
 - in UNIX commands 28
- special files 222
- Special menu, shutting down and 65
- spooler. *See* lpr print spooler
- SRI International 276
- standard error 151
- standard output 27
- standard positions of windows 4
- Standard Positions menu command 269–270
- Standard Size menu command 270
- start account 26, 162
- Starting background processes startup screen 52
- start option 148
- startup# prompt 44
- startup application, definition 42
- startup screen 43
- status option 149
- sticky bit 184
- stop option 149
- stream heads 211
- subdrives 73
- subnet number 121
- subnets 103
 - configuring 127–132
- su command 95
- superblock 221
 - error messages and 238–240
 - fsck program and 225–226
- SVFS. *See* System V file system
- swapping systems 184
- symbolic terms, chmod command 179–180
- “Synopsis” section 35
- sys account 163
- System Console window. *See* A/UX System Console window
- system crash 188–189
- system extension, trouble with new 191
- system files, locations of 281–290
- System Folders 158
 - customizing and 162
 - personal 170
- system name 53
- System V file system (SVFS) 215

T

- tape devices
 - backup and 70
 - cleaning 74
 - error messages and 205
 - formatting cartridge 73–74
 - naming scheme of 73
- tar archive
 - compressing 92
 - files 84–86
- tar command 205
- TBFMIGNORECASE variable 159
- TBTRANSLATESLASHES variable 159
- TBTRANSLATEUXONLY variable 159
- TCP/IP network
 - administering 132–135
 - analyzing problems with 207–209

- configuring interface to 107
- error messages 111–112, 209–211
- overview 100
- security issues 118–121
- telephone handset icon 140
- telnet command 112, 135
- terminators, checking 206–207
- text, editing 16–18
- tftpd daemon 119
- tiled windows 4
- Tile Horizontal menu command 268
- Tile menu command 267
- Tile Vertical menu command 268–269
- time of file access 219
- time zones 54
- tmp directory 220
- transceivers, checking 206
- Transmission Control Protocol/Internet Protocol. *See* TCP/IP network
- triple indirect block 219
- troubleshooting 42
 - basic guidelines 188
- type* field 76, 232

U

- UFS. *See* Berkeley UNIX File System
- UID. *See* user identification number
- UID field 166
- umask command 184–185
- umount command 77, 87
 - changing names and 78
 - file-system problems and 203
- uncompact program 91
- uncompress program 91
- underscore (_) 159
- Undo menu command 265
- unhappy face icon 190
- UNIX file systems, Apple CD-ROM drive and 79–80
- unmounting
 - error messages and 203
 - file systems 77
- unpack program 91
- up option 149

- used-inode maps 258–259
- user accounts 165–173
- user administration, troubleshooting 195–197
- user identification number 157, 166, 174
 - error messages and 195
 - selecting 168
- user modes 55–56
- users
 - adding 167–171
 - removing 171–173
 - types of 156–157
- /usr/lib/skel directory 160
- /usr/preserve/* files 89
- utilities
 - for backing up 68
 - database 69
- uucp account 163

V

- variables, environment 158–159
- verbose mode 189
- vertical line (|) 37–39
- VOID 171
- VT102 terminal emulation 274

W, X, Y

- WAN. *See* wide area network
- who -r command 56
- wide area network (WAN) 101
- width* argument 151–152
- wildcard character 179
- Window menu 266–271
- windows. *See also* CommandShell windows; System Console window
 - appearance of new 13–14
 - arranging 4–5
 - default settings 12–13
 - displaying 4–5
 - names of all accessible 271
 - recording 5–7
 - restoring layouts 15
 - write permission 177–185

Z

zcat program 91

zero-link-table 240–243

Zoom Window menu command 14, 270



Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014
(408) 996-1010
TLX 171-576

030-2845-A
Printed in U.S.A.