

SCO® XENIX® System V Operating System Installation and System Administrator's Guide

SCO® XENIX® System V Operating System

Installation and
System Administrator's Guide



SCO

SCO® XENIX® System V
Operating System
Installation and
System Administrator's Guide

© 1983-1991 The Santa Cruz Operation, Inc.

© 1980-1991 Microsoft Corporation.

© 1989-1991 AT&T.

All Rights Reserved.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, The Santa Cruz Operation, Inc., 400 Encinal, Santa Cruz, California, 95061, U.S.A. Copyright infringement is a serious matter under the United States and foreign Copyright Laws.

The copyrighted software that accompanies this manual is licensed to the End User only for use in strict accordance with the End User License Agreement, which should be read carefully before commencing use of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of The Santa Cruz Operation, Inc.

The following legend applies to all contracts and subcontracts governed by the Rights in Technical Data and Computer Software Clause of the United States Department of Defense Federal Acquisition Regulations Supplement:

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 52.227-7013. The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, California 95061, U.S.A.

SCO and the SCO logo are registered trademarks of the Santa Cruz Operation, Inc.

Microsoft, MS-DOS, and XENIX are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of UNIX Systems Laboratories in the U.S.A. and other countries.

The Santa Cruz Operation is a trademark of the Santa Cruz Operation, Inc.

Document Version: 2.3.4D

Date: 28 March 1991

Contents

1 Before You Start

- Introduction 1-1
- About Your Installation Media 1-2
- Installation Requirements 1-3
- What You Need to Know 1-4
- How to Use This Guide 1-5
- Creating and Formatting a Physical DOS Partition 1-6
- Installing on Disks with Greater Than 1024 Cylinders 1-9
- Responding to Prompts 1-10
- Planning Your Disk Layout 1-11
- XENIX Installation: The Short Form 1-18

2 XENIX-386 Installation Procedure

- Introduction 2-1
- Installation Instructions 2-2

3 XENIX-286 Installation Procedure

- Introduction 3-1
- Installation Instructions 3-3

4 Reinstalling Your System

- Introduction 4-1
- Shut Down the System 4-3
- Back-up the System 4-4
- Reinstallation Procedure 4-6
- Reinstall Packages and Applications 4-13
- Relink the Kernel 4-14
- Prepare Preserved Filesystems for Use 4-15
- Restore Your Backups 4-16
- Reestablish Your Second Hard Disk 4-17

A Troubleshooting Your Installation

- Introduction A-1
- Solving Problems During Installation A-2
- Solving Problems After Installation A-6
- Common System Resource Error Messages A-8
- Solving Undocumented Problems A-9
- Before Calling for Help A-13

B Creating an Emergency Boot Floppy

Introduction B-1

Creating the Floppy or Floppies B-2

C Setting the Time Zone Outside North America

Setting Your Time Zone C-1

D If You Created Additional Filesystems

Preparing Your Filesystems for Use D-1

E Using divvy with Manual Control

Using the divvy Menu E-1

F Installation Glossary

Terms Used During Installation F-1

Chapter 1

Before You Start

Introduction 1-1

About Your Installation Media 1-2

Installation Requirements 1-3

What You Need to Know 1-4

How to Use This Guide 1-5

 Using the Short Form Installation 1-5

Creating and Formatting a Physical DOS Partition 1-6

 Low Level Format Programs 1-8

 Converting fdisk Numbers 1-8

Installing on Disks with Greater Than 1024 Cylinders 1-9

Responding to Prompts 1-10

Planning Your Disk Layout 1-11

 Determining Swap Space Allocation 1-12

 Deciding Whether to Have a Separate (/u) Filesystem 1-13

 Disk Layout Worksheets 1-16

XENIX Installation: The Short Form 1-18

Introduction

1

This guide explains how to install the XENIX Operating System on your computer. Read through this guide thoroughly before beginning the installation procedure so that you become familiar with the various steps and terminology used in each section. If you run into difficulties during the installation, please contact your provider for information on technical services.

There are separate installation procedures for XENIX-386 and XENIX-286 distributions. Be sure you use the correct chapter. In addition, a set of shortened instructions is provided for certain simple configurations. You should read this chapter carefully before proceeding to the "Installation Procedure" chapters.

About Your Installation Media

The XENIX System distribution includes the following:

- floppy diskettes labeled N1, N2, and N3
- Base Utilities diskettes numbered B1, B2, and so forth (XENIX-386 includes a single diskette)
- Extended Utilities, a set of diskettes numbered X1, X2, X3, and so forth

XENIX-386 distributions contain fewer diskettes because they use a compressed format.

Installation Requirements

This is what you need to install the XENIX System:

- a supported computer with an appropriate amount of memory. Check the *Release Notes* for a list of supported machines, to find the amount of memory needed with your system, and any additional requirements.
- a large amount of disk space if you wish to install the complete distribution. Consult your *Release Notes* for this information. If you install a subset of the distribution, add the sizes of the segments you wish to install as instructed in the *Release Notes*.
- one floppy disk drive
- the XENIX System distribution media
- a serial number (an alphanumeric code printed on the Serialization Card)
- an activation key (an alphabetic code printed on the Serialization Card)

What You Need to Know

Before you begin the installation procedure, make sure your computer is fully assembled (and operational) and you are familiar with its operation. In particular you should know the following:

- how to turn the computer off and on
- how to insert floppy disks into the floppy drive
- how to reset the computer

If you have just assembled your computer for the first time or are unsure about the items listed above, briefly review the hardware manuals provided with your computer and hard disk. The hard disk must be connected to your computer according to manufacturer's specifications. Also, we suggest you run a system self-test as described in the *Operator's Guide* for your computer to detect possible hardware problems.

Note

If you plan to also partition the hard disk for use with DOS or OS/2, you must install DOS first. Refer to "Creating and Formatting a Physical DOS Partition" in this chapter and "Using DOS and OS/2" in the *System Administrator's Guide* for details.

How to Use This Guide

This guide includes four chapters: this introductory chapter, which describes how to prepare for your installation, installation procedures for XENIX-386 and 286 distributions, and a chapter explaining how to reinstall the system. In addition, there are several appendixes that contain information on specialized configuration options that are applicable to only a few installations. Three exceptions are as follows:

- Appendix A, “Troubleshooting Your Installation,” contains tips on solving some of the common problems that you may have during, and immediately after, installation.
- Appendix B, “Creating an Emergency Boot Floppy Set,” explains how to create a set of floppies that make it easier to recover your system in the event of a system failure.
- Appendix F, “Installation Glossary,” contains terms used in the installation procedure. If you find a computer term you are not familiar with, look at this appendix.

Using the Short Form Installation

A set of simple instructions is provided in this chapter for certain configurations. It includes exact responses using system defaults so you do not have to plan your configuration. Read the section at the end of this chapter “XENIX Installation: The Short Form” to see if this option is useful for you.

Creating and Formatting a Physical DOS Partition

If you want a physical DOS partition on your XENIX System, you must create one before you install the XENIX System. To create a DOS partition once you have already installed the XENIX System, you must first back up your hard disk, create the DOS partition, and then reinstall your XENIX system and all applications.

Use the utilities and documentation packaged with your computer hardware to create and format a DOS partition.

The following restrictions apply when creating a DOS partition:

- XENIX supports MS-DOS versions 3.3 or earlier only.
- Always install the MS-DOS partition first. Because some versions of MS-DOS allow the DOS partition to start on the first cylinder only, the DOS partition should be the first partition on the first disk.
- Always format your new DOS partition with the following MS-DOS command:

```
A> format c: /s
```

This command formats the disk in the specified drive to accept MS-DOS files. The /s option copies the operating system files listed from the diskette to the hard drive (c:).

Note

The MS-DOS **format** command is not the same as a low-level format.

- Always make sure the size of your XENIX partition is a multiple of the number of heads on the hard disk. (If you remapped the drive during a low-level format of the disk, use the mapped number of heads.) Otherwise XENIX **fdisk** warns that the XENIX partition is not on a cylinder boundary.
- Never begin the XENIX partition on the track immediately following the last track of the MS-DOS partition. Because MS-DOS does not expect another operating system after its ending track, sometimes it uses that space to store programs temporarily. If you install the XENIX partition directly after the DOS partition, you may lose the boot block.
- Your DOS partition should have a minimum size of 2.5 Mbytes; some DOS copy-protection schemes do not install on a partition smaller than 2.5 Mbytes.
- Never use more than 32 Mbytes for each MS-DOS partition.
- Never use the extended partition option in MS-DOS **fdisk**. Extended DOS partitions are not supported. XENIX allows you one MS-DOS partition per drive.
- Never use a third party disk partitioning product, such as Disk Manager, to partition the hard disk before installing MS-DOS and the XENIX System. Use MS-DOS **fdisk** to partition MS-DOS, and XENIX **fdisk** to partition the XENIX System.

Note

If you do not obey the preceding restrictions, you must start over with a low-level format of your hard disk and then reinstall the XENIX System.

Low Level Format Programs

Use a program such as **DOS-Debug** or **Speedstor** to perform a low-level format of the disk. Many ESDI and SCSI disk controllers have on board BIOS routines. You can use these routines to perform a low-level format of your disk.

Follow these steps:

1. Boot MS-DOS.
2. At the DOS prompt, enter:

debug

3. Depending on your disk controller, enter one of the following commands at the “-” prompt:

- For Adaptec SCSI controllers:

g=dc00:6

- For ESDI controllers:

g=c800:5

Converting fdisk Numbers

DOS **fdisk** uses cylinders and XENIX **fdisk(ADM)** uses tracks. If you know the number of tracks per cylinder, you can use consistent numbers when using the two different versions of **fdisk**.

To obtain these numbers, you must know the total number of tracks and cylinders. XENIX **fdisk** lists the total number of tracks; the total number of cylinders should be listed in the documentation for your hard disk. Divide the total number of tracks by the total number of cylinders. You can then use this factor to convert cylinders to tracks. If you need to convert tracks or cylinders to megabytes, divide the total number of tracks or cylinders by the disk size. The result is your conversion factor.

Installing on Disks with Greater Than 1024 Cylinders

If your primary (root) hard disk has more than 1024 cylinders, the following restrictions apply:

- The disk controller must support disks with more than 1024 cylinders.
- If the disk is configured as the primary hard disk, the root filesystem **must** lie within the first 1024 cylinders. This is because the ROM BIOS cannot access boot information if it lies beyond the 1024th cylinder.

If the boot information lies beyond the 1024-cylinder boundary, the system may hang mysteriously or display error messages. For more information, see Appendix A “Troubleshooting Your Installation” in this guide.

You can use the rest of the disk for swap space and additional filesystems. If you install a XENIX partition only, verify that the root filesystem is located within the first 1024 cylinders.

Because this limitation is expressed in cylinders and XENIX **fdisk**(ADM) uses tracks, you should determine the number of tracks per cylinder for your disk. To do this, follow the instructions in the section “Converting fdisk Numbers” earlier in this chapter. To determine the maximum allowable extent of the root filesystem in tracks, divide the total number of tracks by the total number of cylinders and multiply this number by 1024.

Note

If you remapped the disk during formatting, the root filesystem must lie within the first 1024 logical cylinders.

Responding to Prompts

Follow these rules when reponding to prompts for information:

- Enter all requested names and numbers exactly as shown.
- Complete a line by pressing the `<Return>` key. The `<Return>` key is sometimes denoted on the keyboard by a “down-left” arrow, or referred to as the ENTER key.
- Press `<Ctrl>u` to delete everything on a line. (Press the `<Ctrl>` key and another key at the same time. This is the same type of action as using the `<Shift>` key.)
- Erase a character with the `<Bksp>` Key or by pressing `<Ctrl>h`.
- `(y/n)?` is asking “yes or no?” and always requires a response from you. Enter `y` or `n` and press the `<Return>` key.
- Commands referred to in the text are shown in **boldface** with the reference section next to the command in parentheses (for example, **cat(C)**).

Planning Your Disk Layout

If you anticipate a large number of users, heavy development usage, or you plan to use a database program, it is advisable to plan the layout of your hard disk. Systems under heavy use require decisions that affect performance.

This planning section discusses two segments of your hard disk (*/u* filesystems and the swap area) in relation to the XENIX root filesystem (the DOS partition is mentioned here because it affects available disk space):

- Swap space

The swap space is a part of your disk that acts as an extension of your physical memory (RAM). Programs (or parts of programs) that are in active use but currently in a waiting state can be shifted to this area (swapped out) so that others can run in RAM. The swap space must be large if there are many users or large applications, such as databases.

- Separate (mountable) filesystems (*/u*)

A large hard disk (70+ Mbytes) with many users can be slowed down by a large root filesystem. Creating a separate (usually called */u*) filesystem for user accounts improves performance, makes backups easier, and helps protect the root filesystem from corruption during a system crash.

- DOS partition

Those who wish to have DOS share the hard disk must consult the "Using DOS and OS/2" chapter of the *System Administrator's Guide* and load DOS on their disk first. The presence of a DOS partition makes it essential to plan your disk resources well, because it is necessary to reinstall the operating system to alter the layout of your hard disk. Read this section and plan your layout before installing DOS.

A series of simple worksheets is provided at the end of this section for calculating the size of your disk divisions. Although these calculations may look intimidating, they are entirely mechanical and only require the insertion of simple values.

Determining Swap Space Allocation

When the physical memory (RAM) is not large enough to contain all the running processes on the system, sleeping processes (tasks that are on hold while another process is being attended to) are moved to an area known as the "swap space." The swap space is a reserved portion of the hard disk, which should be at least as large as the amount of RAM you have installed. Swap space is allocated during operating system installation. The following are two methods for calculating the swap space size.

Letting `divvy`(ADM) decide for you

If the system is used under typical multiuser operation (no large applications or development system), you can use the default swap space recommended during the installation procedure.

After `badtrk`(ADM) has run, you will be prompted to choose the amount of space to allocate for the swap space. `divvy` calculates the minimum and maximum size based on the size of the hard disk. The following formulas are used (where blocks are 1024 bytes, i.e. 1K):

Minimum = 1000 blocks or (size of disk)/6, whichever is less
Maximum = 2000 blocks or (size of disk)/6, whichever is more
Default = ((3 x Minimum) + Maximum) / 4

If you plan to install the XENIX System V Development System, enter a swap space size that is at least 500 blocks larger than the default. This is especially important if you are planning to use large applications such as databases, spreadsheet programs, or plan to make programs memory-resident for performance reasons.

Calculating the swap space yourself

Systems using major applications (a large relational database, for example) should allocate the swap space according to the following formula. (Use the worksheet provided at the end of this section to perform your calculations.)

1. Multiply the number of users on the machine by the size of the largest process (running program) normally run on the machine, in 1K bytes. If no unusual processes apply, use 512K per user.
2. Take the amount of memory installed in your machine and add 256K. Compare this result with the result from step 1. Use whichever number is larger as your swap space size.

3. Adjust the number upward if multiple users are running different large applications. Try adding 512K per different large application. (Consider the Development System to be a large application.)

For example, the calculations for an 8-user machine with 4Mb of memory and a typical mix of spreadsheet, database, word processor, and graphics package:

1. 8 users x 512K = 4096K.
2. 4096K memory installed + 256K = 4352K, which is greater than the number calculated in step 1.
3. There are 4 users using different large applications. 4 x 512K = 2048K. Add this to the greater amount from steps 1 and 2 to obtain a reasonable calculation of swap space: 4352K + 2048K = 6400K. Since the operating system uses a block size of 1K bytes, this means that a swap space allocation of 6400K translates simply to 6400 blocks.

Deciding Whether to Have a Separate (/u) Filesystem

If you are installing on a hard disk (or partition) that is larger than 70 Mbytes, you are asked to decide whether to divide the XENIX partition into two separate filesystems. Use separate filesystems if the size of your root filesystem and swap space subtracted from the size of your XENIX partition is greater than 20 Mbytes.

You can create additional filesystems even if your partition is smaller than 70 Mbytes; this is simply the automatic formula. This other filesystem is typically called */u*; however, it can be named anything you like except *root* or *swap*. The */u* filesystem is typically used for user accounts and files. There are certain advantages and disadvantages to using a */u* filesystem.

There is no need to create a */u* filesystem if your setup is a runtime system (none or very few operating system packages installed) with a single dedicated application or with little user file storage.

Planning Your Disk Layout

Advantages of a /u Filesystem

There are several advantages to separate filesystems:

- System backups are easier.

In a typical system, user files can change often while XENIX utilities and system files usually remain unchanged once they are installed. Most users create and modify files in their home accounts frequently, so user accounts should be backed up often, while the root filesystem should only require an occasional backup. With separate filesystems, less media (for example, fewer floppy disks, fewer tapes, smaller tape size, and so on) are required to make backups of individual filesystems.

- Smaller filesystems are faster.

File access is faster because the operating system has fewer inodes (file identifiers) and data blocks (filesystem subdivisions) to search. Each filesystem has its own inode and data block areas. Another reason is when a system is shut down improperly ("crashes"), all filesystems are checked and cleaned using `fsck(ADM)`. It takes less time to clean two smaller filesystems than one large filesystem.

- During a system crash, a less active filesystem usually sustains less damage than an active filesystem.

By decreasing activity in the root filesystem, there is less chance of damage to the root filesystem. The `/u` filesystem can be recreated easily by remaking it and restoring it from backups.

Disadvantages of a /u Filesystem

There are two disadvantages, both relating to administration rather than performance:

- Once a division is made, it is impossible or very time consuming to change the size of or delete a filesystem. To do this the entire XENIX partition must be backed up, and the filesystem sizes changed using the `divvy(ADM)` utility; then the entire system must be restored from backups. This should only be done by an experienced system administrator. It is often easier to simply reinstall from scratch.

- There is a degree of additional operating system complexity introduced by an extra filesystem. This can make system administration more complicated for novices. This can include such mundane concerns as remembering to mount other filesystems if they are not mounted automatically at boot time.

Determining Space for a /u Filesystem

To determine if there is sufficient room on your hard disk for a /u filesystem, perform the calculations described in the following steps. (This formula is lengthy; review it here and then use the worksheet at the end of this section.)

1. Add together the sizes of all the operating system packages that you intend to install on your system. Package sizes (in 512 byte blocks) are found in the *Release Notes* under “Operating System Packages.”
2. Calculate the size for the */tmp* directory. */tmp* is used by many XENIX programs as a temporary storage area. Allow at least half a megabyte per user. Add another half a megabyte per user to this number if some users will be using large databases, editing large files, or working on software development.
3. Be sure to include the disk size requirements for any applications on the root filesystem. Most applications usually list disk space requirements in their *Release Notes*.
4. Add the results of steps 1, 2, and 3. This represents the size of a completely full root filesystem.
5. In addition, 10-20% of the size of the root filesystem must be kept as free space after all the software is installed. Take 20% of the root filesystem size obtained in the previous step and add it to that number. As an example, let's assume the root filesystem is 40 Mbytes in size. To keep approximately 20% of it free, we would add 8 Mbytes to the root filesystem size, for a total of 48 Mbytes. When a filesystem becomes filled beyond this point, an inordinate amount of time is spent by the operating system finding free space when needed, causing performance to suffer.
6. Subtract the requirements for the root filesystem (as calculated in step 5) from the size of your XENIX partition (or size of your hard disk if the XENIX system is the sole occupant). You should have more than 10 Mbytes left on your hard disk before considering adding a /u filesystem.

Planning Your Disk Layout

7. Calculate the requirements for the `/u` filesystem. Allow at least 1.5 Mbytes of disk space per user on the `/u` filesystem. Add in the requirements for any databases in the `/u` filesystem. Database requirements depend upon the maximum size the files will reach, how many users will be using the database, and what files if any will be shared among the users of a database. Also, add the size of any applications that will reside in the `/u` filesystem. If the space needed for the `/u` filesystem is less than the number calculated in step 5, then you can safely add a `/u` filesystem.
8. Multiply the size in megabytes of the `/u` filesystem by 1000 to get the number of 1K blocks to allocate to the `/u` filesystem.

Disk Layout Worksheets

Use the following worksheets to perform the calculations discussed in the preceding sections.

Swap Space Allocation			
1. Number of users:	_____	x 512K	= _____
2. Amount of RAM installed:	_____	+ 256K	= _____
3. Greater of (1) and (2):			= _____
4. Number of large Applications:	_____	x 512K	= _____
5. Sum of (3) and (4):			= _____

Sizes of Disk Divisions	
DOS partition:	_____
XENIX partition:	_____
Swap space:	_____
<code>/u</code> filesystem:	_____

/u Filesystem Allocation		
Instructions	Calculation	Result
1. Sum of OS package sizes to be installed (converting 512 byte blocks to megabytes)	_____ / 2000	= _____
2. Number of users times ½ Mbyte per user	_____ x .5 Mbyte	= _____
3. Number of users using database or Development System times ½ Mbyte per user	_____ x .5 Mbyte	= _____
4. Add (2) and (3) for size of /tmp	= _____
5. Disk requirements for applications	= _____
6. Size of root filesystem: add (1), (4), & (5) plus 20%	_____ x 1.2	= _____
7. Subtract size of swap space from XENIX partition	_____ - _____	= _____
8. Subtract size of root filesystem (6) from available disk space (7)	_____ - _____	= _____
9. If (8) is less than 5-10 Mbyte, you cannot have a /u filesystem
10. Number of users times 1½ Mbyte per user	_____ x 1.5 Mbyte	= _____
11. Size of applications to reside in /u	= _____
12. Size of any database files to reside in /u	= _____
13. Add (10) through (12) for size of /u filesystem	= _____
14. If /u (13) is less than (8), continue, else no /u
15. Convert /u (13) size to blocks	_____ x 1000	= _____

XENIX Installation: The Short Form

The following is a shortened set of instructions for installing XENIX. It provides a set of answers that use system defaults and perform a simple installation without any special options. If you have any of the following requirements, you should not use this short form installation:

- An SMS-OMTI controller or SCSI host adapter
- A non-standard hard disk (a disk not defined in your computer's ROM BIOS)
- Multiple filesystems
- A DOS partition

Follow these steps, giving the exact responses specified:

1. Insert the N1 floppy into the floppy drive.
2. Turn on the computer.
3. Press <Return> when the "Boot:" prompt appears.
4. Select your keyboard type when the keyboard selection menu is displayed. Be sure to press <Num Lock> key and use your numeric keypad if one is present.
5. Enter 1 when prompted for the controller type.
6. Enter y when asked if you wish to continue and overwrite your hard disk.
7. Select 3, "Select Default Disk Parameters" when the hard disk configuration menu appears, and quit by entering q and pressing <Return>.
8. Select 2 "Scan Disk" from the main **badtrk** menu.
9. Select 1 "Scan entire XENIX partition."

10. Select **2** “Thorough Scan.”
11. Enter **y** for a destructive scan.
12. Enter **y** when asked if you wish to continue.
13. Press **<Return>** when prompted for the number of bad tracks.
14. Enter **y** to update the device with the new table.
15. Enter **n** when prompted to salvage data on bad tracks.
16. When prompted for the swap space allocation, enter the amount of memory (RAM) you have (or plan to install) in Kbytes multiplied by 1.5. For example, if you have 3 Mbytes (3000 Kbytes) of memory installed, enter 4500.
17. If prompted for a **/u** filesystem, enter **n**.
18. Enter **n** when prompted for “block-by-block” control.
19. Enter serial number and activation key when prompted.
20. Follow the reboot instructions as displayed.
21. Assign root password and time zone as prompted.
22. Select **1** “Continue installation” when prompted.
23. Select **1** “Operating System” when the **custom** menu prompts for a set to customize.
24. Select **1** “Install one or more packages.”
25. Enter **ALL** when prompted for packages and insert volumes when asked to do so.
26. When asked if this installation consists of only a root filesystem, enter **y** and press **<Return>**.
27. To install additional products, you should select option **6** “Select a new set to customize.” This returns you to the main **custom** menu. Then select **4** “Add a supported product.”

XENIX Installation: The Short Form

28. When the **custom** menu returns, enter **q** to quit and follow reboot instructions.
29. You should create an Emergency Boot Floppy as instructed in the “Creating an Emergency Boot Floppy” appendix in this guide.
30. We also recommend that you also do a complete backup of your system as discussed in the “Backing Up Filesystems” chapter of the *XENIX System Administrator's Guide*. This will save you the trouble of reinstalling the system should any problems occur.

Chapter 2

XENIX-386 Installation Procedure

Introduction 2-1

Installation Instructions 2-2

Introduction

This chapter describes how to install XENIX-386. If you have not read Chapter 1, “Before You Start,” please do so now. Chapter 1 contains important information you should be aware of before installing your system.

2

Read through this chapter thoroughly before beginning the installation procedure so that you become familiar with the various steps and terminology used in each section. If you run into difficulties during the installation, support is available from a wide range of sources. Please contact your provider for more information on technical services.

The installation process consists of several stages:

- starting the system from the N1 floppy disk
- preparing the hard disk for installation
- starting the operating system from the newly initialized hard disk
- installing the parts of the XENIX distribution you need and loading your application programs
- configuring your system with the menu-driven **sysadmsh** system administration shell, including adding user accounts
- preparing the system for general use, including setting the system (*root*) password and restarting the system to make your configuration changes take effect

Installation Instructions

This section guides you through the XENIX System installation process. When necessary, you are directed to follow additional steps contained in the appendixes at the back of this guide.

Note

If you wish to partition the disk to accommodate another operating system, stop immediately. If you intend to install an MS-DOS partition, you must create it before installing the XENIX System. If you do not, you may render your XENIX partition unusable. Consult your MS-DOS installation documentation to create your DOS partition. See “Creating and Formatting a Physical DOS Partition” in Chapter 1 of this guide for important information. “Using DOS and OS/2” in the *System Administrator's Guide* explains how the XENIX partition and the MS-DOS partition share the hard disk.

Follow these steps, remembering to press <Return> after you enter your responses on the keyboard.

1. Find the distribution disk labeled N1.
2. Insert the N1 floppy disk into the drive. If you have more than one floppy drive, use the primary drive (sometimes called the boot drive). Check your computer hardware manual if you are not sure which drive is the primary drive.

3. Turn on your computer. The computer loads the XENIX bootstrap program from the floppy disk and executes it. The computer may also display the total amount of memory installed. Next, you see:

```
XENIX System V
```

```
Boot
:
```

2

Press **(Return)** to boot from the floppy drive.

4. Next you see:

```
fd(64) xenix root=fd(64) swap=ram(0) pipe=ram(1) swaplo=0 nswap=1000 ronly
```

5. After the operating system kernel is loaded in memory, the system displays a copyright statement followed by system configuration information. This information appears similar to the following:

```
device  address      vector dma comment
-----
%serial 0x03F8-0x03FF  04   -   unit=0 type=Standard nports=1
%serial 0x02F8-0x02FF  03   -   unit=1 type=Standard nports=1
%floppy 0x03F2-0x03F7  06   2   unit=0 type=96ds15
%console -              -    -   unit=ega type=0 12 screens=68k
%parallel 0x0378-0x037A  07   -   unit=0
%disk   0x01F0-0x01F7  16   -   type=W0 unit=0 cyls=286 hds=16 secs=63
mem: total = 3712k, kernel = 1436k, user = 2276k
rootdev = 1/40, swapdev = 1/41, pipedev = 1/40
nswap = 30000, swplo = 0, Hz = 60
kernel: i/o bufs = 468k
```

The system performs a self-check to determine if there are any problems detected with the hardware. Each stage generates a letter from A-Z, most of which are displayed and overwritten too rapidly to be seen.

6. If the letters stop displaying before the letter "Z" is reached and no boot prompt appears, run hardware diagnostics as explained in your computer manual, correct any identified problems, and start the installation procedure again from the beginning. If the letters stop again, please contact your provider for more information and be prepared to report the last letter displayed.

The self-check using letters A-Z occurs every time you bring up your XENIX System.

Installation Instructions

7. Next, the following is displayed:

```
No single-user login present
Entering System Maintenance Mode
Please press <RETURN>
```

Press <Return>.

8. You see the following menu:

```
Keyboard Selection
```

1. American
2. British
3. French
4. German
5. Italian
6. Spanish

```
Use the Numeric Keypad if present, using <Num Lock> if necessary,
to select one of the above options:
```

Select the number that corresponds to your keyboard.

9. Next, the disk controller menu is displayed:

```
XENIX System V Hard Disk Initialization
```

```
What type of disk controller will be supporting this disk?
```

1. Standard disk support (ST506, MFM, RLL, ESDI, IDE)
2. OMTI 8620 or OMTI 8627
3. SCSI

```
Enter 1, 2, 3 or enter q to quit:
```

Enter the number that corresponds to the controller you have.

10. If you are adding an OMTI controller, an additional prompt appears:

```
What interface will this hard disk use?
```

- 1) ESDI
- 2) ST506

```
Enter 1 or 2 or enter q to quit:
```

2

Enter the number that corresponds to the type of interface the hard disk will use.

11. The following prompt is displayed:

```
During installation you may choose to overwrite all  
or part of the present contents of your hard disk.
```

```
Do you wish to continue? (y/n)
```

Enter y and press (Return).

12. If you have a SCSI controller, you see a boot message about your hard disk similar to this, followed by the prompts shown:

```
%disk 0x0330-0x0332 11 5 type=ASGD00 unit=0 cyl=316 hds=64 secs=32
```

```
The hard disk installation program will now invoke /etc/fdisk.  
Entering 'q' at the following menu will exit /etc/fdisk.  
and the hard disk installation will continue.
```

```
If you wish to exit the entire installation at this menu,  
press the <del> key.
```

Skip to step 18.

Installation Instructions

13. If you have an ST506/ESDI (standard interface) controller, you see the following messages (the %disk message will differ depending on the disk you have installed):

```
The hard disk installation will now invoke /etc/dkinit.  
Entering 'q' at the following menu will exit /etc/dkinit,  
and the hard disk installation will continue.  
  
If you wish to exit the entire installation at this menu,  
press the <del> key.  
  
%disk 0x01F0-0x01F7 16 - type=W0 unit=0 cyl=286 hds=16 secs=63  
Hard Disk Drive 0 Configuration  
  
    1. Display current disk parameters  
    2. Modify current disk parameters  
    3. Select default disk parameters  
  
Enter an option or 'q' to quit:
```

14. If you have an OMTI controller, you see the following additional message:

```
Caution: Consult the SMS-OMTI section of your Release Notes if  
you wish to modify the disk parameters that /etc/dkinit  
will display.
```

Read the section "SMS-OMTI Guidelines" in your *Release Notes*.

If you enter **q**, you see the following message:

```
The hard disk installation program will now invoke  
two disk preparation utilities: fdisk and badtrk.  
Selecting 'q' at the main menu for each utility  
will exit that utility and continue with the hard  
disk installation.  
  
To exit the entire installation at these menus, press the <del> key.
```

Skip to step 18.

15. The **dkinit** menu is intended for unusual or nonstandard disks. If you have a standard hard disk, one that is supported by your computer hardware or special motherboard ROM, enter **3** followed by **<Return>** to select the default parameters. You should then enter **q** and press **<Return>** to continue the installation; skip to step 18.

Note

If you are not sure if your disk is non-standard, check the default parameters using option 1 of the **dkinit** menu. Calculate the size of your disk in bytes using the following calculation:

$$\text{size} = \text{cylinders} \times \text{heads} \times \text{sectors/track} \times 512$$

In addition, some drives are sold by formatted size, others by unformatted size. The formatted size of a drive is approximately 85% of its unformatted size. Note that the parameters displayed by **dkinit** may not match the drive manufacturer's documentation. Some controllers have optional "translation," "mapping," or "63-sector" modes. If one of those modes was chosen during low-level formatting, XENIX must be initialized with the translated parameters and not those of the physical drive. In all cases, the known size of the drive should approximately match the size calculated above from the disk parameters.

If your disk is nonstandard, you must enter information that overrides the ROM disk configuration information, replacing it with the new information. If you are unsure of what parameters to enter for your nonstandard disk, contact your disk manufacturer for this information. The **dkinit** program (called during installation) uses BIOS parameters that should be documented in the hardware manual for your computer.

If you enter 1 or 2, you see the following display:

Disk Parameters	Values
1. Cylinders	<i>value</i>
2. Heads	<i>value</i>
3. Write Reduce	<i>value</i>
4. Write Precomp	<i>value</i>
5. Ecc	<i>value</i>
6. Control	<i>value</i>
7. Landing Zone	<i>value</i>
8. Sectors/track	<i>value</i>

In the actual display, *value* is replaced with the default value for that variable.

Installation Instructions

Note

The “Cylinders” value refers to the number of cylinders on the entire hard disk and should not to be confused with the number of cylinders you allocated (or intend to allocate) to a given partition.

If you entered a **1**, you now see the first menu again. If you entered a **2**, you are now prompted:

```
Enter a parameter to modify or 'q' to return to the main menu:
```

16. Enter any number from 1 to 8 to change the disk parameters, or **q** to return to the previous menu. You see the following:

```
Enter the new value or <RETURN> to use the existing value:
```

If you wish to change the value, enter a new value now or press **<Return>** to use the existing value.

17. After you finish changing the disk parameters, enter **q** to return to the main menu. Next, enter **q** again to save the changes you made. Exiting from **dkinit** by entering **q** overwrites any parameters you changed with the new values. If you wish to restore the default parameters after making modifications, enter **3** from the first menu.
18. The installation program next runs the **fdisk(ADM)** utility to partition the hard disk. You can partition your disk to also support DOS on the same hard disk (if you have DOS already installed), or you can use the whole disk for your XENIX system. Be sure you follow the guidelines for DOS partitions found in the “Before You Start” chapter of this guide.

After a moment, the **fdisk** menu appears on the screen. You see this option list:

- ```

1. Display Partition Table
2. Use Entire Disk for XENIX
3. Create XENIX Partition
4. Activate Partition
5. Delete Partition

```

Enter your choice or 'q' to quit:

2

Select option **1** and press **(Return)**.

If you have never installed an operating system on your disk, you see a table similar to this:

```
Current Hard Disk Drive: /dev/rdisk/1s0
```

| Partition | Status | Type | Start | End | Size |
|-----------|--------|------|-------|-----|------|
|           |        |      |       |     |      |

```
Total disk size: 1220 tracks (5 reserved for masterboot
and diagnostics)
```

```
Press <RETURN> to continue:
```

If you have previously installed an operating system on your disk, the **fdisk** table is filled in. DOS is usually displayed as partition number 4.

19. Press **(Return)** to return to the main **fdisk** menu. If you would like the XENIX partition to occupy the whole disk, select option **2**.

If you have an existing DOS partition you wish to keep, enter **3** and press **(Return)**. You are then prompted for the starting track number for the XENIX partition, followed by the size of the XENIX partition in tracks. **fdisk** displays the updated table when your changes are complete. Press **(Return)** to return to the main **fdisk** menu. Enter **4** to activate the XENIX partition (otherwise the DOS partition is still activated). The updated table is displayed again.

## Installation Instructions

If you selected option 2 and any other operating systems were previously installed on your system, you also see the following warning message:

```
Warning! All data on your disk will be lost!
Do you wish to continue? (y/n)
```

Enter **y** and press **<Return>** only if you want your XENIX system to occupy the whole disk. This ensures that **fdisk** partitions the whole disk.

---

### Note

Most computers have diagnostic programs that write to the last cylinder of the hard disk. This means that the last cylinder should not be allocated to a partition. The last cylinder is not allocated when you choose option 2 from the **fdisk** menu. If you choose option 3, you should not allocate the last cylinder of the hard disk to the XENIX partition.

- 
20. Press **<Return>**, and you see the main **fdisk** menu again. You have now set up the partition(s) on your hard disk. To continue with the next step in the installation procedure, enter **q** and press **<Return>**.

If you have an ST506/ESDI controller, continue with step 21.

If you have a SCSI host adapter, skip to step 31.

If you have an SMS-OMTI controller, enter **q** at the **badtrk** menu and skip to step 31.

---

### Note

Bad tracks are handled automatically by SCSI host adapters and SMS-OMTI controllers and thus steps 21-31 are omitted.

21. Now you see a menu from the program **badtrk**(ADM). With the **badtrk** program, you can scan your hard disk for defective tracks. The program maps any flawed locations to good tracks elsewhere on the disk. It also creates a bad track table, which is a list of all the bad tracks on your hard disk.

The main **badtrk** menu looks like this:

```
1. Print Current Bad Track Table
2. Scan Disk (You may choose Read-Only or Destructive later)
3. Add Entries to Current Bad Track Table by Cylinder/Head Number
4. Add Entries to Current Bad Track Table by Sector Number
5. Delete Entries Individually from Current Bad Track Table
6. Delete All Entries from Bad Track Table
```

Enter an option or 'q' to quit:

Enter **2**, then press **<Return>**.

22. You see the following submenu:

```
1. Scan entire XENIX partition
2. Scan a specified range of tracks
3. Scan a specified filesystem
```

Enter an option or 'q' to quit:

Select option **1**.

23. After you select the area you want scanned, you are given the following choices:

```
1. Quick scan (approximately 7 megabytes/min)
2. Thorough scan (approximately 1 megabyte/min)
```

Please enter an option or 'q' to quit:

Select option **2**.

24. You are prompted:

```
Do you want this to be a destructive scan? (y/n)
```

## Installation Instructions

Enter **y**. You are warned:

```
This will destroy the present contents of the region you are scanning.
Do you wish to continue? (y/n)
```

Enter **y** and press **(Return)**. You see the following message:

```
Scanning in progress, press 'q' to interrupt at any time.
Destructively scanning track x/y, z% of scan completed
```

25. After you respond to the above prompts, the program scans the active partition of the new disk for flaws. The larger your disk, the longer the scanning process takes, so a very large disk may take a while.

As **badtrk** scans the disk, it displays the number of each track it examines, and the percentage of the disk already scanned. Pressing the **q** key at any time interrupts the scan. If you press **q** to interrupt the scan, you do not need to press **(Return)**. You are then prompted to continue scanning or to return to the main menu.

Whenever **badtrk** finds a defective track, it lists the location of that track using both the sector number and cylinder or head conventions. Defective track information is entered into the table and displayed on the screen. Here is an example of a bad track:

```
wd: ERROR : on fixed disk ctlr=0 dev=0/47 block=31434 cmd=00000020
status=00005180, sector = 62899, cylinder/head = 483/4
```

26. When the scan is complete, the menu reappears. Select option **1** to see the results of the scan. Your bad track table looks something like this:

Defective Tracks

|    | Cylinder | Head | Sector Number(s) |
|----|----------|------|------------------|
| 1. | 190      | 3    | 12971-12987      |

Press **<RETURN>** to continue:

Press **(Return)** to return to the main menu.

---

### Note

If there is a flaw in the first few tracks of the XENIX partition, you are returned to the **fdisk** utility (see previous installation steps). Repartition the disk with **fdisk** so that the XENIX partition no longer includes the defective tracks. You have to experiment to determine how many tracks to exclude. Leave these defective tracks unassigned to any operating system. When you leave **fdisk**, **badtrk** is run again and you should scan the disk for further flaws.

This process continues until **badtrk** finds no flaws in the first few tracks.

---

27. To exit **badtrk**, enter **q** and press **<Return>**.
28. You are next prompted for the number of tracks to allocate as replacements for those tracks that are flawed. You should allocate at least as many as the recommended number. Enter the number or just press **<Return>** to use the recommended number that is displayed:

Enter the number of bad tracks to allocate space for  
(or press **<RETURN>** to use the recommended value of *n*):

If you press **<Return>** and do not enter an alternate value, **badtrk** allocates the recommended number of tracks as replacements. This number is based on the number of bad tracks currently in the table, plus an allowance for tracks that may go bad in the future. If you ever exceed the number of allocated bad tracks, you must reinstall the system.

29. Next, **badtrk** prompts:

Do you want to update this device with the new table? (y/n)

Enter **y** and press **<Return>** to save the changes. To correct any mistakes or otherwise alter the bad track table, enter **n**. Modify the bad track table to contain the desired entries, enter **q** at the main menu to return to the prompt displayed above, then enter **y** to update the device with the new table.

## Installation Instructions

30. Next, you see:

```
Do you want to attempt to salvage any valid data
on the bad tracks? [may take a long time] (y/n)
```

Enter **n** and press **<Return>**.

31. Next, the installation program runs the **divvy(ADM)** utility. This program allocates portions of your partitioned disk for the *root* and *swap* areas. It also allocates a small portion of the disk for a *recover* area that is used during autoboot by the **fsck(ADM)** program. **divvy** first prompts you for the swap space allocation:

```
There are n blocks in the XENIX area.
Between x and y blocks should be reserved for the swap area.
```

```
Please enter the swap space allocation, or press <RETURN>
to get the default allocation of z blocks:
```

The actual numbers in this prompt vary depending upon the size of your disk. If you used the Disk Layout Worksheets, enter the value you calculated, otherwise use the default values shown on your screen. (If you plan to install the Development System, enter a swap-area allocation that is at least 500 blocks larger than the default.) Enter your swap space allocation now or press **<Return>** to choose the default values.

---

### Note

Most applications will only install on the root filesystem. Make certain you allocate enough room on the root filesystem for your applications. If you run out of room on the root filesystem, you will have to reinstall XENIX.

- 
32. If you have enough disk space for a */u* filesystem, you see the prompt:

```
Do you want a separate /u filesystem? (y/n)
```

If you have enough storage for a separate user filesystem, and want to create one, enter **y** and press **<Return>**. If you do not wish to have one, respond **n** and proceed to the next step.

Next, you see:

```
Enter block allocation for the /u file system.
(min to max)
```

2

In the display, *min* and *max* are replaced with the minimum and maximum number of blocks that can be allocated for the filesystem. If you used the Disk Layout Worksheets, enter the value you calculated for the */u* filesystem. (See “Planning Your Disk Layout” for more information on the */u* filesystem.)

33. You are now prompted for block-by-block control of the partition:

```
The layout of the filesystems and swap area is now prepared
Do you wish to make any manual adjustments to the sizes or
names of the filesystems or swap area before they are created
on the hard disk? (y/n)
```

If you are not creating a set of special filesystems, enter **n** and press **<Return>**. Manual adjustment means you can choose the exact size of filesystems, name them, and adjust the swap area to fit your needs. Most users do not need this kind of precise control, so answering ‘n’ at this prompt causes **divvy** to use default settings based on the size of your hard disk. If you wish to allocate filesystems manually, enter **y** and press **<Return>**; Appendix E “Using **divvy** with Manual Control” explains how to do this.

34. The system now loads a rudimentary XENIX file system onto your hard disk. This takes several minutes. You see the message:

```
Making filesystems
```

35. Next, you see the RESTRICTED RIGHTS LEGEND followed by the prompt for serialization:

```
Enter your serial number
and press <RETURN>
```

## Installation Instructions

Enter the serial number exactly as it appears on your serialization card and press **<Return>**. Then you see the message:

```
Enter your activation key
and press <RETURN>
```

Enter the activation key exactly as it appears on your serialization card and press **<Return>**.

If you make a mistake, you see:

```
Error: Invalid activation key

Do you wish to try activation again? (y/n)
```

Enter **y** and press **<Return>**. Try your code again. If it still fails, contact your provider. If you enter **n**, the installation is aborted.

36. The system now shuts down and displays instructions on booting the newly initialized hard disk:

```
AFTER you see the message '*** Normal System Shutdown ***'
reboot the system by opening the floppy door and pressing any key.
The screen will clear and you will see the boot message:

 Boot
 :

Press <RETURN> to reboot and continue the installation.
```

37. Open the floppy door and press **<Return>** to reboot the system. You see the following boot prompt:

```
XENIX System V

Boot
:
```

Press **<Return>**.

38. You see the following:

```
hd(40) xenix
```

39. After a moment, copyright messages are displayed, followed by information about the configuration of your system.

2

As before, the system performs a self-check to determine where any problems exist with the hardware. The letters A-Z are displayed, each overlapping the previous letter. If the letters displaying stop before the letter "Z" is reached, run hardware diagnostics as explained in your computer manual. Correct any identified problems and start the installation procedure again from the beginning.

If the letters stop at this point, call your provider for technical support information and be prepared to say at what letter the display ended.

40. The **fsck(ADM)** utility checks the filesystem, displaying a series of messages:

```
No single-user login present
Entering System Maintenance Mode

XENIX System V Hard Disk Installation

Checking root file system

** Phase 1 - Check Blocks and Sizes
** Phase 2 - Pathnames
** Phase 3 - Connectivity
** Phase 4 - Reference Counts
** Phase 5 - Check Free List

xxx files yyyy blocks zzzz free
```

41. You are next prompted to insert the B1 floppy disk:

```
Insert Operating System (Basic Utilities) Volume B1
and press <RETURN>
```

Insert the B1 floppy and press <Return>.

### Note

If there is an error with the extraction procedure, you see the message:

```
Extraction error, try again? (y/n)
```

Make sure you are using the correct floppy disk, and that the door of the floppy drive is completely closed; then enter **y** and press **<Return>**. If the message persists, see "Error reading a floppy" in Appendix A, "Troubleshooting Your Installation," in this guide.

---

42. Next, you are prompted to assign the *root* password:

```
Please assign a password for the super user account, "root".
Enter new password (minimum of 5 characters)
Please use a combination of upper and lowercase letters and numbers.
New password:
```

The new password can be any combination of letters, numbers, and punctuation marks, but should be at least 5 characters long. Enter the new password and press **<Return>**. The system does not display the password as you enter it, so type carefully.

43. After you press **<Return>**, the system displays the message:

```
Re-enter new password:
```

Enter the new password once more and press **<Return>**. Make sure you enter it correctly, otherwise the program prompts you to enter the password again. When you have entered the password correctly, you see some information about passwords and how to change your super user password in the future.

The super user password is now in place. From now on, the password is required whenever you attempt to access the system as super user. The super user password keeps the system safe from unauthorized use. It is very easy to make errors that could destroy files when logged in as the super user. Log in as super user only to

install programs and to do system maintenance tasks. For a complete description of the super user, see the "Starting and Stopping the System" chapter of the *XENIX System Administrator's Guide*.

Do not forget the super-user password. To restore a forgotten super-user password, you must reinstall your XENIX System.

44. Next, you are prompted to set the time zone appropriate to your location. This procedure is simple if you are in North America. Those in other areas should read Appendix C, "Setting Time Zones Outside North America," and set their time zone as described there. The first thing you see is:

2

```
Time zone initialization
Are you in North America? (y/n)
```

If you are not in North America, enter **n** and proceed to Appendix C to set your time zone. If you are in North America, enter **y**. You see the following menu:

```
1. NST - Newfoundland Standard Time
2. AST - Atlantic Standard Time
3. EST - Eastern Standard Time
4. CST - Central Standard Time
5. MST - Mountain Standard Time
6. PST - Pacific Standard Time
7. YST - Yukon Standard Time
8. HST - Hawaiian/Alaskan Standard Time
9. NST - Nome Standard Time
```

Enter the number that represents your time zone or enter **q** to quit:

If, for example, your time zone is Nome Standard time, you would enter the number **9** and press **<Return>**.

45. You see the following:

```
Does daylight savings time (summer time) ever apply at your location? (y/n)
```

If daylight savings or standard time changes occur in your area, enter **y**. If not, enter **n**. The time zone variable, located in */etc/TZ*, is changed accordingly. Your time zone is now set.

## Installation Instructions

46. You see this message:

```
Setting up system directories.
```

47. Next, the following messages are displayed, followed by a "Current Disk Usage" summary:

```
You have now installed the Xenix minimum run time system.
The run time system will support most application programs,
so you may choose to stop installation now and preserve
the maximum available disk space for user files.
```

```
The remainder of the Xenix Operating System, Development
System and Text Processing sets are installable in small
packages. You may either install each set entirely or
selectively choose which packages of the set to install.
```

48. You also see a menu that gives you the option of stopping or continuing with the installation:

```
1. Stop installation now
2. Continue installation
```

So far you have installed the minimal, or "Run Time" system. If you are installing the operating system merely as a platform for an application, you may not need to install more software. However, should you need to administer the system extensively, have user accounts, and use the system administration shell, you need to install more software before completing the installation. If you want to install operating system packages or applications, select option 2 and press `<Return>`.

49. This stage involves "customizing" your system, adding only those parts of the operating system distribution that are useful to you. You can also install your applications as well. The installation program `custom(ADM)` is automatically invoked to install XENIX packages and other products, including applications.

50. The main **custom** menu is displayed:

```

1. Operating System
2. Development System
3. Text Processing System
4. Add a supported product

```

Select a set to customize or enter q to quit:

2

If you do not select a system to “customize” by choosing the ‘q’ option, you stop the installation procedure. You will need to invoke **custom** later to add other XENIX packages. To add Operating System Packages, select option “1”.

51. The following messages are displayed:

```

Installing custom data files ...

```

```

Insert Operating System (Extended Utilities) volume X1
and press <RETURN> or enter q to quit:

```

Insert the diskette as instructed and press <Return>.

52. The following submenu is displayed:

```

1. Install one or more packages
2. Remove one or more packages
3. List the available packages
4. List the files in a package
5. Install a single file
6. Select a new set to customize
7. Display current disk usage
8. Help

```

Select an option or enter q to quit:

Select option “1”. A list similar to the one in your *Release Notes* is displayed. Enter the name of each package separately, unless you select ALL. When you enter the name of a package, **custom** prompts for the necessary volume numbers. Insert the appropriate floppies and follow the screen prompts. For more information on installing portions of the Operating System, see **custom(ADM)** in the *XENIX Reference*.

## Installation Instructions

53. You are instructed to insert Volume X1. Because this volume is already in the drive, press `<Return>`.

Refer to "Packages in This Set" in your *Release Notes* for a list of operating system packages. The only software presently installed on your system is the RTS package, or Run-Time System. This represents the minimal software needed to have a functioning operating system. Other packages can be added that suit your needs; if your system is merely a platform for applications such as spreadsheets, word processors, or databases, you should install the following packages:

|        |                                                     |
|--------|-----------------------------------------------------|
| SYSADM | a user-friendly interface for system administration |
| BACKUP | utilities to perform backups                        |
| LPR    | the line printer spooler                            |
| MAIL   | mail program for sending messages to other users    |

In addition, packages such as BASE, VI, and CSH are useful for those who wish to work in a traditional XENIX environment, using the `ex(C)` or `vi(C)` file editors, various file utilities, and the C-shell (`csH(C)`). In addition, the KSH package containing the Korn shell is available on XENIX-386 distributions.

---

### Note

We recommend that you install the VI, FILE, BASE and LINK packages, as these contain useful utilities for maintaining your system. The VI package is recommended because it contains the `vi` editor, and the documentation uses this as an example editor when you are asked to edit default files. Some products (such as network software) have drivers that require linking into the kernel, so you should install the Link Kit (LINK).

---

If you have a large hard disk and space is not a problem, you can select ALL and install every package of the operating system.

54. If you choose to install ALL or the MAN package with **custom(ADM)**, you see prompts similar to the following:

```
Repacking MAN pages. (may take several minutes)
pack: manpage: X% Compression
```

The second message is displayed as each online manual page is compressed to save space on the hard disk.

2

55. If you choose to install ALL or the BACKUP package, you are prompted for information about the filesystems you configured on your system during installation. This information is used to set up a schedule for backing up your system files. See the "Backing Up Filesystems" chapter of the *XENIX System Administrator's Guide*.) The prompts are displayed after the files are loaded and appear as follows:

```
Does this installation consist of only a root filesystem?
```

If you respond **y**, then the **schedule** file is automatically modified to only back up a root filesystem.

If you respond **n**, you are prompted:

```
Does this installation consist of only the root filesystem
and a user filesystem named /u?
```

If you respond **y**, a message is displayed indicating that the default **schedule** is suitable for your system and need not be altered.

If you respond **n**, you are indicating that you have additional filesystems, which requires manual editing of the **schedule** file.

In addition to the filesystem question(s), you are prompted to create a password for the *backup* account. When you log in as **backup**, you are taken directly to the **sysadmin(ADM)** menu.

56. If you choose to install ALL or the SYSADM package with **custom(ADM)**, you are prompted for the *sysadm* password.

## Installation Instructions

57. If you choose to install ALL or the LINK package, you are prompted for your serial number and activation key again. Enter them as you did earlier in the installation. If you plan to install device drivers and you did not select ALL, you should also install the Link Kit (LINK) package. Some products (such as SNA, XENIX-NET, TCP/IP, and Xsight) also have drivers that require linking into the kernel, so you should install the Link Kit.
58. To install additional products, you should select option 6 "Select a new set to customize." This returns you to the main **custom** menu. You can install your applications using the "Add a Supported Product" option from the main **custom** menu. You are prompted for various volumes (including parts of your operating system distribution) just as you were for XENIX packages.
59. When you are finished installing XENIX packages and any products, quit out of **custom** by entering **q** at the main menu. You see:

```
AFTER you see the message '*** Normal System Shutdown ***'
reboot the system by opening the floppy door and pressing any key.
The screen will clear and you will see the boot message:

 Boot
 :

Press <RETURN> to reboot and continue the installation.
```

The system shuts down. Remove any floppy that is in the drive.

60. Press any key to reboot the system and press <Return> when the boot prompt appears:

```
Boot
:
```

---

### Note

If you do not press `<Return>` within the timeout period (one minute), the system boots automatically. In this case, steps 61 and 62 do not occur. Instead the system comes up in multiuser mode and the boot information is displayed, followed by the login prompt.

2

---

The following is displayed:

```
hd(40) xenix
```

The screen clears and you see some self check diagnostics.

61. Next, you are prompted to continue the startup or perform system maintenance:

```
Enter CONTROL-d to proceed with normal startup
(or give root password for system maintenance)
```

Because there are some system administration steps you should take now, enter the root password and press `<Return>`.

62. You are then asked to enter the terminal type:

```
TERM= (ansi)
```

Press `<Return>`.

63. The next thing you see is the root prompt, also used in system maintenance mode, which is a pound sign:

```
#
```

## Installation Instructions

64. If you wish to add a second hard disk, follow the instructions in “Adding a Second Hard Disk” in the “Managing Filesystems” chapter of the *XENIX System Administrator's Guide*.
65. If you created additional filesystems on your hard disk (such as */u*) earlier in the installation, follow the instructions in Appendix D, “If You Created Additional Filesystems,” before proceeding. This appendix prepares the additional filesystem(s) for use.
66. You should now add user accounts to the system, if desired, using the **mkuser**(ADM) program. If you have created a */u* filesystem for user accounts, you must first mount the filesystem before creating user accounts. Enter the following command:

```
mount /dev/u /u
```

Each person using the system should have a login account and password. The security of your system is greatly dependent on the restrictions you impose on the use of accounts. For more information, see the “Administering User Accounts” chapter of the *System Administrator's Guide*. You can also alter the defaults used to create accounts and administer them.

67. Now that you have installed your software and created your user accounts, you should exit maintenance (single-user) mode by pressing <Ctrl>d. This logs you out and brings the system up in multi-user mode.

Your system is now ready for users to log in.

## **Chapter 3**

# **XENIX-286 Installation Procedure**

---

Introduction 3-1

96tpi Versus 135tpi Distributions 3-1

Installation Instructions 3-3



---

# Introduction

This chapter describes how to install XENIX-286. If you have not read Chapter 1, "Before You Start," please do so now. Chapter 1 contains important information you should be aware of before installing your system.

Read through this chapter thoroughly before beginning the installation procedure so that you become familiar with the various steps and terminology used in each section. If you run into difficulties during the installation, support is available from a wide range of sources. Please contact your provider for more information on technical services.

3

The installation process consists of several stages:

- starting the system from the N1 floppy disk
- preparing the hard disk for installation
- starting the operating system from the newly initialized hard disk
- installing the parts of the XENIX distribution you need and loading your application programs
- configuring your system with the menu-driven **sysadmsh** system administration shell, including adding user accounts and choosing what kind of security you wish to have
- preparing the system for general use, including setting the system (*root*) password and restarting the system to make your configuration changes take effect

## 96tpi Versus 135tpi Distributions

The installation instructions in this chapter are for 96 and 135tpi distributions. Most of the steps are exactly the same regardless of the distribution you have.

However, there are some differences. These are marked with headings (96tpi, or 135tpi) that indicate a special step that applies to a specific distribution. Read the installation instructions carefully.

## **Introduction**

### **135tpi Distributions**

You are ready to start the actual installation process. Find the distribution disks labeled "BOOT" (N1) and "FILESYSTEM" (N2).

### **96tpi Distributions**

If your XENIX distribution is on high density, 96tpi floppies, there is only one BOOT/FILESYSTEM floppy. Throughout the installation procedure, all references to the BOOT and FILESYSTEM floppies refer to the single BOOT/FILESYSTEM floppy.

---

# Installation Instructions

This section guides you through the XENIX-286 installation process. When necessary, you are directed to follow additional steps contained in the appendixes at the back of this guide.

---

## Note

If you wish to partition the disk to accommodate another operating system, stop immediately. If you intend to install an MS-DOS partition, you must create it before installing the XENIX System. If you do not, you may render your XENIX partition unusable. Consult your MS-DOS installation documentation to create your DOS partition. See "Creating and Formatting a Physical DOS Partition" in Chapter 1 of this guide for important information. "Using DOS and OS/2" in the *System Administrator's Guide* explains how the XENIX partition and the MS-DOS partition share the hard disk.

---

**3**

Follow these steps, remembering to press <Return> after you enter your responses on the keyboard.

1. Insert the BOOT (N1) floppy into the drive. If you have more than one floppy drive, use the primary drive (sometimes called the boot drive). Check your computer hardware manual if you are not sure which drive is the primary drive.
2. Turn on your computer and the hard disk.

The computer loads the XENIX-286 bootstrap program from the floppy disk and executes it. In the upper left corner of the screen, the computer may display the total amount of memory installed. Next, you see:

```
XENIX System V

Boot
:
```

Press <Return> to boot from the floppy drive.

## Installation Instructions

3. You see the following:

```
fd(64) xenixroot=fd(64) swap=ram(0) pipe=ram(1) swaplo=0nswap=1000 nonly
```

4. If you have the 135tpi distribution, you see this prompt:

```
Insert Filesystem volume and press <RETURN>
```

Remove the BOOT floppy (N1) and insert the FILESYSTEM floppy (N2).

5. If you have the 96tpi distribution, leave the BOOT/FILESYSTEM floppy in the drive.
6. After the operating system kernel is loaded in memory, the system displays a copyright statement followed by system configuration information. This information appears similar to the following:

```
device address vector dma comment
----- -
%serial 0x03F8-0x03FF 04 - unit=0 type=Standard nports=1
%serial 0x02F8-0x02FF 03 - unit=1 type=Standard nports=1
%floppy 0x03F2-0x03F7 06 2 unit=0 type=96ds15
%console - - - unit=ega type=0 12 screens=68k
%parallel 0x0378-0x037A 07 - unit=0
%disk 0x01F0-0x01F7 16 - type=W0 unit=0 cyls=286 hds=16 secs=63
mem: total = 3712k, kernel = 1436k, user = 2276k
rootdev = 1/40, swapdev = 1/41, pipedev = 1/40
nswap = 30000, swplo = 0, Hz = 60
kernel: i/o bufs = 468k
```

The system performs a self-check to determine if there are any problems detected with the hardware. Each stage generates a letter from A-Z, most of which are displayed and overwritten too rapidly to be seen.

7. If the letters stop displaying before the letter "Z" is reached and no boot prompt appears, run hardware diagnostics as explained in your computer manual, correct any identified problems, and start the installation procedure again from the beginning. If the letters stop again, please contact your provider for more information and be prepared to report the last letter displayed.

The self-check using letters A-Z occurs every time you bring up your XENIX System.

8. Once the system begins to run, the following is displayed:

```
No single-user login present
Entering System Maintenance Mode

Keyboard Selection

1. American
2. British
3. French
4. German
5. Italian
6. Spanish

Use the Numeric Keypad if present, using <NUM LOCK> if necessary,
to select one of the above options:
```

3

Select the option that corresponds to your keyboard.

9. The following message is displayed:

```
XENIX System V Hard Disk Initialisation

During installation you may choose to overwrite all
or part of the present contents of your hard disk.
Do you wish to continue? (y/n)
```

If you do not have any files you want to save, enter **y** and press **<Return>**.

---

### *Note*

If you have files on the hard disk that you wish to save before installing XENIX, respond **n** and the installation will be aborted and the system shut down automatically. You should then reboot your existing operating system, back up any files you want to save, and restart the XENIX installation procedure from the beginning.

Back up files that reside on partitions that you are going to overwrite. Creating a new partition on an unused portion of the hard disk will not overwrite files on another partition; it is not necessary to back up files that reside on an existing partition you are not changing.

## Installation Instructions

10. The **mkdev** program now invokes **dkinit**, which sets the parameters for the hard disk. You see information about your hard disk and this menu:

```
Hard Disk Drive 0 Configuration

 1. Display current disk parameters
 2. Modify current disk parameters
 3. Select default disk parameters

Enter an option or 'q' to quit:
```

**dkinit** is primarily for unusual or non-standard disks. If you have a standard hard disk, one that is supported by your computer hardware or special motherboard ROM, enter **q** followed by  $\langle$ Return $\rangle$  to continue the installation.

Entering **q** at this point selects the default parameters for your hard disk. Unless you know that your disk is non-standard, assume that it is standard and enter **q** to continue your installation with **fdisk(ADM)**. Skip to step 13.

If your disk is non-standard, you must enter in information that overrides the ROM disk configuration information, replacing it with the new information. If you are unsure of what parameters to enter for your non-standard disk, contact your disk manufacturer for this information. The **dkinit** program (called during installation) uses BIOS parameters that should be documented in the hardware manual for your computer.

11. If your disk is non-standard, **dkinit** operates as follows...

If you enter “1” or “2”, you see the following display:

| Disk Parameters  | Values       |
|------------------|--------------|
| 1. Cylinders     | <i>value</i> |
| 2. Heads         | <i>value</i> |
| 3. Write Reduce  | <i>value</i> |
| 4. Write Precomp | <i>value</i> |
| 5. Ecc           | <i>value</i> |
| 6. Control       | <i>value</i> |
| 7. Landing Zone  | <i>value</i> |
| 8. Sectors/track | <i>value</i> |

In the actual display, *value* is replaced with the default value for that variable.

*Note*

The “Cylinders” value refers to the number of cylinders on the entire hard disk and should not to be confused with the number of cylinders you allocated (or intend to allocate) to a given partition.

If you entered a “1”, you now see the first menu again. If you entered a “2”, you are now prompted:

Enter a parameter to modify or 'q' to return to the main menu:

Enter any of “1” - “8” to change the disk parameters, or **q** to return to the previous menu.

Enter the new value or <RETURN> to use the existing value:

If you wish to change the value, enter a new value now or press (Return) to use the existing value.

## Installation Instructions

12. After you finish changing the disk parameters, enter **q** to return to the main menu. Next, enter **q** again to save the changes you made. Exiting from **dkinit** by entering **q** overwrites any parameters you have changed with the new values. If you wish to restore the default parameters after making modifications, enter "3" from the first menu.
13. The installation program next runs the **fdisk(ADM)** utility to partition the hard disk. You can partition your disk to also support DOS on the same hard disk (if you have DOS already installed), or you can use the whole disk for your XENIX system. Be sure you follow the guidelines for DOS partitions found in the "Before You Start" chapter of this guide.

After a moment, the **fdisk** menu appears on the screen. You see this option list:

1. Display Partition Table
2. Use Entire Disk for XENIX
3. Create XENIX Partition
4. Activate Partition
5. Delete Partition

Enter your choice or 'q' to quit:

Select option "1" and press (Return).

If you have never installed an operating system on your disk, you see a table similar to this:

Current Hard Disk Drive: /dev/rhd00

| Partition | Status | Type | Start | End | Size |
|-----------|--------|------|-------|-----|------|
|           |        |      |       |     |      |

Total disk size: 1220 tracks (5 reserved for masterboot  
and diagnostics)

Press <RETURN> to continue

If you have previously installed an operating system on your disk, the **fdisk** table will be filled in.

14. Press `<Return>` to return to the main **fdisk** menu. If you would like the XENIX partition to occupy the whole disk, select option 2.

If you have an existing DOS partition you wish to keep, enter 3 and press `<Return>`. You are then prompted for the starting track number for the XENIX partition, followed by the size of the XENIX partition in tracks. **fdisk** displays the updated table when your changes are complete. Press `<Return>` to return to the main **fdisk** menu. Enter 4 to activate the XENIX partition (otherwise the DOS partition is still activated). The updated table is displayed again.

If you selected option 2 and any other operating systems were previously installed on your system, you also see the following warning message:

```
Warning! All data on your disk will be lost!
Do you wish to continue? (y/n)
```

Enter **y** and press `<Return>` only if you want your XENIX system to occupy the whole disk. This ensures that **fdisk** partitions the whole disk.

---

#### Note

Most computers have diagnostic programs that write to the last cylinder of the hard disk. This means that the last cylinder should not be allocated to a partition. The last cylinder is not allocated when you choose option 2 from the **fdisk** menu. If you choose option 3, you should not allocate the last cylinder of the hard disk to the XENIX partition.

- 
15. Press `<Return>`, and you see the main **fdisk** menu again. You have now set up the partition(s) on your hard disk. To continue with the next step in the installation procedure, enter **q** and press `<Return>`.
  16. Now you see a menu from the program **badtrk(ADM)**. With the **badtrk** program, you can scan your hard disk for defective tracks. The program maps any flawed locations to good tracks elsewhere on the disk. It also creates a bad track table, which is a list of all the bad tracks on your hard disk.

The main **badtrk** menu looks like this:

- ```
1. Print Current Bad Track Table
2. Scan Disk (You may choose Read-Only or Destructive later)
3. Add Entries to Current Bad Track Table by Cylinder/Head Number
4. Add Entries to Current Bad Track Table by Sector Number
5. Delete Entries Individually from Current Bad Track Table
6. Delete All Entries from Bad Track Table
```

```
Please enter your choice or 'q' to quit:
```

Enter **2** and press **<Return>**.

17. You see the following submenu:

- ```
1. Scan entire XENIX partition
2. Scan a specified range of tracks
3. Scan a specified filesystem
```

Enter **1** and press **<Return>**.

18. After you select the area you want scanned, you are given the choice:

- ```
1. Quick scan (approximately 7 megabytes/min)
2. Thorough scan (approximately 1 megabyte/min)
```

Enter **2** and press **<Return>**.

19. You are prompted:

```
Do you want this to be a destructive scan? (y/n)
```

Enter **y** and press **<Return>**. You are warned:

```
This will destroy the present contents of the region you are scanning.
Do you wish to continue? (y/n)
```

Enter **y** and press **<Return>**. You see the following message:

```
Scanning in progress, press 'q' to interrupt at any time.
```

20. After you respond to the above prompts, the program scans the active partition of the new disk for flaws. The larger your disk, the longer the scanning process takes, so a very large disk may take a while.

As **badtrk** scans the disk, it displays the number of each track it examines, and the percentage of the disk already scanned. Pressing the **q** key at any time interrupts the scan. If you press **q** to interrupt the scan you do not need to press **<Return>**. You are then prompted to continue scanning or to return to the main menu.

3

Whenever **badtrk** finds a defective track, it lists the location of that track using both the sector number and cylinder/head conventions. Defective track information is entered into the table and displayed on the screen. An example bad track might be:

```
wd: ERROR : on fixed disk ctlr=0 dev=0/47 block=31434 cmd=00000020
status=00005180, sector = 62899, cylinder/head = 483/4
```

21. When the scan is complete, the menu reappears. Select option "1" to see the results of the scan. Your bad track table might look like this:

Defective Tracks

	Cylinder	Head	Sector Number(s)
1.	190	3	12971-12987

Press **<RETURN>** to continue

Press **<Return>** to return to the main menu.

Note

If there is a flaw in the first few tracks of the XENIX partition, you are returned to the **fdisk** utility (see the previous installation step). Repartition the disk with **fdisk** so that the XENIX partition no longer includes the defective tracks. You will have to experiment to determine how many tracks to exclude. Leave these defective tracks unassigned to any operating system. When you leave **fdisk**, **badtrk** will be run again and you should scan the disk for further flaws.

This process will continue until **badtrk** finds no flaws in the first few tracks.

22. When you are finished making changes to the bad track table, enter **q** and press (Return).
23. You are next prompted for the number of tracks to allocate as replacements for those tracks that are flawed. You should allocate at least as many as the recommended number. Enter the number or just press (Return) to use the recommended number that is displayed:

Enter the number of bad tracks to allocate space for
(or press return to use the recommended value of *n*):

If you press (Return) and do not enter an alternate value, **badtrk** allocates the recommended number of tracks as replacements. This number is based on the number of bad tracks currently in the table, plus an allowance for tracks that may go bad in the future. If you ever exceed the number of allocated bad tracks, you must reinstall XENIX.

24. Next, **badtrk** prompts:

Do you want to update this device with the new table? (y/n)

Enter **y** and press **<Return>** to save the changes. To correct any mistakes or otherwise alter the bad track table, enter **n**. Modify the bad track table to contain the desired entries, enter **q** at the main menu to return to the prompt displayed above, then enter **y** to update the device with the new table.

25. Next, you see:

Do you want to attempt to salvage any valid data on the bad tracks? [may take a long time] (y/n)

Enter **n** and press **<Return>**.

26. Next, the installation program runs the **divvy(ADM)** utility. This program allocates portions of your partitioned disk for the *root* and *swap* areas. It also allocates a small portion of the disk for a *recover* area that is used during autoboot by the **fsck(ADM)** program. **divvy** first prompts you for the swap space allocation:

There are *n* blocks in the XENIX area.
Between *x* and *y* blocks should be reserved for the swap area.

Please enter the swap space allocation, or press **<RETURN>** to get the default allocation of *z* blocks:

The actual numbers in this prompt vary depending upon the size of your disk. If you used the Disk Layout Worksheets from Chapter 1, enter the value you calculated, otherwise use the default values shown on your screen. (If you plan to install the Development System, enter a swap-area allocation that is at least 500 blocks larger than the default.) Enter your swap space allocation now or press **<Return>** to choose the default values.

If you have enough disk space for a */u* filesystem, you see the prompt:

Do you want a separate */u* filesystem? (y/n)

Installation Instructions

If you have enough storage for a separate user filesystem, and want to create one, enter **y** and press **<Return>**. If you do not wish to have one, respond **n** and skip to the next step.

Next, you see:

```
Enter block allocation for the /u file system.  
(min to max)
```

In the display, *min* and *max* are replaced with the minimum and maximum number of blocks that can be allocated for the filesystem. If you used the Disk Layout Worksheets, enter the value you calculated for the */u* filesystem. (See “Planning Your Disk Layout” for more information on the */u* filesystem.)

27. You are now prompted for block-by-block control of the partition:

```
Do you require block-by-block control over  
the layout of the XENIX partition? (y/n)
```

If you are not creating a set of special filesystems, enter **n** and press **<Return>**. Block-by-block control means you can choose the exact size of filesystems and the swap area to fit your needs. Most users do not need this kind of precise control, so answering ‘**n**’ at this prompt causes **divvy** to use default settings based on the size of your hard disk. If you wish to allocate filesystems manually, enter **y** and press **<Return>**; Appendix E “Using **divvy** with Manual Control” explains how to do this.

28. If you have a large root filesystem (40 MB+) you may be asked if you want to allocate an additional, small portion of the disk as scratch space for **fsck**. This scratch space is needed for temporary storage when checking large filesystems. Respond “**y**” and press **<Return>**. (The system uses */dev/scratch*, the filesystem created by **divvy**, when it runs **fsck** on the root filesystem. You can also use */dev/scratch* as a scratch filename at other times when **fsck** prompts for one, provided that the filesystem being checked is not larger than the root filesystem.)

29. The system now loads a rudimentary XENIX file system onto your hard disk. This takes several minutes. You see the message:

```
Making filesystems
```

30. If you have the 96tpi distribution, the Restricted Rights Legend is displayed, followed by the message:

```
Operating System Serialization
```

```
Enter your serial number  
and press <RETURN>:
```

3

Enter your serial number exactly as it is shown on your Serializa-
tion Card and press <Return>. Then you see the message:

```
Enter your activation key and press <RETURN>:
```

Enter your activation key exactly as it is shown on your Serializa-
tion Card and press <Return>. If you have the 135tpi distribution,
this step does not appear until later.

31. When the **mkdev** program is finished, the message is displayed:

```
Hard Disk Initialization procedure complete
```

Installation Instructions

The system then shuts down and displays instructions on booting the newly initialized hard disk. Make note of these instructions. You then see:

```
** Normal System Shutdown **  
  
** Safe to Power Off **  
    - or -  
** Hit Any Key to Reboot **
```

If you want to add a second hard disk, finish the installation procedure first. You will be referred to the proper instructions later in this procedure.

32. If you have the 96tpi distribution, open the floppy door, but leave the BOOT/FILESYSTEM floppy in the drive.
33. Press any key to reboot the system. You see the following boot prompt:

```
Boot  
:
```

34. If you have the 96tpi distribution, press <Return>. You see the following:

```
hd(40) xenix
```

35. If you have the 135tpi distribution, make certain the BOOT floppy (N1) is in the drive, the floppy door is closed and press <Return>.

You see the following:

```
fd (64) xenix root=hd(40) swap=hd(41) pipe=hd(40) swaplc=0 nswap=0
```

36. Next, regardless of your floppy type, you see some copyright information, and information about the memory configuration of your system.

Note

Note that this startup procedure applies only to the first time you install XENIX on your hard disk. Hereafter you need only press <Return> when you see the Boot prompt, no matter what type of floppies you use for your installation.

As before, the system performs a self-check to determine where any problems exist with the hardware. The letters A-Z appear successively on screen. If the letters displaying stop before the letter "Z" is reached, run hardware diagnostics as explained in your computer manual. Correct any identified problems and start the XENIX installation procedure again.

3

If the letters stop at this point, call the Support Center listed on the support information card and be prepared to tell them the last letter displayed.

37. Now the installation program runs the **fsck(ADM)** utility. **fsck** checks the filesystem(s) on your hard disk. In this case, it will check only the root filesystem.
38. If you have the 135tpi distribution, skip to step 42.
39. If you have the 96tpi distribution, you are prompted to insert floppy B1 at this point.
40. Follow any additional screen prompts for floppies. Note that you may not be prompted to insert all the volumes in your distribution at this time.

If you insert a floppy in the wrong order, you see this prompt:

```
Error: incorrect volume in drive!
```

Remove the floppy from the drive, insert the correct volume, and press <Return>.

Installation Instructions

If there is an error with the extraction procedure, you see the message:

```
Extraction error: try again? (y/n)
```

Make sure you are using the correct floppy, make certain the door of the floppy drive is completely closed, then enter `y` and press `<Return>`. If the message persists, see “Error reading a floppy” in the “Troubleshooting” section at the end of this chapter.

41. If you have the 96tpi distribution, skip to step 43.
42. If you have the 135tpi distribution, you see the following message:

```
Operating system serialization.
```

```
Enter your serial number  
and press <RETURN>
```

Enter the serial number exactly as it is shown on your Serialization Card and press `<Return>`. Then you see the message:

```
Enter your activation key  
and press <RETURN>
```

Enter the activation key exactly as it is shown on your Serialization Card and press `<Return>`.

You see the following message:

```
** Normal System Shutdown **  
  
** Safe to Power Off **  
- or -  
** Hit Any Key to Reboot **
```

Remove the BOOT floppy, press any key to reboot the system from the hard disk and press (Return) at the boot prompt. After the system boots, you are prompted to insert volume N3. Insert N3 and press (Return). You are then prompted for the "B" series of floppies.

43. Next, you are prompted to assign the *root* password:

```
Please assign a password for the super user account, "root".
Enter new password (minimum of 5 characters)
Please use a combination of upper and lowercase letters and numbers.
New password:
```

The new password can be any combination of letters, numbers, and punctuation marks, but should be at least 5 characters long. Enter the new password and press (Return). The system does not display the password as you enter it, so type carefully.

3

44. After you press (Return), the system displays the message:

```
Re-enter new password:
```

Enter the new password once more and press (Return). Make sure you enter it correctly, otherwise the program prompts you to enter the password again. When you have entered the password correctly, you see some information about passwords and how to change your super user password in the future.

The super user password is now in place. From now on, the password is required whenever you attempt to access the system as super user. The super user password keeps the system safe from unauthorized use. It is very easy to make errors that could destroy files when logged in as the super user. Log in as super user only to install programs and to do system maintenance tasks. For a complete description of the super user, see the "Starting and Stopping the System" chapter of the *System Administrator's Guide*.

Do not forget the super-user password. To restore a forgotten super-user password, you must reinstall your XENIX System.

Installation Instructions

45. Next, you are prompted to set the time zone appropriate to your location. This procedure is simple if you are in North America. Those in other areas should read Appendix C, "Setting Time Zones Outside North America," and set their time zone as described there. The first thing you see is:

```
Time zone initialization
```

```
Are you in North America? (y/n)
```

If you are not in North America, enter **n** and proceed to Appendix C to set your time zone. If you are in North America, enter **y**. You see the following menu:

- ```
1. NST - Newfoundland Standard Time
2. AST - Atlantic Standard Time
3. EST - Eastern Standard Time
4. CST - Central Standard Time
5. MST - Mountain Standard Time
6. PST - Pacific Standard Time
7. YST - Yukon Standard Time
8. HST - Hawaiian/Alaskan Standard Time
9. NST - Nome Standard Time
```

```
Enter the number that represents your time zone or enter q to quit:
```

If, for example, your time zone is Nome Standard time, you would enter the number **9** and press (Return).

46. You see the following:

```
Does daylight savings time (summer time) apply at your location? (y/n)
```

If daylight savings or standard time changes occur in your area, enter **y**. If not, enter **n**. The time zone variable, located in */etc/TZ*, is changed accordingly. Your time zone is now set.

47. You see this message:

```
Setting up system directories
```

48. Next, the following messages are displayed, followed by a “Current Disk Usage” summary:

You have now installed the Xenix minimum run time system. The run time system will support most application programs, so you may choose to stop installation now and preserve the maximum available disk space for user files.

The remainder of the Xenix Operating System, Development System and Test Processing sets are installable in small packages. You may either install each set entirely or selectively choose which packages of the set to install.

49. You also see a menu that gives you the option of stopping or continuing with the installation:

1. Stop installation
2. Continue installation

So far you have installed the minimal, or “Run Time” system. If you are installing the operating system merely as a platform for an application, you may not need to install more software. However, should you need to administer the system extensively, have user accounts, and use the system administration shell, you need to install more software before completing the installation. If you want to install operating system packages or applications, select option 2 and press `<Return>`.

50. This stage involves “customizing” your system, adding only those parts of the operating system distribution that are useful to you. You can also install your applications as well. The installation program `custom(ADM)` is automatically invoked to install XENIX packages and other products, including applications.

## Installation Instructions

51. The main **custom** menu is displayed:

- ```
1. Operating System
2. Development System
3. Text Processing System
4. Add a supported product
```

```
Select a set to customize or enter q to quit:
```

If you do not select a system to “customize” by choosing the ‘q’ option, you stop the installation procedure. You will need to invoke **custom** later to add other XENIX packages. To add Operating System Packages, select option “1”.

52. The following messages are displayed:

```
Installing custom data files ...
```

```
Insert Operating System (extended utilities) Volume X1
and press <RETURN> or enter 'q' to quit:
```

Insert the diskette as instructed and press (Return).

53. The following submenu is displayed:

- ```
1. Install one or more packages
2. Remove one or more packages
3. List the available packages
4. List the files in a package
5. Install a single file
6. Select a new set to customize
7. Display current disk usage
8. Help
```

```
Select an option or enter q to quit:
```

Select option “1”. A list similar to the one in your *Release Notes* is displayed. Enter the name of each package separately, unless you select ALL. When you enter the name of a package, **custom** prompts for the necessary volume numbers. Insert the appropriate floppies and follow the screen prompts. For more information on installing portions of the Operating System, see **custom(ADM)** in the *XENIX Reference*.

54. You are instructed to insert Volume X1. Because this volume is already in the drive, press `<Return>`.

Refer to “Packages in This Set” in your *Release Notes* for a list of operating system packages. The only software presently installed on your system is the RTS package, or Run-Time System. This represents the minimal software needed to have a functioning operating system. Other packages can be added that suit your needs; if your system is merely a platform for applications such as spreadsheets, word processors, or databases, you should install the following packages:

|        |                                                     |
|--------|-----------------------------------------------------|
| SYSADM | a user-friendly interface for system administration |
| BACKUP | utilities to perform backups                        |
| LPR    | the line printer spooler                            |
| MAIL   | mail program for sending messages to other users    |

3

In addition, packages such as BASE, EX, and CSH are useful for those who wish to work in a traditional XENIX environment, using the `ex(C)` or `vi(C)` file editors, various file utilities, and the C-shell (`csH(C)`).

---

### Note

We recommend that you install the EX, FILE, BASE and LINK packages, as these contain useful utilities for maintaining your system. The EX package is recommended because it contains the `vi` editor, and the documentation uses this as an example editor when you are asked to edit default files. Some products (such as network software) have drivers that require linking into the kernel, so you should install the Link Kit (LINK).

---

If you have a large hard disk and space is not a problem, you can select ALL and install every package of the operating system.

55. If you choose to install the TERMINF package (`terminfo(M)`), you are prompted as to whether you want the `terminfo` database compiled at this time (a lengthy process). If you choose to defer this until later, you are given instructions for doing so.
56. If you choose to install ALL or the BACKUP package with `custom(ADM)`, you are prompted for information about the filesystems you configured on your system during installation. This information is used to set up a schedule for backing up your system

## Installation Instructions

files. See the “Backing Up Filesystems” chapter of the *XENIX System Administrator's Guide*.) The prompts are displayed after the files are loaded and appear as follows:

```
Does this installation consist of only a root filesystem?
```

If you respond **y**, then the **schedule** file is automatically modified to only back up a root filesystem.

If you respond **n**, you are prompted:

```
Does this installation consist of only the root filesystem
and a user filesystem named /u?
```

If you respond **y**, a message is displayed indicating that the default **schedule** is suitable for your system and need not be altered.

If you respond **n**, you are indicating that you have additional filesystems, which requires manual editing of the **schedule** file.

In addition to the filesystem question(s), you are prompted to create a password for the *backup* account. When you log in as **backup**, you are taken directly to the **sysadmin**(ADM) menu.

57. You can install your applications using the “Add a Supported Product” option from the main **custom** menu. You are prompted for various volumes (including parts of your operating system distribution) just as you were for XENIX packages.
58. If you plan to install device drivers, you should install the Link Kit (LINK) using the “Add a Supported product” option of the main **custom** menu. Some products also have drivers that require linking into the kernel, so you should install the Link Kit now. Note that you are prompted for your serial number and activation key again. Enter them as you did earlier in the installation.

59. When you are finished installing XENIX packages and any products, quit out of **custom** by entering **q** at the main menu. You see:

```
AFTER you see the message '** Normal System Shutdown **
reboot the system by opening the floppy door and pressing any key.
The screen will clear and you will see the boot message:
 Boot
 :
Press <RETURN> to reboot and continue the installation.
```

The system shuts down. Remove any floppy that is in the drive.

3

60. Press any key to reboot the system and press <Return> when the boot prompt appears:

```
Boot
:
```

The following is displayed:

```
hd(40) xenix
```

The screen clears and you see some self check diagnostics.

61. Next, you are prompted to continue the startup or perform system maintenance:

```
Enter CONTROL-d to proceed with normal startup
(or give root password for system maintenance)
```

Because there are some system administration steps you should take now, enter the root password and press <Return>.

The next thing you see is the root prompt, also used in system maintenance mode, which is a pound sign:

```
#
```

## Installation Instructions

62. If you wish to add a second hard disk, follow the instructions in “Adding a Second Hard Disk” in the “Managing Filesystems” chapter of the *XENIX System Administrator's Guide*.
63. If you created additional filesystems on your hard disk (such as */u*) earlier in the installation, follow the instructions in Appendix F, “If You Created Additional Filesystems,” before proceeding. This appendix prepares the additional filesystem(s) for use.
64. You should now add user accounts to the system, if desired, using the **mkuser**(ADM) program. If you have created a */u* filesystem for user accounts, you must first mount the filesystem before creating user accounts. Enter the following command:

```
mount /dev/u /u
```

Each person using the system should have a login account and password. The security of your system is greatly dependent on the restrictions you impose on the use of accounts. For more information, see the “Administering User Accounts” chapter of the *System Administrator's Guide*. You can also alter the defaults used to create accounts and administer them.

65. Now that you have installed your software and created your user accounts, you should now exit maintenance (single-user) mode by pressing **<Ctrl>d**. This logs you out and brings the system up in multi-user mode.

Your system is now ready for users to log in.

## **Chapter 4**

# **Reinstalling Your System**

---

**Introduction** 4-1

**Shut Down the System** 4-3

**Back-up the System** 4-4

**If You Plan to Preserve Non-root Filesystems** 4-5

**Reinstallation Procedure** 4-6

**Reinstall Packages and Applications** 4-13

**Relink the Kernel** 4-14

**Prepare Preserved Filesystems for Use** 4-15

**Restore Your Backups** 4-16

**Reestablish Your Second Hard Disk** 4-17



---

# Introduction

The following procedure explains how to reinstall your system. It does not apply if you are installing XENIX for the first time.

It is necessary to perform a safety backup of your system first (if possible) and, when finished with the reinstallation, relink your kernel if you are using any special device drivers.

There are at least two reasons for performing a reinstallation:

- You need to enlarge your swap space or increase the size of your bad track table.
- You need to reinstall your present OS because your hard disk has become corrupted, will not boot, and you have no Emergency Boot Floppy and no backup tapes to restore.

4

The reason you should use this chapter rather than the *Installation Procedure* is that the prompts generated during a reinstallation differ from a normal installation (due to existing filesystems).

If you have turned to this chapter because your hard disk will not boot and you have an Emergency Boot Floppy plus complete backups of your system, you should follow the procedure “Restoring a Corrupted Root Filesystem” in the “Solving System Problems” chapter of the *XENIX System Administrator’s Guide*.

If you are planning to enlarge your swap space or increase the size of your bad track table, you can preserve existing non-root filesystems only if they are on a secondary hard disk. You must back up (using `tar(C)`) all filesystems that are on the active XENIX partition. The other partitions will not be affected unless you change the size of the XENIX partition. During installation, you must recreate filesystems at least as large as your backups require and restore them when the reinstallation is complete.

### *Note*

If you are planning to enlarge one filesystem at the expense of another, you cannot use backup volumes created using **backup(ADM)** to restore the entire reduced filesystem, even if the filesystem was not full when the backup was made. You can restore individual files from your backups (using the “Restore backed up file(s)” option of **sysadmin(ADM)**) but you will not be able to restore an entire filesystem (as with the “Restore an entire filesystem” option of **sysadmin**). You should keep your backups in case it is necessary to restore individual files and directories, but make **tar** backups as instructed in this chapter and use them.

---

Reinstallation does not affect any other operating systems (or partitions), such as MS-DOS, that currently share the hard disk, unless you choose to change partition sizes. Refer to the “Using DOS and OS/2” chapter in the *XENIX System Administrator's Guide* if you want to change the size of your XENIX partition.

The subsections that follow describe each stage of the reinstallation process:

1. Shut the system down to Single User (System Maintenance) Mode.
2. Make a full backup of your system.
3. Perform the reinstallation.
4. Relink the kernel, if necessary.

If you have local system modifications to the kernel, such as additional device drivers, you must relink these into the new link kit after you reinstall XENIX.

---

## Shut Down the System

You should be the only person logged in to your computer when you perform the reinstallation. Make sure any other users know what you are going to do, and when you are going to do it. They may wish to make their own backups before you bring down the system.

Also, advise users to remove unnecessary files. This makes the whole process faster and requires fewer floppies, if they are your only backup media.

To shut the system down and enter single user (maintenance mode), first login as **root** and enter the following command:

```
/etc/shutdown su
```

Give the users five minutes to log off. When you are prompted to enter <Ctrl>d or the root password for system maintenance, enter the root password. This puts you into single-user mode, and your prompt is a "#".

4

---

# Back-up the System

If you are reinstalling with the intention of resizing filesystems, you must first make a full backup of the system. If you are reinstalling because the disk is corrupted and cannot be saved, this step is skipped, of course. Use the **tar(C)** command. Refer to **tar(C)** in the *XENIX Reference*. You cannot use any other filesystem backup utilities.

---

### Note

**tar** does not archive empty directories or device files.

---

Depending on the archive device on your system, you would use a command similar to the following, which uses the cartridge tape drive:

```
tar cvf /dev/rct0 .
```

This creates a backup of the entire root file system. In addition, you should make separate backups of all non-root filesystems. (*/u* is an example of a non-root filesystem.) If you are still in single-user mode, you need to mount them first, for example:

```
/etc/mount /dev/u /u
```

This mounts the */u* filesystem.

You should make a backup immediately before reinstalling so that you have the latest possible version of your system. This is especially important in the case of user files, which can change frequently. Once again, remove any old, unnecessary files. This makes the whole process faster.

You can use any backup media, including floppy and tape drives, to save your files. However, if you use a special driver with your backup device, you cannot use that device immediately after installing the new kernel, since it is not yet linked with the required device driver. This is important, since you can only read information from the default floppy drive until you relink the kernel.

If you are reinstalling XENIX, make certain you have the original distributions of any applications packages you use. You must reinstall those software packages after you reinstall the XENIX system.

## If You Plan to Preserve Non-root Filesystems

Note that you can leave your special filesystems on your primary hard disk untouched during the reinstallation. You must follow certain precautions, however. When the root filesystem is created anew, all the device nodes and the mount points are destroyed, but the actual files and divisions are still intact. (The device node is the device name that refers to the filesystem, such as */dev/u*. The mount point is the directory that the device is “mounted” upon, such as */u*.) During the reinstallation, you will be put into the **divvy** manual control menu and be asked to “name” each of your filesystems over again, which will recreate the device nodes and mount points. You are cautioned to take care not to obliterate or change the filesystems in any way other than to give them names once again using the “n” option. The manual control menu has an “r” (restore) option to undo any changes, but you are warned to take care during this part of the procedure.

---

# Reinstallation Procedure

This section describes how to reinstall your system.

1. You should have your complete XENIX distribution, applications, device drivers, etc. on hand before beginning. Make certain you have your serialization codes and activation keys as well.
2. You should have performed a complete safety backup of your system as instructed in "Back-up the System."
3. If you are increasing your swap space or the size of your bad track table, you will need to recreate your non-root filesystems. If you are preserving any filesystems, you need to rename your non-root filesystems. In order to do either, you must record your present **divvy** table. Enter the following command:

```
/etc/divvy -c 1 -b 1 -p 0
```

Copy the table entries displayed on your screen into the blank table below (you need not copy the *New File System?* column):

### Primary Hard Disk Filesystems

| Name | New File System? | # | First Block | Last Block |
|------|------------------|---|-------------|------------|
|      |                  |   |             |            |

Enter "q" to quit out of **divvy**, followed by "e" when asked if you want to exit.

## Reinstallation Procedure

4. If you have a secondary hard disk with existing XENIX filesystems, they will not be touched by the reinstallation, but you must record the filesystems in your present **divvy** table and later rename the divisions after the reinstallation is complete. Enter the following command:

```
/etc/divvy -c 1 -b 1 -p 1
```

Copy the table entries displayed on your screen into the blank table below (you need not copy the *New File System* column):

### Secondary Hard Disk Filesystems

| Name | New File System? | # | First Block | Last Block |
|------|------------------|---|-------------|------------|
|      |                  |   |             |            |

4

Enter “q” to quit out of **divvy**, followed by “e” when asked if you want to exit.

5. After you have notified users that your system will be down for some time, log in as **root** and use the **shutdown(ADM)** command to halt the system (if you are in single-user mode, just enter **haltsys** at the root prompt):

```
/etc/shutdown 5
```

6. When the system halts, insert the **BOOT** floppy (N1) into the drive and press any key to reboot the system. The system boots from the floppy and the usual messages are displayed. Insert volumes as instructed and select your keyboard nationality as usual.
7. After the hard disk initialization message is displayed, you see the following:

```
During installation you may choose to overwrite all
or part of the present contents of your hard disk.
Do you wish to continue?(y/n)
```

You must respond **y**.

## Reinstallation Procedure

8. Unless you have installed a new hard disk, when the "Hard Disk Drive Configuration" menu appears, simply quit out of it.
9. Next, the **fdisk** menu is displayed:

- ```
1. Display Partition Table
2. Use Entire Disk for XENIX
3. Create XENIX Partition
4. Activate Partition
5. Delete Partition
```

```
Enter your choice or 'q' to quit:
```

Unless you need to change the existing partitions on the disk, you can simply quit out of **divvy** menu. If you wish to resize your XENIX partition and you have other existing partitions, enter **3** and give the starting and ending blocks of the XENIX partition. Be sure not to overlap your other partition(s). Otherwise, enter **2** to use the entire hard disk for XENIX. You have now set up the partition(s) to use XENIX on your hard disk. To continue with the next step in the installation procedure, enter **q** and press (Return).

10. For standard hard disks, is this message is displayed:

```
This device contains a valid division table. Additional
(non-root) filesystems can be preserved across this reinstallation.
If you wish to be able to preserve these file systems later, you must
not change the current limit of the bad track table, which is
n bad tracks. Do you wish to leave it unchanged? <y/n>:
```

If you are reinstalling in order to increase the size of your bad track table, you must select **n**. On the other hand, if you intend to preserve existing filesystems, you must respond **y**; this ensures that the limit of your bad track table is not increased, which would destroy your filesystem(s). You can still add *entries* to your current bad track table, but you will not be allowed to increase the size of the table itself.

If your hard disk is attached to an SMS-OMTI controller or a SCSI adapter, skip to step 12. Otherwise, **badtrk** is run next. You should select to run a thorough non-destructive scan of the XENIX partition.

If you specified at step 10 that you did not want to change the current limit of the bad track table then you are not prompted to enter a new limit. Otherwise, you are next prompted for the number of tracks to allocate as replacements for those tracks that are flawed. You should allocate at least as many as the recommended number.

11. If you still have a valid `/u` filesystem, **divvy** will prompt for the option to preserve it across the reinstallation. As discussed in step 10, this occurs only if you did not increase the maximum number of bad tracks on your hard disk or shrink the size of the XENIX partition using **fdisk**:

```
This device contains a valid disk division table.
Your additional (non-root) filesystems
can be preserved across the reinstallation.
Do you want to preserve your additional filesystems? (y/n)
```

If you do not wish to preserve your filesystems, enter **n** and proceed to the next step. Otherwise, you should repond **y**. **divvy** displays the following warning:

```
If you choose to manually adjust the sizes of the filesystems,
do not change the bounds of any filesystem you wish to retain.

If you choose not to manually adjust the sizes of the filesystems,
your original disk layout will not be modified.
```

(This message does not appear on XENIX-286 distributions.) **divvy** then proceeds with the manual control procedure. Skip to step 13.

Reinstallation Procedure

12. **divvy** prompts for the swap space allocation:

```
There are n blocks in the XENIX area.  
Between x and y blocks should be reserved for the swap area.  
  
Please enter the swap-space allocation, or press <RETURN>  
to get the default allocation of z blocks:
```

Enter your swap space allocation or press (Return) to choose the default value.

If you have a large hard disk, you see the prompt:

```
Do you want a separate /u filesystem? (y/n)
```

We recommend that you have a separate filesystem with disks larger than 40-50Mb, especially if you anticipate having many active users. You should leave at least 15M bytes for the *root* filesystem. Check the “Planning Your Disk Layout” section of Chapter 1 for more information on a */u* filesystem.

If you have enough storage for a separate user filesystem, and want to create one, enter *y* and press (Return).

Next, you see:

```
Enter block allocation for the /u file system.  
(min to max)
```

In the display, *min* and *max* are replaced with the minimum and maximum number of blocks that can be allocated for the filesystem.

13. Next, you are prompted for precise control over the layout of your filesystem:

The layout of the filesystems and swap area is now prepared. Do you wish to make any manual adjustments to the sizes or names of the filesystems or swap area before they are created on the hard disk? (y/n)

(This prompt refers to “block-by-block” control on XENIX-286 distributions.) If you are preserving or recreating any filesystems, you must respond **y**. If you are not recreating or preserving filesystems, respond “**n**” and proceed to step 14. The manual procedure is described in Appendix E “Using *divvy* with Manual Control.”

If you are preserving your filesystems, refer to the table you recorded on page 4-6. Note that the name column may be conspicuously different from your original table, unless you have only one non-root filesystem called **/u**. Skipping root and swap, you should rename all of your old filesystems (including **/u**), starting with division 2. Do not enter any command other than “**n**” or “**r**”. When you are finished renaming divisions, proceed to step 15.

4

If you are recreating your filesystems, use most of the data from the table you recorded on page 4-7 to re-create filesystems at least as large as your original ones, so that your backups will fit within the space allocated. To choose a command, enter the first letter of the command, then press **<Return>**. If you are enlarging one filesystem, you must reduce another. If you are increasing the size of your root filesystem, for example, you could first reduce another filesystem with the “**e**” command, then enlarge the root filesystem with the “**s**” command.

Note that if any of the divisions overlap, an error message will appear when you try to exit and you will be returned to the *divvy* menu to correct the situation.

Note

You can use the “**r**” or “restore” command to restore the original partition table if you make a serious mistake and want to start over.

Reinstallation Procedure

14. After you quit from **divvy**, you may be prompted for a */dev/scratch* file. When the filesystems are being created you see:

Making filesystems

Depending on your media type, you may also be prompted for operating system serialization. Follow the prompts.

15. Finally, you are asked to reboot the system. Follow the prompts. When you reboot the system, the usual copyright and configuration information is displayed, followed by the A-Z self check.
16. Depending on your media type, you may then be prompted for operating serialization (if you were not already asked to do so), and **fsck** checks your filesystems.
17. You are next prompted to insert additional floppies to complete installation of the Run Time System.
18. When you see the prompt to stop or continue the installation, select **2** to continue.

Reinstall Packages and Applications

Now that the Run Time System is installed, the **custom** menu appears and you should reinstall your packages, applications and device drivers just as you did when you installed them originally.

Relink the Kernel

If you have special device drivers, you need to link them into the XENIX kernel when you are finished installing. In this case, make sure you install the link kit software when you reinstall XENIX. See **custom(ADM)** for information on extracting packages from XENIX System V distributions.

When you are finished installing, make sure you are in single-user, or system maintenance mode.

Prepare Preserved Filesystems for Use

In order to use the filesystems you preserved on your root hard disk, you must create the directories that they are mounted upon. Use the **mkdev** command to ensure that the filesystems you preserved are correctly set up:

```
mkdev fs /dev/name /directory
```

You will also be asked if you wish to have this filesystem mounted automatically when the system starts up multiuser. You should repeat this command for each of your non-root filesystems you preserved. For example, if you named a filesystem **/u**, you would enter the following:

```
mkdev fs /dev/u /u
```

In this example, the **/u** directory will be created at the mount point for **/dev/u**, along with a lost+found directory to facilitate cleaning the filesystem.

To mount and unmount the filesystem, enter these commands, respectively:

```
mount /dev/name /directory
```

```
umount /dev/name
```

Restore Your Backups

You should restore your backups while still in single user mode. If some of your backups were created by an update script, use the command line specified in the *Release Notes* to restore them. For example, if your backups are on a cartridge tape drive, you would use this command:

```
tar xvf /dev/rct0
```

Don't forget to mount your non-root filesystems before restoring their respective backups.

Reestablish Your Second Hard Disk

This section explains how to prepare your second hard disk to function with your reinstalled system:

1. Enter single user mode.
2. Use the **fixperm** utility to create the necessary device nodes:

```
cd /
fixperm -c -dHD1 /etc/inst.perms
```

If you do not know what **fdisk** partitions your XENIX filesystems are in, run **fdisk**:

```
fdisk -f /dev/rhd10
```

Enter “1” at the main **fdisk** menu to display the partition table. Write down the numbers of the partitions that contain XENIX, then exit from **fdisk**.

3. Invoke **divvy** with the following command line:

```
divvy -b 1 -c 1 -p 1 -v partition-number
```

In the above command line, substitute the actual partition numbers that you copied from the **fdisk** table. Follow this **divvy** procedure for all of the possible **fdisk** partitions.

Examine your **divvy** table. You should see a list of **divvy** divisions, which may contain valid filesystems, that do not have names. Only the **divvy** division number and sizes are displayed.

Next, use the table of “Secondary Hard Disk Filesystems” you recorded on page 4-7 to rename each of the divisions to their original names. Use the “n” (name) command. If you make a mistake, use the “r” (restore) command to undo any changes you made. When you have named all of your **divvy** divisions, enter “q” to quit from **divvy**.

Reestablish Your Second Hard Disk

4. At this point you are given the choice to retain your changes or exit:

```
i[nstall]   Install the division set-up shown
r[eturn]   Return to the previous menu
e[xit]     Exit without installing a division table
```

Please enter your choice:

If you wish to keep your changes, enter "i". If you have made a mistake, enter "r" to return to the main **divvy** menu.

5. Finally, use the **mkdev** command to prepare your preserved filesystems for use:

```
mkdev fs /dev/name /directory
```

In the above command, substitute the name of the actual device and the mount directory. For example:

```
mkdev fs /dev/u /u
```

The filesystems are now ready for use.

Appendix A

Troubleshooting Your Installation

Introduction A-1

Solving Problems During Installation A-2

System Does Not Boot from the N1 Floppy A-2

Error Reading a Floppy A-2

System Reboots from N1 or Screen Blanks After Boot:
Prompt A-3

Hardware Self-Check Stops at H6 A-3

System Does Not Boot from the Hard Disk A-4

No floating point emulator found in /etc/emulator A-5

Solving Problems After Installation A-6

Hardware Does Not Work Under the XENIX System A-6

XENIX System and DOS Do Not Work A-6

Problems Booting on a Disk with Greater Than 1024
Cylinders A-6

Common System Resource Error Messages A-8

Solving Undocumented Problems A-9

Hardware or Software? A-9

Hardware under the XENIX System and MS-DOS A-9

Identifying Your Problem A-10

Before Calling for Help A-13

A Description of the Problem A-13

System Configuration A-13

Introduction

Sometimes errors occur during the installation procedure. These may happen as you install the software, or they may cause problems when you boot immediately after installation. Most often, there is no problem with the software or the procedure itself. Occasionally there is a problem with the hardware, but most are minor, such as improperly connected cables. This appendix describes what to do when some common difficulties arise. If you have difficulty installing the software, this appendix describes some of the most common problems, how to avoid them, and how to fix them if they happen.

If you do not find your problem described in the sections that follow, use the “Solving Undocumented Problems” section of this appendix to try and identify your problem. If your system hangs mysteriously, this section can help eliminate some of the more common difficulties. (Additional troubleshooting information is also found in the “Troubleshooting Your System” chapter of the *System Administrator’s Guide*.) If you cannot solve your problem and you must contact your provider for help, be sure and record your system information as described in “Before Calling for Help” at the end of this appendix.

During the installation procedure, do not assume that you know what is about to happen, even if you have installed the system before. Read the installation documentation closely.

A

Solving Problems During Installation

This section covers some common problems that you may have while installing the software.

System Does Not Boot from the N1 Floppy

If the system does not boot from the N1 floppy as described at the beginning of the installation procedure, check the following:

1. Verify that the N1 floppy is in the drive and that the door to the floppy drive is completely closed. If another floppy is in the drive, the system does not display an error message; the system simply does not boot.
2. If the N1 floppy still does not boot, verify that your floppy drive works by booting an MS-DOS disk.
3. If the floppy drive works with the MS-DOS disk, you may have damaged your N1 floppy. If this is the case, and you do not have a backup copy, call your provider to arrange for a new N1 floppy.

Error Reading a Floppy

When an error occurs on a floppy during installation, sometimes you see uppercase E's repeated on the screen. In other cases an error message is displayed. If an error occurs while reading a floppy, check the following:

1. Verify that the correct disk is in the floppy drive.
2. Make sure that the floppy is inserted correctly. (See your owner's manual if you do not know how to insert the floppy correctly.)
3. Verify that the door to the floppy drive is completely closed after you insert the floppy.

4. If you still see the read error message, tap the edge of the floppy disk lightly against a hard surface, such as a table top. This should be done carefully because you can damage the floppy. Try inserting the floppy disk again.
5. If the error message persists, call your provider to get a new floppy.

System Reboots from N1 or Screen Blanks After Boot: Prompt

If you cannot install because the system keeps rebooting from the N1 floppy or the screen blanks after pressing <Return> at the Boot: prompt, one of two known situations may be causing this problem:

- an improperly configured EGA adapter
- an incompatible VGA adapter

If you have an EGA or VGA adapter, check the following:

- Verify that the the adapter is supported. Check the supported hardware Appendix in the *Release Notes*.
- Make sure that the switches on the adapter are set correctly.

Many cards have emulation modes; you should use the card in its native mode. You can also experiment with the EGA/VGA/CGA/MONOCROME switches.

If the card has an AUTOSWITCH feature, the system can also hang when H6 is displayed during boot. See the next section, “Hardware Self-Check Stops at H6” for information on correcting this problem.

Hardware Self-Check Stops at H6

When you start the installation from the N volumes, the screen displays information about your hardware configuration, followed by a series of diagnostic letters. These letters display quickly and overwrite each other, so they are not normally visible unless the startup process halts unexpectedly. If you have an EGA or VGA adapter and the screen display stops at H6, the adapter card may be improperly installed. Shut the computer off and check your adapter’s documentation.

A

Solving Problems During Installation

If your graphics card has the AUTOSWITCH feature, it must be disabled to install the XENIX system. The documentation included with your graphics card should indicate how to disable AUTOSWITCH. This is typically controlled by setting a small switch on the card to the off position. You can then restart the installation from the beginning. If the problem persists, contact your provider for support.

Another symptom of this problem is observed when the boot prompt is displayed:

```
Boot
:
```

If you are using an AUTOSWITCH EGA card that normally displays in high resolution (EGA or VGA mode) and this prompt is displayed in low resolution (CGA mode), reset the machine until the high-resolution display appears. (Standard CGA text is low-resolution; dark lines are visible through the characters. EGA and VGA text is high resolution; no dark lines are visible.)

System Does Not Boot from the Hard Disk

If the system does not boot from the hard disk as described near the end of the installation procedure, you may see a message such as this, or another kind of error message:

```
kernel: PANIC: iinit
```

If you have an ST506 or ESDI disk and you did not run a bad track scan during installation and the boot block was written on a bad track, the system does not boot from the hard disk.

Whatever the reason for this problem, you must reinstall your XENIX system.

If the operating system still does not boot, run the hardware diagnostics that came with your computer and hard disk.

No floating point emulator found in /etc/emulator

If your installation fails with the error this error message, it is probably because of a problem with dual floppy drives. Certain machines with two floppy drives have setup programs which allow you to logically swap the primary and secondary floppy drives, making what physically is the secondary floppy drive act as the primary floppy drive and vice versa. This is particularly useful if the primary floppy drive is a 3.5 inch drive and the secondary drive is a 5.25 inch drive. You can probably install XENIX with 5.25 inch diskettes even though your primary floppy drive is a 3.5 inch drive. Because XENIX bypasses the information in the BIOS that states the drives have been swapped, XENIX attempts to read the floppy filesystem from the primary drive. Because */etc/emulator* is the first file read from the floppy, the error message is displayed.

You must physically switch the drives to make what initially is the secondary floppy drive, the primary floppy drive. Remember to reconfigure the system with the setup program to inform it of the physical change. Also remember to clear the swap flag, which is used to swap drives virtually instead of physically. (This flag works under MS-DOS, not XENIX.)

A

Solving Problems After Installation

This section describes some common problems that you may encounter when you boot the system immediately after installing the software.

Hardware Does Not Work Under the XENIX System

If some hardware (for example, a tape drive) does not seem to work with the XENIX system installed, although it works fine under another operating system, the hardware may not be supported by the XENIX system.

Refer to the “Compatible Hardware” section in the *Release Notes* for information on the hardware that you can use with the XENIX system.

XENIX System and DOS Do Not Work

If you install the XENIX system after installing DOS, and both operating systems do not work, you need to back up your DOS files and reinstall both systems. You should make certain that you followed the guidelines found under “Creating and Formatting a Physical DOS Partition” in chapter 1 of this guide.

Problems Booting on a Disk with Greater Than 1024 Cylinders

If your hard disk has more than 1024 cylinders and your system hangs mysteriously or displays one of the following error messages at boot time:

```
bn void number  
  
/boot not found  
  
xenix not found  
  
Bad magic number
```

The problem may be that the boot information lies beyond the 1024 cylinder boundary.

Solving Problems After Installation

If this happens, you must reinstall your XENIX system, run the manual hard disk initialization, and rearrange your filesystems so that the root filesystem is located within the first 1024 cylinders.

Because this limitation is expressed in cylinders and XENIX **fdisk**(ADM) uses tracks, you should determine the number of tracks per cylinder for your disk. To do this, follow the instructions in “Converting fdisk Numbers” in Chapter 1, “Before You Start” in this guide.

If your disk does not have more than 1024 cylinders, these error messages may indicate other problems. See the section on solving boot problems in the “Troubleshooting Your System” chapter of the *XENIX System Administrator's Guide*.

A

Common System Resource Error Messages

When system limits are exceeded, the operating system advises you by displaying messages on the console. Some of the messages are advisory only. Others precede a system panic: the system displays some additional diagnostic messages and then “hangs,” requiring you to reboot. You can expand these limits by reconfiguring values known as *tunable kernel parameters*. The kernel should not be reconfigured because a system resource error message was received once, or even a couple of times, but when a single message persists between system sessions.

Table A.1
Error Messages and Associated Tunable Parameters

Error Message	Parameter	Category
iget - inode table overflow	NINODE	3
File table overflow	NFILE	3
No more processes (system-wide)	NPROC	4
No more processes (per-user)	MAXUPRC	4

If you encounter any of the messages listed in Table A.1, note the name of the parameter and the category number. Read the instructions found in “Reallocating Kernel Resources with configure” in the “Tuning System Performance” chapter of the *XENIX System Administrator’s Guide*. If you need to adjust a resource, first try to increase the value by a small amount. If the problem persists, increase it by 100 percent or more of its original value. If the problem is still not solved, more detailed research is required to locate the exact program and sequence that caused the error.

Solving Undocumented Problems

If your problem is not described in this appendix, or in “Troubleshooting Your System” in the *XENIX System Administrator’s Guide*, this section can help you to isolate, identify, and solve the problem.

Hardware or Software?

It is important to characterize a problem as software or hardware. In general, hardware problems are intermittent, but software problems are consistent and easily reproduced.

Hardware under the XENIX System and MS-DOS

The fact that a given piece of hardware works under MS-DOS is no guarantee that it will work with the XENIX system. A critical difference between the XENIX system and DOS is the way they access hardware. One way DOS or an application under DOS can read or write information to a device controller (disk controller, tape controller, serial/parallel etc.) is to make a call to the BIOS (Basic Input Output System) which then takes the data and reads/or writes it to the appropriate device. Except for very short periods of time during installation and during the boot process, the XENIX system never uses the BIOS.

DOS or an application makes a device-independent call to the BIOS to transfer data. The BIOS takes this device-independent call and translates it into a device-dependent set of instructions to transfer data to or from a particular device. Under the XENIX system, device drivers handle many of the functions for which DOS can use the BIOS. By writing code that reads/writes directly to the device, DOS can also use devices without going through the BIOS. The ability to write directly to a device is one reason why some hardware works under DOS, but not under the XENIX system. Because the XENIX system does not make use of the BIOS, it expects a piece of hardware to be at a specific interrupt, DMA channel and base address, and only recognizes the hardware if it is configured properly.

A

Identifying Your Problem

If the system hangs mysteriously, whether at boot time, during installation, or shortly thereafter, use the steps below to isolate and identify the problem:

1. Confirm that your hardware is listed as supported in the *Release Notes*. If not, we recommend that you use supported hardware.
2. If possible, see if a suspect component works on another machine with the same configuration (under the XENIX system, not MS-DOS).
3. If any third-party drivers are installed, such as those supplied with multiport cards, remove them, relink the kernel and see if the problem persists.
4. Make certain that your devices are recognized at bootup. Watch the boot display, check `/usr/adm/messages`, or use the `hwconfig(C)` utility.
5. Check for hardware conflicts between components, including DMA, interrupts, and memory addresses. The information displayed by `hwconfig` or found in `/usr/adm/messages` can be useful in locating a conflict. Figure A-1 is a sample display from `/usr/adm/messages`. This display shows no conflicts in the address, vector, or dma columns. The display for your machine may indicate a conflict. Although this display is useful, the hardware settings should be double-checked to make certain they are in agreement. Be certain you have checked for documented incompatibilities or limitations in Appendix A of your *Release Notes*. Under XENIX-386, the command `hwconfig -ch` is useful for detecting conflicts.

device	address	vector	dma	comment
%serial	0x03F8-0x03FF	04	-	unit=0 type=Standard nports=1
%serial	0x02F8-0x02FF	03	-	unit=1 type=Standard nports=1
%floppy	0x03F2-0x03F7	06	2	unit=0 type=96ds15
%console	-	-	-	unit=ega type=0 12 screens=68k
%parallel	0x0378-0x037A	07	-	unit=0
%tape	0x0338-0x033C	05	1	type=W
%disk	0x01F0-0x01F7	16	-	type=W0 unit=0 cyls=286 hds=16 secs=63

Figure A-1 Sample Boot Display

Solving Undocumented Problems

6. If your machine has features such as shadow RAM or memory caching, disable them. These features can have unpredictable results. If you have a caching controller, for example, you can try disabling the caching using your computer's setup program until after the XENIX system is installed. If XENIX does not work on your machine with caching enabled, this is a hardware problem and you should contact your hardware vendor for repair or replacement.
7. If you have a DOS partition installed, be sure it follows the guidelines found under "Creating and Formatting a Physical DOS Partition" in Chapter 1 of this guide.
8. If your hard disk has more than 1024 cylinders, it must observe the restrictions found under "Installing on Disks with Greater Than 1024 Cylinders" in Chapter 1 of this guide. See "Problems Booting on a Disk with Greater Than 1024 Cylinders" in this appendix for more information.
9. If you have a QIC-40 or QIC-80 mini-cartridge tape drive installed, the system can fail unpredictably if it is not installed correctly. Check the following:
 - the tape device must not be configured as the third device on the controller. (This is not supported under the XENIX system as it is under MS-DOS.) You should always configure the drive as the second device, unless you have a 4-device controller, where it can be the fourth device.
 - the jumpers on the drive must be configured properly.
 - you must use your computer's setup program to indicate there is no floppy device attached.
10. If your video card is incorrectly installed, the system can also fail unpredictably. Check the following:
 - check the section "System Reboots from N1 or Screen Blanks After Boot:" under "Solving Problems During Installation" in this appendix.
 - there are known conflicts between video cards and network cards. Attempts to send or receive data from the network card can be blocked. This problem can be resolved if you avoid using IRQ2 (interrupt vector 2) for your network card.

A

Solving Undocumented Problems

Some graphics cards use the additional interrupt, causing the conflict to occur. Check the Appendix A of your *Release Notes* for warnings that apply to specific cards and card combinations.

- Known conflicts exist between some 16-bit VGA boards and floppy or tape data transfer. We recommend that you use 8-bit VGA boards, or that you put 16-bit boards in 8-bit mode. Consult your *Release Notes* and your hardware documentation for more information.
11. There can be conflicts involving hardware clock rates. Check your components against the clock rate for your computer and the components that you have installed. Try lowering the machine speed.
 12. If you did a low-level format of your hard disk with a third-party format program, you may have a format that does not work. Format programs that are known to work are DOS Debug and Speedstor.

Before Calling for Help

If your problem is not discussed in the troubleshooting information in this chapter or “Troubleshooting Your System” in the *XENIX System Administrator’s Guide*, and you cannot isolate the problem using “Solving Undocumented Problems,” you can call your provider for help. Before doing this, you should enter critical information in the tables included in this section so that your problem can be analyzed and dealt with quickly. This includes a description of your problem and your system configuration.

A Description of the Problem

You should write down a complete description of your problem, including the precise series of commands or steps taken that lead to the problem. Include any error messages displayed; write the error message down exactly as it appears, complete with any punctuation and upper-case characters.

A

System Configuration

Knowing the hardware and software configuration of your system is vital to a correct diagnosis of your problem. The process is greatly speeded up if this information is on hand when speaking to your provider. List all the hardware components, including their brand names and model numbers. You should also list all applications and third-party device drivers you have installed.

Appendix B

Creating an Emergency Boot Floppy

Introduction B-1

Creating the Floppy or Floppies B-2

Introduction

The Emergency Boot Floppy (or floppy set for low density diskettes) is an important tool used in administering your system. It allows you to recover your system in the event of a catastrophic system failure, where the computer does not respond when you attempt to start it. You can use these diskettes to restore a corrupted root filesystem without reinstalling the operating system. If you have more than one system, you should make one Emergency Boot Floppy for each machine. Because each machine has a unique emergency set, a set made on one system does not work with any other system. Be sure to keep these diskettes separate; if you use an emergency floppy set on the wrong machine, it does not work and further corruption may result.

Note

It is important to create an Emergency Boot Floppy after installing a tape drive. This ensures that you will be able to access the tape drive to back up or restore when using the Emergency Boot Floppy.

The utility used to create the Emergency Boot Floppy is called **mkdev fd**. It is a menu-driven program that creates two types of disks: simple filesystem (described in “Using Floppy Disks and Tape Drives” in the *XENIX System Administrator’s Guide*) and the Emergency Boot Floppy. The formats supported are: 96 tpi-15 sectors/track in the 5-1/4 inch format, and 135 tpi-9 sectors/track in the 3-1/2 inch format.

B

Creating the Floppy or Floppies

To create your floppy, follow this procedure:

1. Log in as *root* and enter:

```
mkdev fd
```

2. You see the following display:

```
Floppy Disk Filesystem Creation Program

      Choices for type of floppy filesystem.

1. 48tpi, double sided, 9 sectors per track
2. 96tpi, double sided, 15 sectors per track
3. 135tpi, double sided, 9 sectors per track
4. 135tpi, double sided, 18 sectors per track

Enter an option or enter q to quit:
```

Enter the number of the disk type desired and press **<Return>**.

3. You see the following prompt:

```
Insert a type floppy into drive 0.
Press <RETURN> to continue or enter q to quit:
```

Press **<Return>**.

4. Next you see:

```
Choices for contents of floppy filesystem.

1. Filesystem only
2. Bootable only
3. Root filesystem only
4. Root and Boot (for 96ds15 and 135ds18 floppies only)

Enter an option or enter q to quit:
```

If you do not have a high density floppy drive on your system, you must create separate root and boot disks; create the bootable diskette first. Enter the appropriate option and press **(Return)**.

5. The following prompt is displayed:

```
Would you like to format the floppy first? (y/n)
```

If you have already formatted the floppy, enter **n** and the filesystem is immediately created. If the floppy has not yet been formatted, enter **y** and you see:

```
formatting /dev/type  
track 00 head 0
```

The track and head numbers count up as the floppy is formatted. (If */etc/default/format* contains **VERIFY=Y**, the format is also verified after formatting.)

6. The following is displayed:

```
Copying files to /dev/type ...
```

If you are creating a boot/root diskette, relevant files are copied from the root filesystem. **mkdev** also checks the filesystem with **fsck(ADM)**; messages similar to the filesystem check displayed at boot time are displayed.

B

7. The following message is displayed when the disk is ready:

```
type floppy created and checked successfully
```

8. You are then returned to the main menu. If you created a boot/root diskette, you can enter **q** and press **(Return)**; your Emergency Boot Floppy is finished. If you are creating separate boot and root diskettes, you should now create the root filesystem diskette. Enter **3** and press **(Return)**.

Creating the Floppy or Floppies

9. You see the following prompt:

```
Insert a type floppy into drive 0.  
Press <RETURN> to continue or enter q to quit:
```

Press <Return>.

10. The following prompt is displayed:

```
Would you like to format the floppy first? (y/n)
```

If you have already formatted the floppy, enter **n** and the filesystem is immediately created. If the floppy has not yet been formatted, enter **y** and you see the formatting messages described earlier.

11. The following messages are displayed:

```
Copying files to /dev/type root filesystem ...  
Copying special files to /dev/type root filesystem ...
```

12. As with the bootable floppy, **mkdev** also checks the filesystem with **fsck(ADM)**; similar messages are displayed.

13. The following message is displayed when the disk is ready:

```
type floppy created and checked successfully
```

Store these diskettes in a safe place. You will need them if your system becomes corrupted and is no longer bootable.

Appendix C

Setting the Time Zone Outside North America

Setting Your Time Zone C-1

Setting Your Time Zone

This appendix describes how to set your time zone if you are not located in North America. Follow these steps:

1. After the initial prompt described in the time zone procedure, you see the following:

```
What is the abbreviation of your standard time zone?  
Enter 1-9 characters or enter q to quit:
```

Enter the abbreviation of your time zone.

2. You see:

```
How many hours west of Greenwich Mean Time are you?  
Enter hh[:mm:ss] (e.g. 10:30:00 or 10:30, use negative  
numbers for locations east of GMT) or enter q to quit:
```

Enter a number.

3. The following prompt is displayed:

```
Does summer time (daylight saving time) apply at your location (y/n)?
```

If summer time does not apply, enter **n** and proceed to step 7.

If summer time does apply, enter **y**. You then see:

```
What is the summer abbreviation of your time zone?  
Enter 1-9 characters or enter q to quit:
```

Enter the summer abbreviation of your time zone.

C

Setting Your Time Zone

4. Next you see this prompt:

1. Week of the year (1-52).
2. Week of a specific month (eg. 1st week of April).
3. Day of the year, ie. Julian date (1-365).

Select the method your time zone uses to convert from standard time to summer time (daylight saving time) or enter q to quit:

Select a method and follow the prompts.

5. You see:

At what time of day is the conversion made (use 24 hour clock)?
Enter hh[:mm:ss] or press RETURN for default value of 2 am
or enter q to quit:

Enter a number or press <Return>.

6. Depending upon which method your time zone uses, you see the following prompt:

How many hours does your timezone adjust for summer time (daylight saving time)?
Enter hh[:mm:ss] or press RETURN for the default value of 1 hour
or enter q to quit:

Enter a number or press <Return>.

7. The time zone variable, */etc/TZ*, is changed accordingly. Your time zone is now set. There is no need to change the *.profile* for users unless they call in from a different time zone and want to override the local standard.

You should now return to Chapter 2 and continue the installation from step 47.

Appendix D

If You Created Additional Filesystems

Preparing Your Filesystems for Use D-1

Preparing Your Filesystems for Use

If you created any additional filesystems (such as */u*) earlier in the installation, you must prepare them for use by following the instructions in this appendix.

1. Before leaving system maintenance mode, you must check the new filesystem with **fsck**(ADM). For example, for a */u* filesystem, enter the following command:

```
fsck /dev/u
```

The **fsck** check phases are displayed as with the root filesystem.

2. Now you are ready to add the new filesystem to the system. The **divvy n[ame]** menu option created the device node for the filesystem (for example: */dev/u*). The next step is to configure the filesystem so that it is recognized and ready for use. Using the filesystem */u* as an example, this is how the actual filesystem is created:

```
mkdev fs /dev/u /u
```

This command does the following:

- Creates a directory */u* (also known as the mount point).
- Creates the *lost+found* directory for **fsck**(ADM). Files are then created in the *lost+found* directory, then removed. This allocates inodes for the directory, so that if the filesystem is corrupted and runs out of inodes, **fsck** is still able to recover files.
- Mounts the device (*/dev/u*) on */u*, and cleans it using **fsck**.
- Removes write permissions on the directory */u* for group and all other users except root.

D

Preparing Your Filesystems for Use

- Adds the following line to */etc/checklist*:

```
/dev/u
```

- Prompts you to decide if the new filesystem is to be automatically mounted, checked, and cleaned at boot time or not.
- Unmounts the filesystem.

Note

Your new filesystem does not have to be called */u*. You can name your filesystems and mount points as you choose. It is recommended that mount points and the corresponding filesystems have the same name to avoid confusion.

3. Next, you must edit the file */etc/default/mkuser*. As distributed, the file has an entry that looks like this:

```
HOME=/usr
```

If you wish to use */u*, change the entry accordingly. Now, whenever you run the **mkuser** command to add a new user, that user account will be in */u*. Make certain that the */u* filesystem is mounted before you run the **mkuser** script, or the new user's directory will not be accessible when */u* is mounted.

To mount or unmount */dev/u* on */u*, use the following two commands respectively:

```
mount /dev/u /u
```

```
umount /dev/u
```

Only the super-user can use the **mount** command. The system administrator can permit users to mount specific filesystems (with or without password protection) by using the **mnt(C)** command. (For more information, refer to "Permitting Users to Mount Filesystems" in the "Maintaining System Security" chapter of the *XENIX System Administrator's Guide*.)

Appendix E

Using divvy with Manual Control

Using the divvy Menu E-1

Using the divvy Menu

The **divvy** program divides a partition into filesystems. The manual adjustment option allows you complete control over the layout of filesystems in the XENIX partition.

Note

If possible, try to limit your filesystems to 60-80 megabytes. System maintenance tools work faster and more efficiently on this size filesystem. Remember that a block is 1024 bytes. To convert blocks to megabytes, simply divide by 1000.

1. You see the main **divvy** menu and a display that shows how your disk is divided similar to the one below:

Name	New File System?	#	First Block	Last Block
root	yes	0	0	25302
swap	no	1	25303	28000
	no	2	-	-
	no	3	-	-
	no	4	-	-
	no	5	-	-
	no	6	-	-
d1057all	no	7	0	28000

27866 blocks for divisions, 244 blocks reserved for the system

n[ame] Name or rename a division.
 c[reate] Create a new filesystem on this division.
 p[revent] Prevent a new filesystem from being created...
 s[tart] Start a division on a different block.
 e[nd] End a division on a different block.
 r[estore] Restore the original division table.

Please enter your choice or 'q' to quit:

E

Each row in the **divvy** table corresponds to a filesystem (also known as a division).

Using the divvy Menu

There are a maximum of 7 divisions on your XENIX partition. Three divisions are reserved: 0 for root, 1 for swap, and 7 refers to the entire partition. Each filesystem you create must have the following:

- if the division does not already exist it must first be created using the **c** (create) command.
- a beginning block number, defined by the **s** (start) command.
- an ending block number, defined by the **e** (end) command.
- a filesystem name, defined by the **n** (name) command. Filesystems can have any name you choose. For example, you could name a filesystem *u* (for “user”). This name is in turn used to create the device name (for example, */dev/u*).

Warning

Note that filesystem boundaries must not overlap. For example, filesystem 0 cannot end on the block number where filesystem 1 begins.

Do not change the configuration of filesystem 7; it is reserved for internal use by the operating system.

Do not name a filesystem *usr*; this directory already exists on the root filesystem.

Exit from **divvy** by entering **q**. The program prompts whether to install the new partition table, return to the main menu, or exit the program without installing the partition table. Select option **i** to install the partition table.

For more information, see the **divvy(ADM)** manual page.

2. The system now creates the filesystems on your hard disk. This takes several minutes.
3. Note that you must run **mkdev fs** to allow the system to actually use any non-root filesystems. You will be instructed at the proper time to do this.

If you are installing XENIX-386, you should now return to Chapter 2 and continue the installation from step 35. If you are installing XENIX-286, you should now return to Chapter 3 and continue the installation from step 28.

Appendix F

Installation Glossary

Terms Used During Installation F-1

Terms Used During Installation

If you are relatively new to XENIX Systems and/or to operating systems in general, the following describes most of the terms used in this guide.

backups	are copies of files from the hard disk that are saved on floppy disks or tapes in case of data loss.
bad track	is a small section of the hard disk that has lost its ability to store data.
bad track table	is a list of known bad tracks on the hard disk. It also contains space for tracks that may go bad in the future. The operating system avoids using these tracks.
badtrk	is a XENIX utility that scans a hard disk and records bad tracks.
BIOS	is Basic Input/Output System. The data tables describe the hardware recognized by the system that is hard-coded into the main circuit board of the computer (motherboard).
bit	is a "0" or a "1". In the binary system, there are no other digits.
block	refers to different quantities depending on the context. For the operating system and the utilities used to divide the hard disk (as discussed in divvy , for instance), a block is 1024 bytes (1 Kbyte); for most utilities and discussion of disk space usage, a block is 512 bytes (as used by the custom utility).
boot	means to begin the startup process. The common expression is "boot the system."
boot floppy	is a floppy disk containing the kernel that "boots" the system from the floppy drive.

Terms Used During Installation

bootstrap	is the program that loads the kernel into memory.
byte	is a group of 8 bits. A byte is the basic unit of information. A single byte can describe a character such as "a" or "5" or "#". For instance, in the ASCII character set (see <code>ascii(M)</code>), the character "A" is represented by a byte containing the bits 0100 0001, which is the number 65 in decimal.
custom	is a program that installs products and optional packages of the operating system.
cylinder	refers to the same track on each of the platters in a hard disk. For example, if a hard disk has five platters, the first cylinder consists of track one on each platter.
device driver	is a program that allows the kernel to communicate with a particular device. Certain drivers, such as the floppy and hard disk driver, are already built into the kernel. Others must be attached or "linked" into the kernel using the link kit.
divvy	is a program that divides a partition into separate filesystems.
dkinit	is a program that enables the XENIX System to recognize a hard disk that is not contained in the ROM BIOS.
driver	See device drivers .
fdisk	is the program that divides a disk into partitions, or different sections of the disk that will be occupied by different operating systems.
filesystem	is a subdivision of a partition that is separate from the area where the operating system and utilities are located. This filesystem can be mounted and unmounted, just like a floppy or other device.
flaw map	is a list of known bad tracks provided by the manufacturer.

Terms Used During Installation

fsck	is the program that checks and cleans a filesystem after a system crash, reconciling various system tables and lists to what is actually on the hard disk.
gigabyte	is one K of megabytes and is abbreviated "G." In other words, 1024 X 1024 X 1024 bytes or 1024 X 1,048,576 bytes or 1,073,741,824 bytes or 2 to the 30th power. This is a huge amount of information. You can create and use files of up to 4 gigabytes, although other factors may impose a lower limit of 1 or 2G. You may also have a gigabyte or more of storage on your hard disk.
hard disk	is actually a series of disks, or platters, that are scanned by a series of heads like those used on a tape recorder.
kernel	is the center of the operating system, a program that is always in memory and underlies all programs, applications, and utilities that run on top of it.
kilobyte	is also known as "K" (for kilo), which means 1000. One K is 1024 bytes. The reason a K is not just "1000" is because 2 to the 10th power is 1024, and in the binary computer world, everything is based on powers of 2 instead of powers of 10.
link kit	is a set of programs used to "link" or connect a device driver to the XENIX kernel. See also device driver .
megabyte	is sometimes abbreviated "meg" or "M" or "Mbyte". One megabyte is one K of Ks. In other words, 1024 X 1024 bytes or 1,048,576 bytes or 2 to the 20th power. This is quite a bit of information, and is often used to measure things such as the size of a hard disk.
mkdev	is a program that creates the device file associated with a device. To the system, a device is simply a file that is written to or read from. Thus, the primary hard disk is associated the file <i>/dev/hd00</i> .

Terms Used During Installation

partition	is a division of a disk that contains an entire operating system. If an entire disk is used for a XENIX System, there is a single partition on the disk.
process	is each program or separate part of a program that is running concurrently on a system.
RAM	is Random Access Memory; it is the hardware memory (or primary storage) as opposed to the hard disk (secondary storage). On systems with virtual memory (like those based on the 80386 processor), the RAM and disk storage are essentially one unit.
reboot	is the act of bringing a system up after shutting it down.
root	is the top-level access account on a XENIX System that has unrestricted access to all files and utilities. This account is used only for system maintenance.
root directory	refers to the top level of the root filesystem.
root filesystem	is the filesystem containing the operating system and its utilities.
run time system	or RTS, is the minimal software necessary on a XENIX System to run properly. Other packages outside the RTS can be loaded using custom .
sector	is a pie-shaped section of a single track of a disk.
shutdown	is the program that shuts down the system in preparation for switching off or rebooting.
single-user mode	See system maintenance mode .
super user	is another name for root.

Terms Used During Installation

swap space	is an area of the hard disk (not a filesystem) that acts as an extension of the hardware memory. Programs that are running but are waiting for input or device access are “swapped out” or moved to the swap space so that other programs can run in memory.
system maintenance mode	is also known as single-user mode. This is a stage of the startup process where certain maintenance functions are done, and other users are not allowed to log in. Like the <i>root</i> or super-user account, this mode has unrestricted access to the system.
tpi	is tracks per inch, a measure of floppy disk storage density.
track	is one of the concentric rings that a hard disk is divided into, similar to a groove on a phonograph record.
utility	is a program that is sometimes considered part of the operating system, but is not part of the kernel. It is an independent user program that runs on top of the system like an application. New utilities can be written by users and are often known as “tools.”

SCO[®] XENIX[®] System V

Operating System

System Administrator's Guide

The Santa Cruz Operation, Inc.

© 1983-1991 The Santa Cruz Operation, Inc.

© 1980-1991 Microsoft Corporation.

© 1989-1991 AT&T.

All Rights Reserved.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, The Santa Cruz Operation, Inc., 400 Encinal, Santa Cruz, California, 95061, U.S.A. Copyright infringement is a serious matter under the United States and foreign Copyright Laws.

The copyrighted software that accompanies this manual is licensed to the End User only for use in strict accordance with the End User License Agreement, which should be read carefully before commencing use of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of The Santa Cruz Operation, Inc.

The following legend applies to all contracts and subcontracts governed by the Rights in Technical Data and Computer Software Clause of the United States Department of Defense Federal Acquisition Regulations Supplement:

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 52.227-7013. The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, California 95061, U.S.A.

SCO and the SCO logo are registered trademarks of the Santa Cruz Operation, Inc.

Microsoft, MS-DOS, and XENIX are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of UNIX Systems Laboratories in the U.S.A. and other countries.

The Santa Cruz Operation is a trademark of the Santa Cruz Operation, Inc.

Hayes is a trademark of Hayes Microcomputer Products, Inc.

Document Version: 2.3.4C

Date: 28 March 1991

Contents

1 Introduction

- Overview 1-1
- The System Administrator and Administrative Roles 1-2
- Summary of Administrator's Tasks 1-4
- Making Administration Easier with the sysadmsh 1-8
- Locating Manual Pages 1-10
- The Super-User Account 1-12
- The Keyboard 1-13
- Running Programs Simultaneously with MultiScreen 1-14
- Educating Users 1-16
- Site Planning Considerations 1-18

2 sysadmsh: Using the System Administration Shell

- Introduction 2-1
- Starting sysadmsh 2-2
- How the Screen is Organized 2-3
- Selecting Menu Items 2-5
- Using Forms 2-7
- Using Scan Windows 2-13
- Getting Help 2-15
- Changing the Current Directory within sysadmsh 2-18
- The Function Keys 2-19
- Using Shell Escapes to Access the XENIX Command Line 2-20
- sysadmsh Environment Variables 2-21
- sysadmsh Menu Options 2-22

3 Starting and Stopping the System

- Introduction 3-1
- Starting the System 3-2
- Logging in as the Super User 3-7
- Stopping the System 3-9
- Understanding the Boot Display Information 3-12
- Changing the Boot Process 3-14

4 Managing Filesystems

- Introduction 4-1
- What Is a Filesystem? 4-2
- Mounting and Unmounting a Filesystem 4-3
- Permitting Users to Mount Filesystems 4-4

- Maintaining Free Space in Filesystems 4-6
- Maintaining Efficient Filesystem Organization 4-17
- Adding Disk Space and Filesystems 4-20
- Filesystem Integrity 4-21

5 Maintaining System Security

- Introduction 5-1
- General Security Practices 5-3
- User Passwords 5-5
- Maintaining File Permissions 5-7
- Managing File Ownership 5-14
- Forcing a New Password 5-16
- Logging Out Idle Users 5-19
- Adding Dial-in Password Protection 5-20
- Permitting Users to Mount Filesystems 5-22
- Monitoring Login Sessions 5-24
- Using XENIX Accounting Features 5-26
- Setting up a Restricted Shell 5-30

6 Backing Up Filesystems

- Introduction 6-1
- Strategies for Backups Using sysadmsh 6-2
- Preparations for Scheduled Backups 6-4
- Performing a Scheduled Backup 6-11
- Performing an Unscheduled Backup 6-15
- Getting a Backup Listing 6-18
- Restoring Individual Files or Directories From Backups 6-20
- Restoring an Entire Filesystem 6-24
- Editing /etc/default/filesys and /etc/default/archive 6-29
- An Explanation of Backup Levels 6-31

7 Adding Hard Disks

- Introduction 7-1
- Before You Start 7-3
- Installing the Hard Disk 7-6
- Adding the New Filesystem(s) 7-18

8 Tuning System Performance

- Introduction 8-1
- Reallocating Kernel Resources with configure 8-3
- Reconfiguring Because of Persistent Error Messages 8-6
- Reconfiguring for Performance 8-7
- Defining Efficient System Usage Patterns 8-12
- Using vmstat to Diagnose System Inefficiency 8-14
- Summary of Tunable Parameters 8-17

9 Using DOS and OS/2

- Introduction 9-1
- OS/2 Coexistence 9-2
- Partitioning the Hard Disk Using fdisk 9-3
- Installing a XENIX Partition on a DOS System 9-8
- Using a XENIX System and DOS with Two Hard Disks 9-9
- Removing an Operating System from the Hard Disk 9-10
- DOS Accessing Utilities 9-11
- XENIX Systems and DOS On Non-Standard Disks 9-14

10 Administering User Accounts

- Introduction 10-1
- Adding a User Account 10-2
- Creating a Group 10-7
- Changing a User's Login Group 10-9
- Changing a User ID 10-11
- Removing a User Account 10-13

11 Building a Remote Network with UUCP

- Introduction 11-1
- Configuring UUCP on Your System 11-10
- Detailed Descriptions of UUCP Configuration Files 11-15
- Special UUCP Configuration Options 11-39
- Administering Your UUCP System 11-44
- Complete UUCP Examples 11-49
- UUCP Error Messages 11-55

12 Building a Local Network with Micnet

- Introduction 12-1
- Planning a Network 12-2
- Building a Network 12-9
- Starting the Network 12-16
- Testing a Micnet Network 12-17
- Using a UUCP System 12-21

13 XENIX Directories and Special Device Files

- Introduction 13-1
- XENIX Directories 13-2
- Log Files 13-7
- Special Device Files 13-9

14 Adding Terminals and Modems

- Introduction 14-1
- Adding and Using Terminals 14-2
- Installing and Using Modems 14-24

15 Using Printers

- Introduction 15-1
- Installing a Printer 15-3
- Adding a Local Printer 15-12
- Print Service Command Summary 15-16
- Starting and Stopping the Print Service 15-18
- Managing the Print Service 15-20
- Managing the Printing Load 15-25
- Using Interface Programs 15-29
- Setting Up RTS/CTS Protocol Serial Printers 15-36
- Using a Printer Without the Spooler 15-39
- Creating an Init Device File 15-40

16 Using Floppy Disks and Tape Drives

- Introduction 16-1
- Using Cartridge Tape Drives 16-2
- Using Floppy Disks 16-15

17 Adding Multiport Cards, Memory, and Other Bus Cards

- Introduction 17-1
- Installing Bus Cards 17-2
- Adding More Memory 17-5
- Adding and Configuring Serial Ports 17-7

18 Using a Mouse

- Introduction 18-1
- Configuring the Hardware 18-2
- Installing a Mouse 18-3
- Using the Mouse 18-9

19 Troubleshooting Your System

- Introduction 19-1
- Solving Startup (Boot) Problems 19-2
- Resolving Console Problems 19-11
- Fixing Filesystem Problems 19-15
- Troubleshooting Installation Problems 19-26

Troubleshooting Your Modem 19-28
Troubleshooting the Print System 19-35
Resolving Process Errors 19-42
Fixing Problems with Schedulers 19-45
Recovering from Other System Failures 19-47
Troubleshooting Your Tape Drive 19-54
Solving Terminal Problems 19-58
Troubleshooting Network Connections 19-62

A Customizing System Startup

Introduction A-1
Changing Scripts in /etc/rc.d A-2
Modifying .profile and .login Files A-5
Changing the /etc/motd File A-6

B Using the Job Scheduling Commands: at, cron and batch

Introduction B-1
Automatic Program Execution with cron B-2
Delaying Program Execution with batch and at B-5
Permitting Users Access to Job Scheduling B-9
Managing cron Jobs with the queuedefs File B-10

C Using the System Console and Color Displays

Introduction C-1
Console Keyboard Type Selection C-2
Using the Console Screen Protection Feature C-4
Using Multiscreen C-5
Changing Video Fonts C-8
Switching Video Adapters C-9
Controlling Color Displays with setcolor C-10

Chapter 1

Introduction

Overview 1-1

The System Administrator and Administrative Roles 1-2

Summary of Administrator's Tasks 1-4

Making Administration Easier with the sysadmsh 1-8

Locating Manual Pages 1-10

The Super-User Account 1-12

The Keyboard 1-13

Running Programs Simultaneously with MultiScreen 1-14

Educating Users 1-16

Site Planning Considerations 1-18

Overview

The XENIX system is designed to accommodate a wide variety of needs, including the use of peripheral devices, including terminals, printers, and tape drives. Your 1 system also requires careful control of its operation and a regular schedule of maintenance to keep system running smoothly. This guide explains how to run and maintain the XENIX Operating System on your computer, ensuring maximum performance with the fewest problems.

The tasks presented in this guide range from simple ones requiring very little knowledge about XENIX systems, to complex tasks requiring extensive knowledge about the operating system and your computer. Each chapter explains the tools and knowledge you need to complete the tasks described in that chapter. In some cases, you may be referred to other manuals.

This guide contains chapters about computer hardware you may wish to use with your system. The use and interaction of various devices with the operating system is described in a comprehensive fashion. For example, “Using Floppy Disks and Tape Drives” discusses the use of magnetic storage media, and it covers the basics of preparing the operating system for such a device, installing it, and how to use the drive once it is installed.

This guide also explains how to expand your system with remote and local networks. For local networking over serial lines, **micnet** can link XENIX systems in your work environment. For remote communications over phone lines, UUCP can be set up to communicate with XENIX and UNIX system sites all over the world. (See “Building a Serial Network with Micnet” and “Building a Remote Network with UUCP” in this guide for a complete explanation of network facilities available.)

An important part of system operation is the protection of data on the system. Security is discussed in great detail in “Maintaining System Security.”

Pay special attention to “Troubleshooting Your System.” This chapter is an excellent resource to help you keep your system running smoothly. Refer to this chapter whenever you run into difficulties.

The System Administrator and Administrative Roles

Every XENIX system should have at least one person in charge of system maintenance and operation. In this guide, such a person is called a system administrator. It is the responsibility of system administrators to ensure the smooth operation of the system. These duties require that the system administrator(s) become proficient with a wide variety of functions.

Depending on the size of the system and the number of users on it, system administration can be anything from a once-a-day task to a full-time job. Even if the system is small, the system administrator should faithfully perform each required maintenance task, because sloppy maintenance can adversely affect system performance.

The system administrator should keep a log of all system modifications and system events. Each event, message, backup, or modification should be logged with the date, time, and name of the person logging, and the circumstances surrounding the event. For example, if a new application is added to the system software, an entry should be placed in the log. This entry should include the time, date, and name of the person installing, and any notes about the software or installation that may be helpful. An accurate log helps in diagnosing system problems and charting the growth and use of a system.

All tasks in this guide are presented from a system administrator's point of view, but many can also be accomplished by ordinary users. Because some of the tasks dramatically change the system's operation, we recommend that, whenever possible, the system administrator perform these tasks. However, no matter who performs an operation, it should be entered in the system log. Following these rules can prevent unwanted or unnecessary changes to the system.

The System Administrator and Administrative Roles

A system administrator has several tasks to perform, sometimes on a daily basis:

- Make certain the integrity of the system is not compromised through use of security mechanisms.
- Make certain that adequate backups (regular copies of files on the system) are made and stored for future use.
- Handle problems related to use of limited computer resources (disk space, number of processes, and so on).
- Alleviate system communication (network) stoppages due to failed connections.
- Apply operating system updates and maintenance fixes.
- Provide general support to users.

1

Summary of Administrator's Tasks

A system administrator has numerous tasks to perform. They can be divided into groups according to how often they are carried out. The following list ranges from tasks that must be performed more often than once a day to those that need be performed less often than once a month. The administrator may have to do some of the tasks listed in Table 1.1 more or less often, depending upon the size and complexity of the system. For more information on each task, see the reference indicated.

Table 1.1
Task List

As Needed Tasks	For More Information:
<input type="checkbox"/> Record all system modifications and events in log.	Preceding paragraphs in this section
<input type="checkbox"/> Be on call for panics, crashes, power spikes, user questions.	Chapter 19, "Troubleshooting Your System"
<input type="checkbox"/> Maintain security of hardware, software, data file access.	Chapter 5, "Maintaining System Security" and chapters on using each of the hardware components

Daily Tasks	For More Information:
<input type="checkbox"/> Perform backups.	Chapter 6, "Backing up Filesystems"
<input type="checkbox"/> Check usage levels.	"Using Performance Tools to Diagnose System Inefficiency" in Chapter 8, "Tuning System Performance"
<input type="checkbox"/> Check for runaway processes.	"Runaway Processes" in Chapter 19, "Troubleshooting Your System."

(Continued on next page.)

Task List (Continued)

Daily Tasks	For More Information:
<input type="checkbox"/> Check disk space.	"Displaying Free Space" in Chapter 4, "Managing Filesystems"
<input type="checkbox"/> Check printer status.	"Displaying Printer Status" in Chapter 15, "Using Printers"
<input type="checkbox"/> Check communications links, if active.	Chapter 11, "Building a Remote Network with UUCP" and Chapter 12, "Building a Serial Network with Micnet"
<input type="checkbox"/> Check for unattended login sessions.	Man page for who(C) command in the <i>XENIX Reference</i> .
<input type="checkbox"/> Remove <i>core</i> and <i>*.out</i> files.	"Maintaining Free Space in Filesystems" in Chapter 4, "Managing Filesystems"

Weekly Tasks	For More Information:
<input type="checkbox"/> Run fsck(ADM) on all filesystems.	"Filesystem Integrity" in Chapter 4, "Managing Filesystems"
<input type="checkbox"/> Check printer spooler status report.	Check lp(C) account mailbox for messages
<input type="checkbox"/> Check log files such as <i>/etc/wtmp</i> and those in <i>/usr/adm</i> and <i>/usr/spool</i> and clear, trim, or truncate.	"Maintaining Free Space in Filesystems" in Chapter 4, "Managing Filesystems"
<input type="checkbox"/> Use vmstat(ADM) to generate a report of activity.	"Using Performance Tools to Diagnose System Inefficiency" in Chapter 8, "Tuning System Performance"

(Continued on next page.)

Summary of Administrator's Tasks

Task List (Continued)

Weekly Tasks	For More Information:
<input type="checkbox"/> Generate detailed report of user disk utilization.	"Displaying Disk Usage" in Chapter 4, "Managing Filesystems"
<input type="checkbox"/> Remove temporary files, and <i>lost+found</i> files.	"Maintaining Free Space in Filesystems" in Chapter 4, "Managing Filesystems"

Monthly Tasks	For More Information:
<input type="checkbox"/> Perform full system backup.	Chapter 6, "Backing up Filesystems"
<input type="checkbox"/> Archive critical files if changed.	"Archiving Files on Tape" and "Using Floppies for File Storage" in Chapter 16, "Using Floppy Disks and Tape Drives"
<input type="checkbox"/> Re-tune system and re-allocate resources, if needed.	Chapter 1, "Introduction," and the "Reallocating Kernel Resources" section in Chapter 8, "Tuning System Performance"
<input type="checkbox"/> Perform hardware maintenance.	Chapter 16, "Using Floppy Disks and Tape Drives" and hardware documentation
<input type="checkbox"/> Change dial-in passwords, if needed.	"Adding Passwords for Dial-in Lines" in Chapter 5, "Maintaining System Security"
<input type="checkbox"/> Change root password, if needed.	"Changing a User Password" in Chapter 10, "Administering User Accounts"

(Continued on next page.)

Task List (Continued)

Occasional Tasks	For More Information:
□ Upgrade OS and Application software, as needed.	Installation and/or Release Notes of the new version
□ Fix permissions on software.	fixperm (ADM) man page in <i>XENIX Reference</i>
□ Re-distribute space in filesystems.	"Restoring Free Space" in Chapter 19, "Troubleshooting Your System" or "Maintaining Free Space" in Chapter 4, "Managing Filesystems"
□ Find SUID or SGID files, check owner, size.	"Locating Files" in Chapter 4, "Managing Filesystems" and find (C) man page in the <i>XENIX Reference</i>
□ Locate huge files and verify their purpose.	"Locating Files" in Chapter 4, "Managing Filesystems"
□ Find "orphan" files (no real user).	"Locating Files" in Chapter 4, "Managing Filesystems"
□ Locate sparse directories and compress, if needed.	"Maintaining Efficient Filesystem Organization" of Chapter 4, "Managing Filesystems"

Making Administration Easier with the sysadmsh

The **sysadmsh**(ADM) command is a menu interface designed to simplify the task of system administration. The menus, submenus, and screens allow you to simply point and pick, or fill in blanks. The **sysadmsh** allows less-experienced system administrators to use XENIX commands that would otherwise require memorization and constant reference to manual pages. The **sysadmsh** includes context-sensitive help; simply press the <F1> key from any menu to display further explanations of the menu options.

If you are new to XENIX operating systems, we strongly recommend that you become familiar with the concepts and tasks covered in the *XENIX Tutorial*. This guide assumes some familiarity with XENIX systems; after studying the *XENIX Tutorial*, you should be able to perform the basic system administrative tasks described here.

To aid users of **sysadmsh**, the documentation of this guide is supplemented by **sysadmsh** references that appear below XENIX command-line instructions.

For example, the following instructions refer to the **custom** utility, used to add more software to your system. Below the actual command is a sequence of **sysadmsh** menu selections.

Enter the following command:

custom

△ **sysadmsh** users select: System→Add→Software

This means that you can access the functions of the **custom** command by first selecting System at the main **sysadmsh** menu, followed by selecting Add, and then Software at the next lower level. Selections can be made from the menu in any of the following ways:

- Move through the menu options using the <Space> key and press <Return> on the option you want.
- Move left and right through the options using the arrow keys and press <Return> on the desired option.

Making Administration Easier with the sysadmsh

- Press the first letter of the option desired. This is the quickest way. Using the example above, you would simply enter sas (without the <Return> key) to reach the **custom** menu.

For more instructions on using the **sysadmsh**, refer to the “sysadmsh: Using the System Administration Shell” chapter in this guide.

1

Locating Manual Pages

When you use the command line rather than the `sysadmsh` menu interface, you have direct access to utilities and data. Notice the form used for commands in this guide. Each command is printed in bold type, and each has a suffix to help you find more information about it.

Table 1.2 lists the locations of the manual pages for the commands with the indicated suffixes. To find information about a command, note the letter or letters that appear in parentheses following the command, then look up the command in the appropriate reference book or guide. For example, the command `lpstat(C)` is defined in the Commands (C) section of the *XENIX Reference*.

Table 1.2
Manual Page Locations

Command Suffix	Book and Purpose
ADM	<i>XENIX Reference</i> - commands reserved for the exclusive use of system administrators
C	<i>XENIX Reference</i> - operating-system commands available to all users
CP	<i>Programmer's Reference</i> - programming commands used with the development system
CT	<i>Text Processing Reference</i> - commands used with the text processing system
DOS	<i>Programmer's Reference</i> - DOS routines used with the development system
F	<i>XENIX Reference</i> - (File Formats) description of system files not found in the Miscellaneous (M) section
HW	<i>XENIX Reference</i> - information about hardware devices and device nodes

(Continued on next page.)

Manual Page Locations (*Continued*)

Command Suffix	Book and Purpose
M	<i>XENIX Reference</i> - miscellaneous information used for access to devices, system maintenance, and communications
S	<i>Programmer's Reference</i> - system calls and library routines for C and assembly-language programming



The Super-User Account

The super-user login (also known as *root*) is a special account for performing system maintenance tasks. It gives the system administrator unusual privileges that ordinary users do not have, such as access to all files in the system, and executing privileged commands. Many of the tasks presented in this guide require that the system administrator be logged in as the super user. To do this, the system administrator must know the super-user password created during the installation of your system. (See the *Installation Guide*.)

Log in as the super user only to perform system-maintenance tasks. Even if the system administrator is the only one using the system, that person should create a user account for day-to-day work, reserving the super-user account for system-maintenance tasks only.

Few users should know the super-user password. Misuse of the super-user powers by naive users can result in a loss of data, programs, and even the operating system itself.

The Keyboard

Many keys and key combinations perform special actions on XENIX systems. These have names that may not correspond to the keytop labels on your keyboard. Table 1.3 shows which keys on a typical terminal correspond to special actions on XENIX systems. A list for your particular login device is in **keyboard(HW)**. Many of these keys can be modified by the user; see **stty(C)**.

Table 1.3
Special Keys

XENIX Name	Action
<code><Return></code>	terminates a command line and initiates an action. Also called the ENTER key; keytop may indicate a down-left arrow.
<code><Esc></code>	exits the current mode; for example, exits insert mode when in the editor vi . Also known as the ESCAPE key.
<code></code>	stops the current program, returning to the shell prompt. This key is also known as the INTERRUPT or the DELETE key.
<code><Bksp></code>	deletes the character to the left of the cursor. May be indicated by a large left arrow, as opposed to the small "cursor left" arrow.
<code><Ctrl>d</code>	signals the end of input from the keyboard; exits current shell, or logs you out if the current shell is the login shell. Not interchangeable with the BREAK key.
<code><Ctrl>h</code>	deletes the first character to the left of the cursor. Also called the ERASE key.
<code><Ctrl>q</code>	restarts printing after it is stopped with <code><Ctrl>s</code> .
<code><Ctrl>s</code>	stops printing at the standard output device, such as a terminal. Does not stop the program.
<code><Ctrl>u</code>	deletes all characters on the current line. Also called the KILL key.
<code><Ctrl>\</code>	quits current command, creates a <i>core</i> file. (Recommended for debugging only.) See core(F) .

Running Programs Simultaneously with MultiScreen

With MultiScreen,TM you can run several programs on your console at the same time. You see the display for each program on a different "screen," but you never have to leave your single console. Pressing a simple key combination switches you from one screen to another, and each screen acts independently from the others.

When you log in normally, you see the screen associated with the <Alt><F1> key combination. To open a second screen, press and hold the <Alt> key, then press <F2> or another function key on your keyboard. Function keys are generally located across the top or down the far left side of your keyboard.

After you press the key combination, the screen clears and a login prompt appears. Log in again and you can begin work on the second "screen." Press <Alt><F1> to switch instantly back to the first screen. Switching between screens in this way can speed up procedures that require working in two different programs. You can run both programs simultaneously, and work on either one at any time. See Figure 1-1.

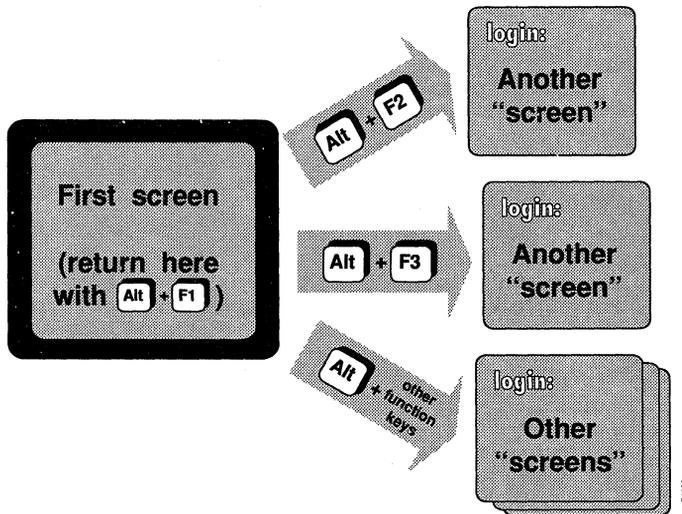


Figure 1-1 MultiScreen Example

Running Programs Simultaneously with MultiScreen

For example, you can start **sysadmsh** on your first screen, then press `<Alt><F2>` to create a second screen. Login again and begin editing a log file with **vi**. Use `<Alt><F1>` and `<Alt><F2>` to switch back and forth between screens, taking actions in **sysadmsh** and recording them in the log.

1

You can open more than two multiscreens at once. However, make sure to log out of all the extra screens before logging out of your first screen. By following this procedure you avoid unauthorized access from leaving login sessions unattended.

If you have several screens open at once, you can rotate through them by pressing the Control and Print Screen key combination, `<Ctrl><PrtSc>`. This is helpful if you cannot remember which process is running on which screen.

Note that you can configure the system to use `<Ctrl><Alt>` *function-key* combinations in addition to `<Alt>` *function-key* combinations to change multiscreens. This is especially useful in applications that reserve the `<Alt>` *function-key* combinations for their own use. This can be configured using the **mapkey(M)** utility.

For more information, refer to **multiscreen(M)** and **screen(HW)**.

Educating Users

The following list contains items the system administrator can explain to users so they can take advantage of the system's resources without overloading them or causing unnecessary system problems. The better users understand the system and its limits, the less demands are placed on the system administrator.

If a user is completely new to XENIX systems, the system administrator should recommend a training course or at least a careful review of the *XENIX Tutorial*. Augment this training with the items listed here.

What Every User Should Know:

- How to log in - Describe how to generate a user password. Emphasize password secrecy, and methods of memorization. Explain limits on password length and type, schedule for changing passwords, rules about changing passwords. Tell users how to specify their terminal type, and help them preset the terminal type in their *.login* or *.profile* files if they use the same type of terminal regularly. See Chapter 10, "Administering User Accounts."
- How to manage files and directories - Make sure user is familiar with basic commands; refer user to the *XENIX Tutorial* if necessary.
- How to edit files - Make sure user knows basic **vi** commands, or the basic commands of the default editor; refer user to the *XENIX Tutorial* or training class if necessary.
- How to invoke programs - Relate names, locations, and invoking commands for the most-used programs.
- How to print files - Explain appropriate print commands for output of most-used programs. Tell user location of default print service printer. Demonstrate how to replenish paper and toner, tape, or ribbon cartridge. Refer user to printer documentation if necessary.
- Good mail etiquette - Explain how to read and send mail, and how to how to organize messages by appending them to mailbox files. Show how these files can be read with the **mail -f** command. Explain how to clean out unwanted messages by deleting or saving to floppy. Emphasize importance of maintaining free disk space.
- Good filesystem planning - Explain limits of directory size. Login and working directories should have less than 30 entries, data storage directories should have less than 286 entries. Warn users that directories do not get smaller, even if entries are removed.

Discourage users from saving mail messages in separate files rather than appending them to existing mailbox files. See Chapter 4, “Managing Filesystems”

- How to store (archive) files - Show users how to use **tar(C)** or **cpio(C)** to archive unused files or directories to tape or floppy disk. See Chapter 16, “Using Floppy Disks and Tape Drives.”
- How to reset a scrambled terminal - Teach users how to escape from most-used programs and how to reset a terminal with **tset** and **stty sane**. Tell users how to power cycle (turn on and off) their terminal if necessary. Make sure users try these procedures before asking a system administrator to disable and re-enable the scrambled terminal. See Chapter 19, “Troubleshooting Your System.”
- How to kill a hung process - Instruct users how to use **ps -flu** on a neighbor’s terminal to find the hung process on their own terminal, and how to kill the process without causing undue system problems. See Chapter 19, “Troubleshooting Your System.”

Site Planning Considerations

Before installing or expanding your system, review the following list of considerations with management and create a physical plan of the system that makes the most efficient use of available resources, and allows users the most direct and complete access to those resources.

If growth is expected, plan for it. Make sure the resources and the communications lines that connect them are adequate for expected growth.

- Set up the computers in a place where they will not be bumped or moved at any time. If possible, they should be in a room by themselves, with little or no foot traffic. If workstations are used for data storage, at least place them on stable furniture and leave no cables exposed to traffic.
- Keep the computer room cool and give each machine excellent ventilation; keep all machines away from walls and, if possible, provide a separate air conditioner for the computer room, with more-than-adequate cooling capability.
- Install a Halon fire extinguishing system in the computer room rather than sprinklers.
- Store backup media in a separate room from the computers. This room should be fireproof, or should at least have a Halon fire extinguishing system rather than a sprinkler system.
- Assure adequate and uninterrupted power for the computers. This means more than enough amperage, and at least surge suppressors, if not a means of guaranteeing uninterrupted power supply. If power fluctuations and outages are common, provide backup power supply. The computers should also be on an isolated, fully grounded circuit.
- If you install a Local Area Network, plan the cabling and location of all machines and peripherals carefully. Seek the assistance of a networking expert to make these plans. Good planning and the use of adequate connecting media and compatible hardware are essential for long-term network performance.
- If you need modem lines for off-site connections, arrange for these with your local telephone company.

Site Planning Considerations

- If you plan to connect a printer to a parallel port, locate it close to the machine running it, but keep the machine out of the path of traffic to and from the printer.
- If you connect terminals, printers, or other peripherals to serial ports, consider using phone-line cabling and switching hardware, especially if your system is expected to grow in size and complexity. You can readily adapt phone lines for serial hardware, and telephone connecting and switching technology is mature and flexible.

1

Chapter 2

sysadmsh: Using the System Administration Shell

Introduction 2-1

Starting sysadmsh 2-2

How the Screen is Organized 2-3

Selecting Menu Items 2-5

Using Forms 2-7

Using Scan Windows 2-13

Getting Help 2-15

Changing the Current Directory within sysadmsh 2-18

The Function Keys 2-19

Using Shell Escapes to Access the XENIX Command Line 2-20

sysadmsh Environment Variables 2-21

sysadmsh Menu Options 2-22

Introduction

The **sysadmsh** (system **admin**istration **shell**) is a menu interface designed to simplify the task of system administration. The **sysadmsh** allows you to run the scores of system administration commands with their numerous options without having to use the traditional XENIX command line.

2

This chapter explains how to use the **sysadmsh** interface. To use **sysadmsh** effectively, you also need to know something about the XENIX commands called by **sysadmsh**. A list of these commands and their corresponding **sysadmsh** menu options is given at the end of this chapter. This list refers you to other sections of the XENIX documentation that contain detailed information on each command. You will find it easier to learn the material in this chapter if you start the **sysadmsh** and actually run the examples as you get to them.

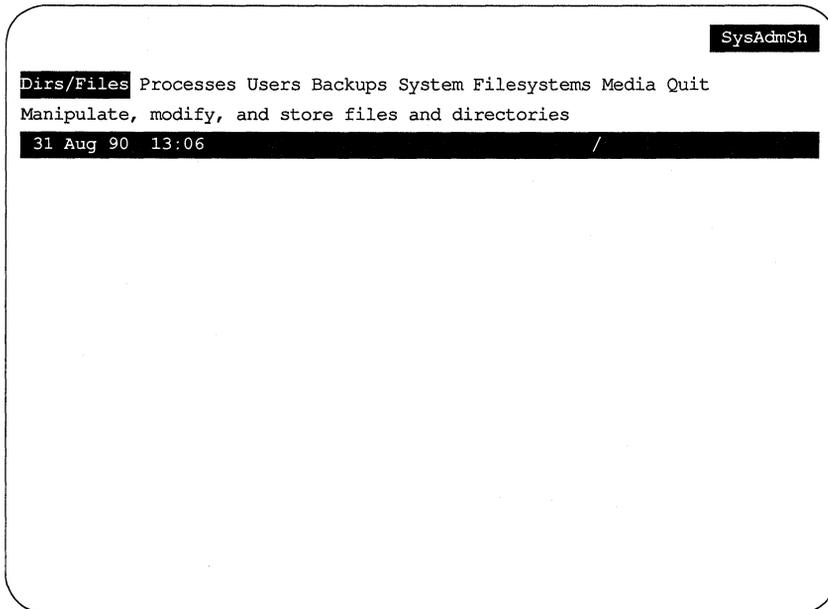
You should become familiar with the concepts covered in the *XENIX Tutorial* before using the **sysadmsh** menus.

Starting sysadmsh

For the purposes of this tutorial, log in as *root* and enter the following command:

sysadmsh

The main **sysadmsh** menu is displayed:

A screenshot of the sysadmsh menu interface. The window has a title bar in the top right corner that says "SysAdmSh". The main content area shows a menu with the following items: "Dirs/Files", "Processes", "Users", "Backups", "System", "Filesystems", "Media", and "Quit". Below the menu items is a line of text: "Manipulate, modify, and store files and directories". At the bottom of the window, there is a status bar that displays "31 Aug 90 13:06" followed by a slash and a space.

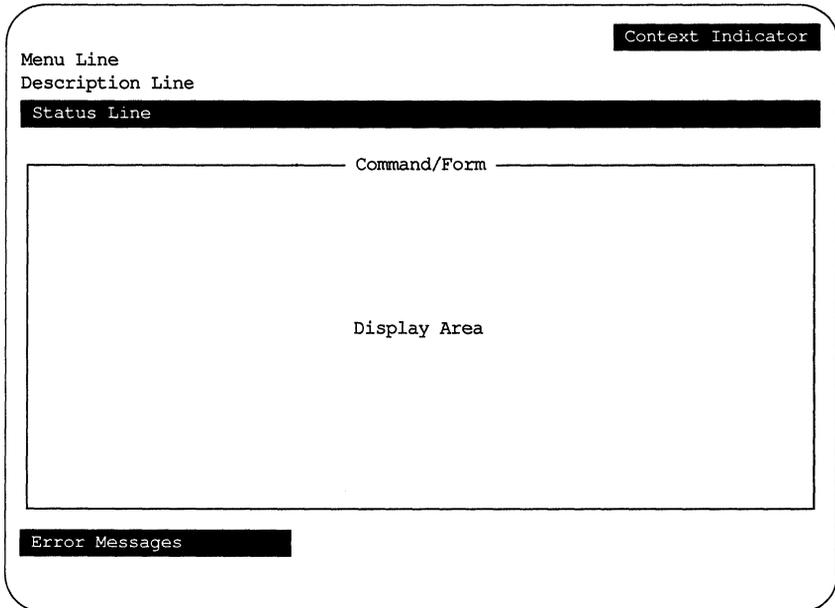
```

SysAdmSh
Dirs/Files Processes Users Backups System Filesystems Media Quit
Manipulate, modify, and store files and directories
31 Aug 90 13:06 /
```

How the Screen is Organized

This is a schematic of the **sysadmsh** screen. Areas shown in black appear on the screen as highlighted areas or bars of text. Each area displays specific types of information:

2



- The Context Indicator is the highlighted bar of text in the upper-right corner of your screen. It displays the name of the current menu. The Context Indicator for the **sysadmsh** opening screen shows *SysAdmSh*.
- The Menu Line displays the menu options that are currently available. The main **sysadmsh** menu consists of eight options: Dirs/Files, Processes, Users, Backups, System, Filesystems, Media, and Quit.
- The Description Line gives you a brief description of the currently highlighted menu option.

How the Screen is Organized

- The Status Line is the highlighted bar of text that separates the Menu and Description Lines from the Display Window. The Status Line in the **sysadmsh** opening screen contains the date, time, and current working directory. When a XENIX command is executed, the name of the command and the options used are displayed briefly at the far left of the Status Line.
- the Display Area. The title can be either a XENIX command name or the name of a **sysadmsh** form. When a command name is displayed, the location of the manual page associated with the command is appended in parentheses. For example, when System→Report→Users is selected, the Command Line displays “who(C).” This means that the command can be found on the **who** manual page in the *XENIX Reference*.
- The Display Area displays **sysadmsh** forms and scan windows. Forms and scan windows are explained in detail later in this chapter.
- Error Messages and recovery instructions appear on the last line of the screen in highlighted text.

Selecting Menu Items

The keystrokes listed in Table 2.1 are used to traverse the menus. Note that there are several ways to select options; if you have used menu-based programs before, use the method you are most familiar with.

Table 2.1
Basic Menu Keystrokes

Keystroke	Action
Arrow keys or <Space> (same as right arrow)	moves to menu option.
First letter of option, or move highlight to option and press <Return>	selects menu option.
<Esc>	retreats to previous menu.
<F1>	gets help.

You can familiarize yourself with the menu options by using the Arrow keys or <Space> to move the highlight from option to option. Each time you move the highlight to a new option, a description of that option appears on the description line.

sysadmsh has a hierarchical menu structure. Many of the menu options move you down to another menu. For example, when you select the Processes option from the main menu, a submenu containing more options is displayed which lets you check on and manipulate your machine's processes. The menu hierarchy makes it easy to find the command you need by moving down from one menu to the next. Eventually you get to a menu option that either executes a XENIX command or displays a form that you must fill in with the details that the command needs. Note that typing the first letter of the option name is the quickest way to move through menu levels; in time you will be able to reach the function you need instantly by pressing three- and four-letter codes you have memorized. (Table 2.9 at the end of this chapter lists all available **sysadmsh** selections and their shorthand forms.)

The best way to learn how to use menus is to practice making menu selections with these keystrokes. If you select an option by mistake, you can always retreat to the previous menu by pressing the <Esc> key. If you are

Selecting Menu Items

several levels deep, you can return to the main menu by pressing the `<F2>` key and then typing `n`. `<F2>` takes you to the Quit option, and `n` returns you to the main menu. (The `<F2>` key does not work if you have been dropped into a XENIX command, such as `vi(C)`.) To help you find your way through the `sysadmsh` menus, Table 2.2 contains a map of the second-level menus.

Table 2.2
Map of Second-Level Menus

Menu Map						
Dirs/Files	Processes	Users	Backups	System	Filesystems	Media
↓	↓	↓	↓	↓	↓	↓
List	Report	Report	Create	Report	Check	List
View	Terminate	Add	Restore	Halt	Add	Extract
Copy		Delete	Schedule	Execute	Mount	Archive
Edit		Modify		Configure	Umount	Format
Modify		Communicate		Add		Duplicate
Print				Delete		Tapedump
Archive						
Differences						
Remove						
UseDOS						

This chapter uses a syntax convention for denoting a string of menu options. For example, to print a file you must select the Dirs/Files option from the main menu, and then select the Print option from the Dirs/Files menu. This sequence is denoted by the shorthand notation `Dirs/Files→Print`, and it can be executed by typing `dp`.

When you select a menu option, one of three things happens:

- A lower-level menu is displayed.
- You are dropped into a form.
- A XENIX command is executed and the result displayed in a scan window.

The next two sections explain forms and scan windows.

Using Forms

Some menu options require additional information to perform the correct task. For example, the Print option cannot do anything until you tell it what you want to print and which printer to use. When you select this type of option, a form appears on the screen. By filling in the form, you give the command the information it needs.

2

The following example demonstrates how forms work by showing you how to print a file in your current directory. After the example, Tables 2.3, 2.4, and 2.5 list the keystrokes that allow you to move around the form, edit it, and make “point-and-pick” selections.

To print a file, first select Dirs/Files→Print. The Print form is displayed:

The screenshot shows a terminal window with a rounded border. At the top, there is a text input field containing "Enter file or directory name" and a "Print" button on the right. Below this is a status bar showing "31 Aug 90 13:06" and a "/" symbol. The main area of the window is titled "Print Files" and contains two prompts: "Enter file(s) to print: [?]" and "Enter destination printer: [?]", each followed by a closing bracket "]" on the right side.

Using Forms

Notice that the highlight is on the first item in the form. You can fill in the field or obtain a list of choices by pressing <F3>. You can enter the filename if you know it but, for the sake of this tutorial, assume you need to find the filename and press <F3> now. A window opens up overlapping part of the Print form:

```
Enter file or directory name Print
/*
31 Aug 90 13:06 /

----- Print Files -----

Enter file(s) to print: [? ]
Enter destination printer: [? ]

file1      file2      file3      file4
file5      file6
```

The window contains a list of files that you can select. To select a file, “point” to it by highlighting it, and “pick” it by pressing return. This is known as point-and-pick, and it is used whenever a range of choices is displayed. After making your selection, the window closes and you return to the Print form.

Note that the name of the file you selected is now displayed in the form. You can now change the name using the edit keys (listed in Table 2.4 later in this section), or press <Return> to move to the next field.

Now enter the name of the printer to be used. If you do not know the printer name, press `<F3>` and another, smaller window opens that contains a list of installed printers:

The screenshot shows a terminal window with the following content:

```

Enter printer name or type ?<Return> for a list of printers Print
-----
31 Aug 90 13:06 /
-----
Print Files
-----
Enter file(s) to print:  [./file1 ]
Enter destination printer:  [? ]
-----
...
printer1
printer2
printer3
  
```

2

You can select the printer just as you did the name of the file. After selecting a printer, you return to the Print menu.

The keystrokes listed in the following tables allow you to use forms easily.

Table 2.3
Form Keys

Keystroke	Action
⟨Esc⟩	tells the program that you changed your mind and do not want to finish filling in this form. The form is removed, and no action is performed. You are returned to the previous menu. In addition, ⟨Esc⟩ followed by ⟨Return⟩ acknowledges that an error message was read and that you are ready to continue.
Up, Down Arrow	moves to other fields in a form. Some fields are restricted and no input is allowed. The Arrow keys skip over these. Other fields must be filled in. Pressing the Down Arrow key on the last item in a form brings you back to the first item.
Left, Right Arrow	moves left and right in the current field. Allows changing of text without retyping the entire line.
⟨Return⟩	completes the data entry to a field and moves the cursor to the next field. In the last field, ⟨Return⟩ completes the entire form and tells the shell that the data is ready to use.
⟨Ctrl⟩x	exits and executes the form from wherever you are. Think “x” for “execute.” ⟨F10⟩ does the same.
⟨F4⟩	calls the spelling checker utility when you are in a form. If you think a word might be misspelled, press ⟨F4⟩ while the cursor is on the word and a list of possible correct spellings appears in a point-and-pick list. The word you select replaces the misspelled word.

Table 2.4
Edit Keys

Keystroke	Action
<Ctrl>y	deletes the current line, and starts over.
<Ctrl>w	deletes the current word.
<Ctrl>g-<Ctrl> h	moves the cursor to the beginning of the line.
<Ctrl>g-<Ctrl> l	moves the cursor to the end of the line.
<Ctrl>v	toggles into or out of overstrike mode.
	deletes character over the cursor.
<Bksp>	backs up and deletes one character (left of cursor).
<Ctrl>u	pages up.
<Ctrl>d	pages down.
<Ctrl>n	goes to the next word.
<Ctrl>p	goes to the previous word.
Left, Right Arrow	moves left and right within the edit line.

Table 2.5
Point-and-Pick Keys

Keystroke	Action
<Return>	selects the item.
<Esc>	ends the selection process. The list is removed and no action is performed.
<Ctrl>v	toggles between selecting all or none of the items appearing in a list.
Up, Down Arrow	moves to other items in a list.
Left, Right Arrow	moves across a multicolumn display.
<Space>	marks items when the application accepts more than one. A marked item is indicated by a "*" character in the left column. It may be unmarked by pressing <Space> a second time while on the item. The entire collection of marked items is selected by pressing <Return>.
<F5>	finds items in long listing. It is called the "Search" key. A prompt appears and you enter the string to search for, and press <Return>. If the item is found the highlight moves to that item, and another <Return> selects the item. If no match is found, the highlight does not move. The ; and : keys repeat the previous search forward and backward, respectively.
First letter	selects an item by its first letter. It is the fastest method of selection. Pressing <Return> selects the highlighted item. (If there is only one item beginning with that letter, it is marked by typing its first letter. There is no need to press <Return> again.) If several items begin with the same letter, the cursor moves to the first occurrence in the list.

Using Scan Windows

When you execute a XENIX command by selecting a **sysadmsh** menu option, the result of the command is typically displayed in a scan window. Scan windows also display the contents of files and directory listings. To demonstrate the use of scan windows, let's say you want to know who is currently logged on to the system. To do this, select System→Report→Users. (This runs the XENIX **who(C)** command.)

When you select the Current option, a scan window displaying the output of the **who(C)** command appears in the display area:

```

Users
-----
<esc> to exit, movement keys are active
31 Aug 90 13:06 who -H
----- who (C) -----
NAME      LINE      TIME
root      tty01     24 May 10:23
faithz    tty1a     24 May 11:03
stevem    tty1b     24 May 8:16
naomib    tty1c     24 May 8:00
terib     tty1d     24 May 8:16
martinm   tty1e     24 May 9:09
mattb     tty1f     24 May 7:49
teresae   tty1g     24 May 10:29
danju     tty1h     24 May 10:05
docadm    tty2a     24 May 11:05
stuartc   tty2b     24 May 8:26

```

Note that the name of the command (**who**) and the reference section in which its description can be found (**C**) are displayed at the top of the window. Also note that the option given to the command (**-H**) is displayed in the right hand side of the Status Line. If you do not understand the information displayed, look up the proper manual page for more information.

Using Scan Windows

You can recognize a scan window by the vertical “scroll bar” that appears at the extreme right edge of the window. When the window is at the top of your text, the asterisk symbol (*) at the top of the scroll bar is visible. If it is at the bottom, the plus symbol at the bottom of the scroll bar is visible. You can see both asterisks when the window contains all of the text.

The scroll bar also indicates where you are in the window. The highlighted portion of the bar represents the section of text that is currently displayed in the window. As you scroll up and down, the highlighted bar moves with you.

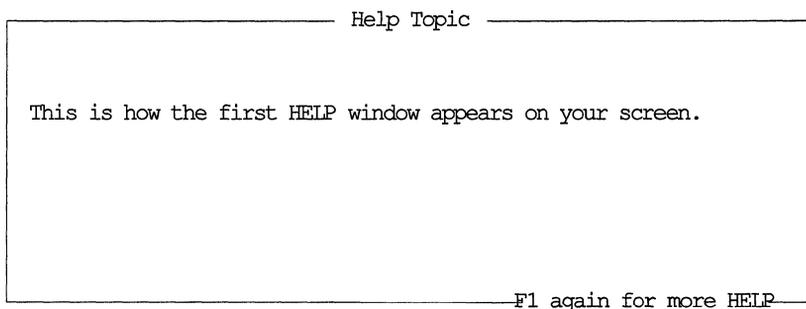
Use the keys listed in Table 2.6 when you are in a scan window.

Table 2.6
Scan Keys

Action	Keystroke
<Esc>	exits the file.
Up Arrow	moves up one line.
Down Arrow or <Return>	moves down one line.
<PgDn> or <Space>	moves down a page.
<PgUp>	moves up a page.
<Home>	moves to top of display.
<End>	moves to bottom of display.
<F5>	searches for a pattern in display. (; and : repeat the search forward and backward, respectively)
<F7>	send the command output or file from the scan window to the printer.

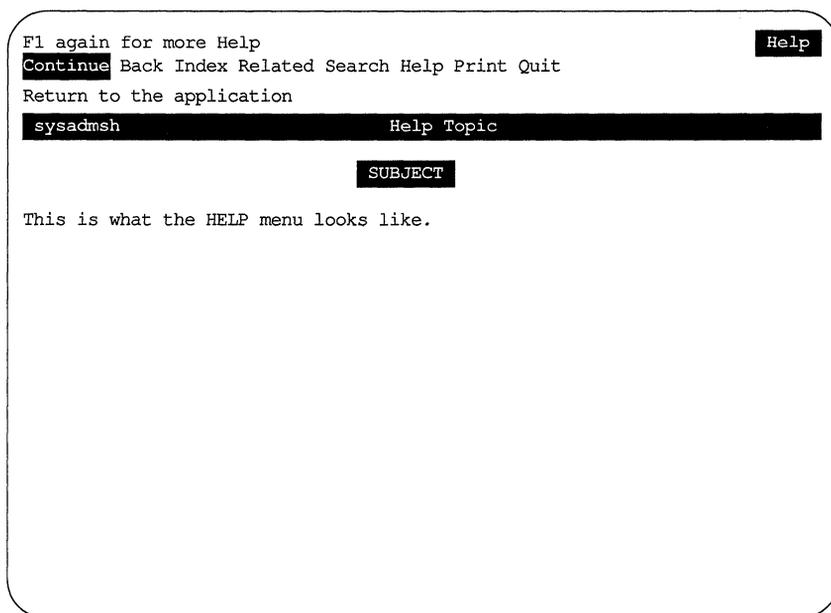
Getting Help

You can press the <F1> key to display more information to help you with your selection. When you press the <F1> key, a Help window opens within your current screen. It looks similar to this:



2

The window contains some basic information. If you need more help, you can press <F1> again and the complete Help menu is displayed:



When you are finished, select Quit from the Help menu and you return to your place in the **sysadmsh** menu.

Getting Help

The menu options for Help are listed in Table 2.7.

Table 2.7
Help Options

Option	Action
Continue	continues on to the next page. All the vertical movement keys are active: Up and Down Arrows, Page Up and Down, Home and End. If there is no further information, the highlight moves to the Quit option on the Help menu and the description line reads "Return to the application."
Back	moves back to topics that were seen previously. There is no corresponding "Forward." This also backs up to more general topics. You can go back until the top-level introductory topic is reached.
Index	chooses a new topic from a list of indexed topics.
Related	chooses a new topic related to the current one.
Search	searches for a new topic by matching a pattern. First, you specify where to look (the titles, the text lines, or both), and then give the pattern. The pattern can be a simple keyword (like "create" or "date") or a more complex "regular expression." A list of topics containing the pattern is presented.
Help	explains how the help facility itself is used. A table similar to this one is displayed on the screen. If you need further information, look for your topic in Index, Related, or Search.
Print	Make a hardcopy printout of the current topic. First, you select which printer to use, then choose Go from the submenu to send the printout.
Quit	exits Help and returns to <code>sysadmsh</code> . <code><F2></code> or <code><Esc></code> are other ways to exit quickly.

Each Help screen has general information available, as well as specific information about each option listed on the menu from which Help was selected. Each descriptive passage is preceded by the associated menu line and followed by a reference to the operating system documentation.

Note

When you are within an actual XENIX command, you do not have access to the Help facility. For example, when you select Dirs/Files→Edit, you are within a file editor, and the **sysadmsh** keys no longer function. When you exit the command and return to the **sysadmsh**, the keys function as expected. If no element of the **sysadmsh** is visible on the screen (menu line, boxes, context indicator, and so on) then Help probably is not available. If you need help, exit from the current process and press the ⟨F1⟩ key to view Help. In general, it is best to use Help prior to executing a menu selection.

Changing the Current Directory within sysadmsh

There are many occasions when it is necessary to change your current directory to use certain files and commands. You can move to another directory by pressing the `<F6>` key. The current directory is displayed at the top of the screen. You can use the `<Bksp>` key to erase the name of the current directory (to start over), or you can add to or alter part of the current name. When you press `<Return>`, your directory change is executed and reflected on the status line.

The Function Keys

The function keys give you access to several time-saving features. See Table 2.8.

Table 2.8
Function Keys

2

Key	Action
<F1> Help key	displays help for the current context within the application. Further information is available by pressing <F1> again.
<F2> Exit key	activates the Quit option on the top menu level. Press n to return to sysadmsh .
<F3> Pop-up key	(used within a form) displays a list of items that are acceptable for the current field.
<F4> Spell key	(used within a form) displays a list of words that are possible correct spellings of the word in the current field. Select a word from the list by pressing <Return>. The word is then placed in the field.
<F5> Search key	(used within a window) prompts for a string to search for. When you enter a string and press <Return>, the screen scrolls to the location of the string, if it exists.
<F6> New directory key	offers the opportunity to change your current working directory. Note that this does not change the directory you will return to upon leaving sysadmsh .
<F7> Print key	prints the output of any command that is displayed in a scan window.

Using Shell Escapes to Access the XENIX Command Line

You can execute a XENIX command from within a **sysadmsh** menu by typing the shell-escape character, an exclamation point (!). The menus are replaced by a subshell that displays a text-entry line and a prompt asking for a command. When you enter the command and press <Return>, the command is executed by the shell. After the command is completed, the output is displayed on the screen, and you are prompted to press any key to return to the shell.

Note

The XENIX command line can only be accessed from the shell menus. It cannot, for example, be accessed from a form or a point-and-pick list.

sysadmsh Environment Variables

The **sysadmsh** uses the following environment variables, which can be defined in user *.login* or *.profile* files:

SA_EDITOR	If not set, the default editor is Lyrix if installed, or vi(C) if Lyrix is not available.
SA_MAIL	If not set, the default mailer is SCO Portfolio email if installed, or regular XENIX mail (C) if not.
SA_PRINT	If not set, the default printer device is <i>/dev/lp</i> .

2

sysadmsh Menu Options

Table 2.9 lists the options available in the **sysadmsh** menus. The main or top-level menu options appear in boldface type at the left margin of the table. The options listed below each top-level menu option are in lower levels of the menu hierarchy. Those with one arrow are in the first-level menus, those with two arrows are in the second-level menus, and so on. For instance, at the beginning of the table, Report is a first-level menu option in the System menu, and Activity is an option in the second-level Report menu.

Once you are familiar with the menu options, you can enter shorthand menu paths to reach lower level options without having to see all of the upper level menus. In Table 2.9, the "Path" column lists the shorthand menu path for each menu option. Enter the first letter of each menu option in the path down to the option you want. For example, to choose the Activity option, enter the first letters of system, report, and activity, **sra**, without pressing the (Return) key.

If there are no menu options listed below the one you choose, either a form or a display appears. For example, when you choose the Activity option, a scrollable display or scan window appears, showing system processes.

Table 2.9
sysadmsh Menu Map

sysadmsh option	Path	XENIX command	Function
Dirs/Files	d		interacts with files and directories.
→List	dl	lc(C)	lists files in current directory.
→View	dv	cat(C)	views contents of file.
→Copy	dc	copy(C)	copies directory or file.
→Edit	de	SCO Lyrinx, vi(C), ed(C)	edits one or more files.
→Modify	dm		changes file parameters.
→→Permissions	dmp	chmod(C)	changes file permissions.
→→Ownership	dmo	chown(C)	changes file ownership.
→→Group	dmg	chgrp(C)	changes file group ownership.
→→Name	dmn	mv(C)	renames or moves files.
→→Size	dms	compress(C)	compacts files.
→→Format	dmf	translate(C)	changes file formats.
→Print	dp	lp(C)	prints files.
→Archive	da	tar(C) -c	stores files.
→Differences	dd	dircmp(C) or diff(C)	compares two text files or directories.
→Remove	dr	rm(C) -rf	removes specified files or directories.
→UseDOS	du		uses DOS utilities to manipulate DOS files.
→→List	dul	dos(C): dosdir or dosls	lists DOS files in current directory.
→→Remove	dur	dos(C): dosrmdir or dosrm	removes a DOS file or directory.
→→MakeDir	dum	dos(C): dosmkdir	creates a DOS directory.
→→Copy	duc	dos(C): doscp	copies files between DOS and XENIX.
→→View	duv	dos(C): doscat	displays DOS files.
→→Format	duf	dos(C): dosformat	formats DOS media.
Processes	p		
→Report	pr	ps(C) -ef	lists current processes.
→Terminate	pt	kill(C)	terminates a process.
Users	u		
→Report	ur	who(C) -H	lists users currently logged in.
→Add	ua	mkuser(ADM)	adds a user account.
→Delete	ud	rmuser(ADM)	removes a user account.
→Modify	um		
→→Username	umu		changes a user name.
→→Passwd	ump	passwd(C)	changes a user password.
→→Shell	ums		changes a user's login shell.
→→Environment	ume	edit .profile or .login	modifies .profile or .login.
→→Group	umg		modifies user group membership.
→Communicate	uc	mail(C) or SCO Portfolio email	sends mail to a user.

(Continued on next page.)

sysadmsh Menu Options

sysadmsh Menu Map (Continued)

sysadmsh option	Path	XENIX Command	Function
Backups	b		performs backups of files.
→Create	bc	sysadmin(ADM) , tar(C) -c, dd(C) or a utility similar to diskep(C)	creates backups.
→Restore	br	tar(C) -x or sysadmin(ADM)	restores filesystems and files.
→Schedule	bs	edit /usr/lib/sysadmin/schedule	modifies scheduled backup frequency and device used.
System	s		system-wide reports, configurations.
→Report	sr		reports on current state of system.
→→Activity	sra	ps(C) -ef	reports on current system activity.
→→Users	sru	who(C) -H	reports users currently logged in.
→→Printers	srp	lpstat(C) -t	reports on current status of printers.
→→Disk	srd	df(C) -v -i	reports on current disk usage.
→→Network	srn		reports on current network files.
→→→Xnet	srnx	xnstatus(NADM)	reports on XENIX-NET status.
→→→Micnet	srnm	netutil(ADM)	reports on Micnet status.
→→→UUCP	srnu	uustat(C)	reports on UUCP status.
→→Messages	srn		reads system messages.
→→→Mail	srmm	mail(C) -u root	reads root's mail.
→→→Console	srnc	tail(C) /usr/adm/messages	displays recent console messages.
→→Software	srs	custom(ADM)	checks installed software/package status.
→Halt	sh	shutdown(ADM)	shuts down system to remove power or reboot.
→Execute	se	run script in /usr/lib/sysadm/local	executes programs that are system specific.
→Configure	sc		configures system files.
→→Time	sct	asktime(ADM)	sets the system time.
→→Autologout	sca	idleout(ADM)	
→→Kernel	sck	configure(ADM)	changes kernel parameters or capabilities.
→→Network	scn		configures networking files.
→→→UUCP	scnu	uuinstall(ADM)	configures UUCP files, enable/disable tty.
→→Defaults	scd		modifies system default parameters.
→→→Message	scdm	edit /etc/motd	modifies message of the day file.
→→→Checklist	scdc	edit /etc/checklist	lists filesystems checked at startup.
→→→Other	scco	edit any file in /etc/default	modifies other default files.

(Continued on next page.)

sysadmsh Menu Map (Continued)

sysadmsh option	Path	XENIX Command	Function
→→International	sci		configures system for international use.
→→→System	scis	<i>edit /etc/default/lang</i>	sets system locale variables.
→→→Individual	scii	<i>edit .profile or .login</i>	changes a user's default locale.
→→→Display	scid	mapchan(F)	changes mapping of terminal character set.
→→→Keyboard	scik	modifies <i>/usr/lib/keyboard/keys</i>	changes mapping of console keyboard.
→Add	sa		
→→HardDisk	sah	mkdev(ADM) hd	adds a hard disk.
→→Printer	sap	mkdev(ADM) lp	adds a printer.
→→Software	sas	custom(ADM)	adds software.
→→Card_Serial	sac	mkdev(ADM) serial	adds a multiport card.
→Delete	sd		
→→Printer	sdp	mkdev(ADM) lp	removes a printer.
→→Software	sds	custom(ADM)	removes software.
Filesystems	f		checks/acts on filesystems.
→Check	fc	fsck(ADM)	checks and repairs filesystem.
→Mount	fm	mount(ADM)	mounts a filesystem.
→Unmount	fu	umount(ADM)	unmounts a mounted filesystem.
Media	m		reads, copies, compares, formats floppies/tapes.
→List	ml	tar(C) -t	lists the contents of a floppy or tape.
→Extract	me	tar(C) -x	extracts the contents of a floppy or tape.
→Archive	ma	tar(C) -c	stores files/directories/filesystems on media.
→Format	mf	format(C) or dos(C): dosformat	formats either a XENIX or DOS floppy.
→Duplicate	md	similar to diskcp(C)	makes a copy of floppy or tape.
→Tapedump	mt	tapedump(C)	displays physical contents of tape.
Quit	q		exits sysadmsh.

Chapter 3

Starting and Stopping the System

Introduction 3-1

Starting the System 3-2

 Loading the Operating System 3-2

 Cleaning Filesystems 3-3

 Choosing the Mode of System Operation 3-3

 Setting the Time and Date 3-5

Logging in as the Super User 3-7

Stopping the System 3-9

 Using the wall Command 3-9

 Using the shutdown Command 3-9

 Using the haltsys Command 3-10

Understanding the Boot Display Information 3-12

Changing the Boot Process 3-14

 Changing the /etc/default/boot File 3-14

 Booting Automatically 3-15

Introduction

This chapter explains how to do the following:

- start and stop your system
- log in as the super-user (root)
- change the system boot procedure
- use the device and system configuration information displayed at boot time.

3

Additional information on customizing the system startup process is found in the “Customizing System Startup” appendix in this guide.

Starting the System

Starting a XENIX system requires more than just turning on the power. You must also perform a series of steps to initialize the system for operation. To start the system, you must do the following:

- Load the operating system.
- Check the filesystems (if the system was improperly stopped).
- Choose the mode of system operation.

The following sections describe each of these procedures.

Loading the Operating System

The first step in starting the system is to load the operating system from the computer's hard disk.

1. Turn on power to the computer and hard disk. The computer loads the XENIX bootstrap program and displays this message:

```
Boot
:
```

2. Press the <Return> key. The bootstrap program loads the operating system.

When the system is loaded, it displays information about itself and checks to see if the root filesystem (that is, all files and directories) is in order and not corrupted. If a filesystem is uncorrupted and in good order, it is called "clean." If the root filesystem is clean, you can choose the mode of operation. If not, the system requires you to clean the filesystem before choosing.

Cleaning Filesystems

You must clean the filesystem if the system displays the message:

```
Proceed with cleaning (y or n)?
```

This message is displayed only if the system was not stopped properly, as described in the section “Stopping the System” later in this chapter.

Each filesystem generates this message. The operating system requires clean filesystems to work properly. If the above message does not appear, your filesystem is clean and ready to use.

To clean the filesystem, enter **y** (for “yes”) and press the **<Return>** key. The **fsck(ADM)** utility cleans the filesystem, repairing damaged files or deleting files that cannot be repaired. It reports on its progress as each step is completed. At some point, you may be asked if you wish to salvage a file. Always answer by entering **y** or **n** and pressing the **<Return>** key. For an explanation of how **fsck** works, see “Filesystem Integrity” in the “Managing Filesystems” chapter later in this guide.

When cleaning is complete, the system asks you to choose the mode of operation.

Choosing the Mode of System Operation

You may choose the mode of operation as soon as you see the message:

```
Type CONTROL-d to continue with normal startup,  
(or give the root password for system maintenance):
```

The system has two modes: normal operation and system maintenance. (Normal operation is known as multiuser mode, while system maintenance mode is known as single-user mode.) Normal operation is for ordinary work on the system. This is the mode that allows multiple users to log in and begin work. System maintenance mode is reserved for work to be done by the system administrator. It does not allow multiple users.

To choose normal operation, press **<Ctrl>d**. The system displays a startup message, and you are prompted to enter the system time (see the next section). Then the system executes commands found in the *etc/rc* files and directories, generating startup messages for the various system services, such as the printer or network services. (These scripts are described later

Starting the System

in this chapter.) Next, the system displays the “login:” prompt. You can now log in as a normal user, as described in the “Logging In” chapter of the *XENIX Tutorial*, or as the super-user, as described later.

To choose system maintenance mode, enter the super-user password (also called the “root password”) and press `(Return)`.

Note

The super-user (root) password is assigned during system installation. If you do not know the root password, ask the administrator who installed your system.

Unless you are entering system maintenance mode, you are asked to set the system time, and this request is followed by the super-user prompt (`#`). The commands in the */etc/rc* scripts are not executed. (Choose system maintenance mode only if you must do system maintenance work that requires all other users to be off the system.) When you log out of system maintenance mode using `(Ctrl)d`, the system automatically enters normal operation.

Entering System Maintenance Mode by Shutting Down

To go from normal operation to system maintenance mode, log in as *root* and give the following command to shut down the system:

```
/etc/shutdown n
```

`Δ sysadmsh` users select: System→Halt

where *n* is the number of minutes until multiuser mode is stopped. You are asked to confirm your choice and to press any key to reboot.

Entering System Maintenance Mode Directly

To go from normal operation to system maintenance mode directly, log in as *root* and give the following command:

```
/etc/shutdown su
```

The `su` indicates that you want to go directly into single user mode rather than shut the system down.

Note

There is no **sysadmsh** equivalent for this command.

Setting the Time and Date

Once normal operation starts, the system asks for the correct time and date. It displays the current time and date and then the following message:

```
Current System Time is Wed Nov 29 08:19:00 PST 1989
Enter new time ([yymmdd]hhmm):
```

Unless your clock battery is drained or removed, there should be no need to change the date. To leave the time and date unchanged, simply press **<Return>**. If you need to change the time and date, enter the new time and press **<Return>**. The new values must be entered as two or more consecutive pairs of digits, where the digits may be one or more of the following:

- | | |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>yy</i> (optional) | represents the last two digits of the current year. It may be any two-digit value, from 00 to 99. |
| <i>mm</i> (optional) | represents the current month. It may be any two-digit value, from 01 to 12 for the months January to December, respectively. |
| <i>dd</i> (optional) | represents the current day. It may be any two-digit value, from 01 to the last day of the month. |
| <i>hh</i> | represents the current hour. It may be any two-digit value, from 00 to 23. Hours are expressed in military time, where morning hours range from 00 to 11 and evening hours from 12 to 23. |
| <i>mm</i> | represents the current minutes. It may be any two-digit value, from 00 to 59. |

For example, to change the time and date to February 3, 1991 at noon, enter:

9102031200

Starting the System

Press `<Return>`. After accepting the new value, the system then displays the new time and date:

```
Sun Feb 03 12:00:00 PST 1991
```

If you enter an incorrect value, the system prompts you to try again. If you do not enter an optional value, the current value for that item remains unchanged. If you type a new value for the year, you must also type values for the month and day. Similarly, if you type a new value for the month, you must type a value for the day.

The time and date is followed by service startup messages and the “log-in:” message.

Logging in as the Super User

Many system maintenance tasks, when performed during normal operation, require you to log in as the super-user. For example, you must be logged in as the super-user to stop the system.

To log in as the super-user, you must know the super-user password. You also need to see the “login:” message on your terminal’s screen. If you do not see this message, press <Ctrl>d until it appears.

3

Note

The super-user (root) password is assigned during system installation. If you do not know the root password, ask the administrator who installed your system.

To log in as the super-user, follow these steps:

1. When you see the “login:” message, enter the super-user login name:

root

Now press the <Return> key. The system prompts you for the super-user password.

2. Enter the super-user password and press the <Return> key. The system does not display the password as you enter it, so enter each keystroke carefully.

The system opens the super-user account and displays the message of the day and the super-user prompt (#).

Logging in as the Super User

Take special care when you are logged in as the super-user. In particular, you should be careful when deleting or modifying files or directories. This is important because the super-user has unlimited access to all files; it is possible to remove or modify a file that is vital to the system. Avoid using wildcard designators in filenames and keep track of your current working directory. (Wildcard designators are described in the *XENIX Tutorial* in the “Special Characters” section of the “Basic Concepts” chapter under “Naming Conventions.”)

You can leave the super-user account at any time by pressing (Ctrl)d.

Stopping the System

Stopping a XENIX system requires more than just turning off the computer. You must prepare the system for stopping by using either the **shutdown**(ADM) or (under certain conditions) the **haltsys**(ADM) command. The following sections describe each command.

Using the wall Command

3

Before stopping the system with the **shutdown**(ADM) command, you should notify users of the impending shutdown. You may want to include other details, such as when the system will be restarted.

To send a system-wide message to the terminals of all the users that are currently logged in, use the **wall**(ADM) (“write to all”) command:

```
wall
```

Press <Return>. Enter the message, pressing <Return> to start a new line if necessary. When you have finished entering the message, press <Ctrl>d. This displays your message on all system terminals.

Using the shutdown Command

The **shutdown** command is the normal way to stop the system and should be used whenever the system is in normal operation mode. It warns other users that the system is about to be stopped and gives them an opportunity to finish their work. The warning message that **shutdown** displays at all terminals can be customized. (If desired, the system administrator can also use the **wall**(ADM) command to send a message about the impending shutdown prior to running the actual **shutdown** command.)

Stopping the System

To stop the system with the **shutdown(ADM)** command, follow these steps:

1. Log in as the super-user. See the section “Logging in as Super-User” earlier in this chapter. The system opens the super-user account and displays the message of the day and the super-user prompt.
2. Enter the following command and press **<Return>**:

```
/etc/shutdown
```

Δ sysadmsh users select: System→Halt

3. You are then prompted for the number of minutes until shutdown:

```
Minutes till shutdown? (0-15):
```

Enter a number from 0 to 15 and press **<Return>**. The system displays a warning message at each terminal, asking logged-in users to finish their work and to log out. As soon as all users are logged out or the specified time has elapsed, the system closes all accounts and displays the following message:

```
** Normal System Shutdown **  
  
** Safe to Power Off **  
-or-  
** Press Any Key to Reboot **
```

4. Turn off the computer or press any key to reboot the system.

Using the **haltsys** Command

The **haltsys(ADM)** command halts the system immediately. This command should be used only when in single user mode. If there are any users logged into the system when the **haltsys** command is given, they are logged out and their work in progress is lost. In addition, network servers and other programs are terminated abnormally and could create problems when they are restarted.

To stop the system with the **haltsys** command, follow these steps:

1. You should be in single-user mode. (Remember that the **haltsys** command should not be used in multiuser mode.)
2. Enter:

/etc/haltsys

Now press the `<Return>` key. The system displays the following message:

```
** Normal System Shutdown **  
  
** Safe to Power Off **  
    -or-  
** Press Any Key to Reboot **
```

3

3. Turn off the computer, or press any key to reboot the system.

Understanding the Boot Display Information

At boot time, a table of hardware information is always displayed after the copyright information. This table represents your hardware configuration, as it is recognized by the operating system. Figure 3-1 is an annotated version of the boot screen, as it appears on a sample machine.

device	address	vector	dma	comment
fpu	-	35	-	type=80387
floppy	0x03F2-0x03F7	06	2	unit=0 type=96ds15
serial	0x02F8-0x02FF	03	-	unit=1 type=Standard nports=1
parallel	0x0378-0x037A	07	-	unit=0
console	-	-	-	unit=ega type=0 12 screens=68k
disk	0x01F0-0x01F7	36	-	type=W0 unit=0 cyls=791 hds=16 secs=48

Figure 3-1 Sample Boot Display

This key explains the sample display in Figure 3-1:

device, address, vector, dma, comment	The name of the hardware, address in hexadecimal, interrupt vector, direct memory access channel, other details about the hardware, respectively
fpu	floating-point unit present, specifically the Intel 80387 chip
floppy	high density floppy drive
serial	this is COM 1; COM 1 has one port (no multiport card is installed)
parallel	this is your parallel port, <i>/dev/lp0</i>
console	The console has an EGA video adaptor compatible with (type 0) the IBM EGA design
disk	Western Digital st506 controller number 0 (W0), hard drive 0 (unit 0), as well as the number of cylinders, heads, and sectors per track

Understanding the Boot Display Information

The **hwconfig(C)** utility displays or accesses this information at any time, using the configuration information stored in the file */usr/adm/hwconfig*. Refer to the **hwconfig(C)** manual page for more information.

Changing the Boot Process

Each time the computer is started, the system runs the **boot** program. Unless you give different instructions at the prompt, **boot** loads the default kernel program using the configuration values specified in the file */etc/default/boot* on the default root filesystem. You can edit the */etc/default/boot* file to change the default configuration values for future boot operations. You can also set certain options in */etc/default/boot* to allow the system to boot automatically.

Changing the */etc/default/boot* File

The **boot(HW)** manual page describes the default boot options that you can change by editing the */etc/default/boot* file.

To change which program is loaded by default when you enter only a **<Return>** at the boot prompt, modify the default bootstring set with the **DEFBOOTSTR** option in */etc/default/boot*. For example, the following setting in */etc/default/boot* causes the **boot** program to load the kernel from a hard disk by default.

```
DEFBOOTSTR=hd(40)xenix
```

The first two letters of the argument specify the device (hd for hard disk or fd for floppy disk). The number in parentheses is the minor device number (40 for the root filesystem on the hard disk). Following the parentheses is the pathname of the program to be loaded.

The *Release Notes* and **boot(HW)** manual page describe certain keywords that you can add to the **boot** command line to deal with special hardware situations. You can also add these keywords to the end of the default bootstring set with **DEFBOOTSTR**. For example, to use an Intel Inboard™ in high-speed mode automatically every time you boot, add the keyword **inboard** to the default boot string:

```
DEFBOOTSTR=hd(40)xenix inboard
```

Booting Automatically

The settings of the `LOADXENIX`, `TIMEOUT`, and `PANICBOOT` options in `/etc/default/boot` control if, and when, automatic booting occurs.

If `LOADXENIX=NO`, the `boot` program waits indefinitely for a response to the prompt. You can set `LOADXENIX=YES` to allow the system to boot automatically if no response is given at the prompt after a certain amount of time. The amount of time `boot` waits for a response before booting automatically is 60 seconds by default, but you can set the number of seconds to wait with the `TIMEOUT` option. If a timeout occurs, `boot` behaves as though you entered a `<Return>` to the boot prompt, performing the default boot process using the configuration values specified in `/etc/default/boot`.

3

You can set the `PANICBOOT` option to `YES` or `NO` to indicate whether or not the system should reboot after a panic (a panic always causes the system to halt).

Refer to the `autoboot(ADM)` manual page for more information.

Chapter 4

Managing Filesystems

- Introduction 4-1
- What Is a Filesystem? 4-2
- Mounting and Unmounting a Filesystem 4-3
- Permitting Users to Mount Filesystems 4-4
- Maintaining Free Space in Filesystems 4-6
 - Strategies for Maintaining Free Space 4-6
 - Displaying Free Space 4-7
 - Sending a System-Wide Message 4-8
 - Displaying Disk Usage 4-8
 - Displaying Blocks by Owner 4-9
 - Mailing a Message to a User 4-9
 - Locating Files 4-9
 - Finding Temporary Files 4-11
 - Finding Files of a Certain Size 4-11
 - Finding Files by Permissions 4-11
 - Executing Commands Based on find Output 4-12
 - Locating core Files 4-13
 - Checking and Clearing Log Files 4-14
 - Automatic Cleaning of Log Files 4-15
- Maintaining Efficient Filesystem Organization 4-17
 - Disk Fragmentation 4-17
 - Large Directories 4-18
 - Limiting Login Directory Size 4-18
 - Removing Excess Inode Numbers 4-18
- Adding Disk Space and Filesystems 4-20

Filesystem Integrity	4-21
How XENIX Systems Maintain Files	4-22
How XENIX Systems Maintain Filesystems	4-23
Causes of Filesystem Corruption	4-24
Rules for Checking Filesystems	4-25
Repairing Filesystems with fsck	4-25
Summary of fsck Phases	4-27
Automatic Filesystem Check	4-28

Introduction

This chapter describes one of the most important responsibilities of a system administrator: creating and maintaining filesystems. General maintenance activities are described, such as strategies for maintaining free space. The concept of “filesystem integrity” is introduced, and how the operating system repairs damaged filesystems. Filesystem creation is discussed in the “Adding Hard Disks” chapter of this Guide. For information on file permissions and other security considerations, see the “Maintaining System Security” chapter in this guide.

What Is a Filesystem?

The area of the hard disk occupied by the XENIX system is known as the hard disk partition. (If XENIX is the only system on the hard disk, the XENIX partition occupies the entire hard disk.) A filesystem is a distinct structure within the XENIX partition, consisting of files, directories, and the information needed to locate and access them. A filesystem can be thought of as a structure upon which directories and files are constructed.

Each XENIX system has at least one filesystem on the primary hard disk. This filesystem is called “the root filesystem” and is represented by the symbol “/”. The root filesystem contains the programs and directories that comprise the operating system. On small hard disks, the root filesystem includes all the user directories as well. The primary hard disk can also be divided into more than one filesystem. This is described in the *XENIX Installation Guide*. One of the most common divisions is the `/u` filesystem, used to isolate user accounts from the root filesystem. For more details on these filesystems, see the *XENIX Installation Guide*.

A XENIX system can also have other filesystems that contain special directories and application programs. Dividing the primary hard disk into multiple filesystems protects the data and makes maintenance easier. Adding still more filesystems by installing other hard disks expands the system storage space. New filesystems can be specifically created by the system administrator, then “attached” (mounted) and “detached” (unmounted) when needed, in the same way that a floppy disk is accessed. This process is described in the “Adding Hard Disks” chapter in this guide.

Mounting and Unmounting a Filesystem

The **mount**(ADM) command attaches a filesystem. A filesystem is “attached” or mounted upon a directory called a *mount point*. You must specify the name of filesystem you are mounting. For example, to mount or unmount */dev/u* on mount point */u*, you would use the following two commands, respectively:

```
mount /dev/u /u
```

△ **sysadmsh** users select: Filesystems→Mount

```
umount /dev/u
```

△ **sysadmsh** users select: Filesystems→Unmount

Only the super user can use the **mount** and **umount** commands.

Files in a filesystem are not accessible unless the filesystem is mounted. When a filesystem is unmounted, the mount point appears to be an empty directory. If files are copied to or created in the mount point directory while the filesystem is unmounted, those files will appear to be in that filesystem when they are not. When the filesystem is mounted, these files seem to “disappear” when the filesystem is mounted over them.

Permitting Users to Mount Filesystems

Only the super user can use the **mount** command. However, the super user can set up parameters to define which filesystems can be mounted by users with the **mnt(C)** command, including the use of an access password, if desired.

Each filesystem must have an entry in the file */etc/default/filesys*. Figure 4-1 contains a sample set of entries.

```
bdev=/dev/root cdev=/dev/rroot mountdir=/ \
desc="The Root Filesystem" rcmount=no mount=no

bdev=/dev/u cdev=/dev/ru mountdir=/u rcmount=yes \
fsckflags=-y desc="The User Filesystem"

bdev=/dev/x cdev=/dev/rx mountdir=/x mount=yes \
rcmount=yes fsckflags=-y desc="The Extra Filesystem"
```

Figure 4-1 Sample */etc/default/filesys* File

The sample entries determine the behavior shown in Table 4.1.

Table 4.1
Sample Filesystem Mount Specifications

Filesystem	When Mounted	Can User Mount?
<i>root</i>	boot time	no
<i>/u</i>	multiuser	no
<i>/x</i>	anytime	yes

Permitting Users to Mount Filesystems

If you wish to make any non-root filesystem mountable by users, simply add “mount=yes” to the entry for the given filesystem. In addition, when the **mnt** command is invoked without an argument (no filesystem name), the program checks all non-root filesystems to see if they can be mounted and, if so, mounts them. The option “mount=prompt” asks the user if they want to mount each filesystem where a mount is permitted.

For more information on the **mnt** command, including a complete list of options, refer to the **mnt(C)** manual page.

Maintaining Free Space in Filesystems

An important task of the system administrator is filesystem maintenance, which includes keeping the system running smoothly, keeping the filesystems clean, and ensuring adequate space for all users. To maintain the filesystems, the system administrator must monitor the free space in each filesystem, and take corrective action whenever the free space gets too low.

This section explains the filesystem maintenance commands. These commands report how much space is used, locate seldom-used files, and remove or repair damaged files.

A XENIX system operates best when at least 15% of the space in each filesystem is free. In any system, the amount of free space depends on the size of the disk containing the filesystem and the number of files on the disk. Because every disk has a fixed amount of space, it is important to control the number of files stored on the disk.

If a filesystem has less than 15% free space, system operation usually becomes sluggish. If no free space is available, the system stops any attempts to write to the filesystem. This means that the user's normal work on the computer (creating new files and expanding existing ones) stops.

The only remedy for a filesystem that has less than 15% free space is to delete one or more files from the filesystem. The following sections describe strategies for keeping free space available.

Strategies for Maintaining Free Space

The system administrator should regularly check the amount of free space on all mounted filesystems and remind users to keep their directories free of unused files. You can remind users by including a reminder in the */etc/motd* (message of the day) file.

If the amount of free space slips below 15%, the system administrator should follow these steps:

1. Send a system-wide message asking users to remove unused files.
2. Locate exceptionally large directories and files, and send mail to the owners asking them to remove unnecessary files.
3. Locate and remove temporary files and files named *core*.
4. Clear the contents of system log files.
5. Reduce disk fragmentation by making a complete backup of the filesystem, removing all the files, and then restoring them from the backup.
6. If the system is chronically short of free space, it may be necessary to create and mount an additional filesystem.

The above suggestions are described in detail in the following sections.

4

Displaying Free Space

You can find out how much free space exists in a particular filesystem with the **df** (for “disk free”) command. This command displays the number of “blocks” available on the specific filesystem. A block is 512 characters (or bytes) of data.

The **df** command has the form:

df *specialfile*

△ **sysadmsh** users select: System→Report→Disk

where *specialfile* can be the name of a XENIX special file corresponding to the disk drive containing the filesystem. If you do not give a special filename, then the free space of all normally mounted filesystems is given.

For example, to display the free space of the root filesystem */dev/root*, enter:

df /dev/root

Press <Return>. The command displays the special filename and the number of free blocks.

Maintaining Free Space in Filesystems

You can find the percentage of free space on your system with the command:

```
df -v
```

Sending a System-Wide Message

If free space is low and you need users to do something about it immediately, you can send a message to all users on the system with the **wall** (for “write to all”) command. This command copies the messages you enter at your terminal to the terminals of all users currently logged in.

To send a message, enter:

```
wall
```

Press **<Return>**. Enter the message, pressing **<Return>** to start a new line if necessary. After you have entered the message, press **<Ctrl>d**. The command displays the message on all terminals in the system. To leave the **wall** command, press **<Ctrl>d**. This removes the connection to other terminals.

Displaying Disk Usage

You can display the number of blocks used within a directory by using the **du** command. This command finds excessively large directories and files.

The **du** command has the form:

```
du directory
```

The optional *directory* must be the name of a directory in a mounted filesystem. If you do not give a directory name, the command displays the number of blocks in the current directory.

For example, to display the number of blocks used in the directory */u/johnd*, enter:

```
du /u/johnd
```

Press **<Return>**. The command displays the name of each directory in the */u/johnd* directory and the number of blocks used.

Displaying Blocks by Owner

You can display a list of users and the number of blocks they own by using the **quot** (for “quota”) command. The command has the form:

```
quot specialfile
```

The *specialfile* must be the name of the special file corresponding to the filesystem.

For example, to display the owners of files in the filesystem on */dev/u*, enter:

```
quot /dev/u
```

Press **<Return>**. The command displays the users who have files in the filesystem and the numbers of blocks in these files.

4

Mailing a Message to a User

If a particular user has excessively large directories or files, you can send a personal message to the user with the **mail** command.

To begin sending a message through the mail, enter:

```
mail login-name
```

Press **<Return>**. The *login-name* is the login name of the recipient. To send a message, enter the message, press **<Return>**, and then press **<Ctrl>d**. If the message has more than one line, press **<Return>** at the end of each line. The **mail** command copies the message to the user’s mailbox, where it can be viewed by the user via the **mail** command. See the *XENIX User’s Guide* for details.

Locating Files

You can locate all files with a specified name, permissions setting, size, type, owner, or last access or modification date using the **find** command. This command is useful for locating seldom-used or excessively large files, or files owned by a particular user.

Maintaining Free Space in Filesystems

The **find** command has the form:

find *pathname option*

The *pathname* is the pathname of the directory that you want to search. The **find** command searches recursively, downward through all the directories under the named directory, for files that match the criteria specified by *option*. Some options also indicate certain actions for **find** to take on located files. See the **find(C)** manual page for complete details.

Table 4.2 describes some of the options to **find** that are useful for system administrators.

Table 4.2
Useful **find** Options

Option	Description
-atime <i>+number</i>	locates files that have not been accessed for the specified <i>number</i> of days.
-exec <i>cmd</i>	locates files that match the specified criteria and executes <i>cmd</i> on those files. The command argument (<i>{}</i>) is replaced by the current pathnames of the files that find located. An escaped semicolon (<i>\;</i>) must follow the <i>cmd {}</i> construction.
-group <i>gname</i>	searches for files that belong to the group <i>gname</i> . If <i>gname</i> is numeric and does not appear in <i>/etc/group</i> , it is interpreted as a group ID.
-name <i>file</i>	searches for files with the specified name.
-ok <i>cmd</i>	works like -exec except that it displays the generated command line in prompt form; <i>cmd</i> is executed only if the user enters y at the prompt.
-perm <i>onum</i>	locates all files with permissions that exactly match <i>onum</i> (the octal number used with chmod(C)).
-print	displays the locations of any files that find locates.
-size <i>+number</i>	searches for files larger than the specified number of blocks (512 bytes per block).
-type <i>x</i>	locates files of a specific type; for example, type d for directory or f for a file. See find(C) for an explanation of the different types.
-user <i>uname</i>	locates all files that belong to the user <i>uname</i> . If <i>uname</i> is numeric and does not appear in <i>/etc/passwd</i> , it is interpreted as a login ID.

Note

If you do not include the **-print** option, **find** does not display the list of files that match the search criteria.

Finding Temporary Files

To locate and display all files named *temp* recursively in the */usr* directory, use the following command:

```
find /usr -name temp -print | more
```

Finding Files of a Certain Size

You can use **find** to locate files of a specified size or type. For example, to locate and print a list of all the directories (**-type d**) greater than 3 blocks in size (**-size +3**) in all the directories (*/* and below), enter:

```
find / -type d -size +3 -print
```

Finding Files by Permissions

Using the **-perm** *onum* option to **find**, you can locate all files with permissions that exactly match *onum* (the octal number used with **chmod(C)**). For example, to locate and display all the files in the */u* directory that give all users read, write, and execute permissions (*onum* is **0777**), enter the following command:

```
find /u -perm 0777 -print
```

Maintaining Free Space in Filesystems

Executing Commands Based on find Output

Using the **-exec** option, you can execute a specific shell command on the files that **find** locates. The most common use of **-exec** is to locate a group of files and then remove them. For instance, when you remove a user, you can use **find** to locate all the files owned by that user, back them up, and then remove them from the system. To do this with **find**, enter:

```
find / -user edwarda -print | cpio -ovBc > /dev/rfd096
```

```
find / -user edwarda -exec rm {} \;
```

The first command locates all the files owned by user *edwarda* and copies the files to an archive. The second command locates the files and then removes them. For more information on copying files to an archive, see the **cpio(C)** manual page.

You can instruct **find** to display a prompt (the command line that **find** generates) before executing the shell command on each file by using the **-ok** option in place of **-exec**. When you use **-ok**, **find** prompts you with the generated command line:

```
<rm ... /u/edwarda/billboard >?
```

To execute the command (in this case, **rm**), enter **y**.

The command is not executed if you enter a character other than **y**, or if you press **<Return>**.

Another common use of **find** with the **-exec** option is to locate all the files that belong to a particular group and change them. For example, if a user changes groups, you can use **find** to locate and change all their files to the new group:

```
find / -user edwarda -exec chgrp moms {} \;
```

If you retire a user and you want to transfer ownership of their files to another user, use the following command:

```
find / -user edwarda -exec chown earnestc {} \;
```

Using this construction to execute a command on a large group of files can be very slow because the **-exec** option forks a separate process for each file in the list. A more efficient method for doing this is to use **xargs(C)** in place of **-exec**. The **xargs** command forks fewer processes to execute the command on the entire group of files.

The following example illustrates how to use the **xargs** construction with **find**:

```
find / -user edwarda -print | xargs chown earnestc
```

This command accomplishes the same thing as the previous example, only much more efficiently.

Note

If the syntax for the command that you want to execute with **xargs** deviates from the standard order (*command options arguments*), you must use **-exec**.

Locating core Files

4

One useful application of **find** is to use the **-name** option to locate core and temporary files for removal.

A core file contains a copy of a terminated program. A XENIX system sometimes creates such a file when a program causes an error from which it cannot recover. A temporary file contains data created as an intermediate step during execution of a program. This file may be left behind if a program contained an error or was prematurely stopped by the user. The name of a temporary file depends on the program that created it. In most cases, the user has no use for either core or temporary files, and they can be safely removed.

When searching for core or temporary files, it is a good idea to search for files that have not been accessed for a reasonable period of time. For example, to find all core files in the */usr* directory that have not been accessed for one week (**-atime +7**), enter:

```
find /usr -name core -atime +7 -print
```

Once you locate the core files, you can remove them easily using one of the following two commands:

```
find /usr -name core -atime +7 -exec rm {} \;
```

```
find /usr -name core -atime +7 -print | xargs rm
```

These commands find all the core files in the */usr* directory that have not been accessed in the last seven days and remove them.

Checking and Clearing Log Files

A XENIX system maintains a number of log files that contain information about system usage. When new information is generated, the system appends it to the appropriate log file, preserving the file's previous contents. Because some log files can rapidly become quite large, it is important to check the files periodically and, if necessary, clear them by deleting their contents. It may be necessary to retain the most recently appended information in some log files, such as */etc/ddate*. In these cases, trim the files by deleting the previous data and leaving the last entry.

Some log files only grow in special situations. For example, if you want to record all attempts to use the **su** command, and you set the **SULOG** option in */etc/default/su*, the log file */usr/adm/sulog* may build up rapidly.

Table 4.3 lists some of the log files that are most likely to need clearing or trimming. Your system may have different log files than those listed, depending on its configuration and the utilities and application software installed. You may also need to check the files more or less frequently than indicated, depending on system activity. Use the **find(C)** command with the **-size** flag to locate your system's unlisted large log files.

Table 4.3
Administrative Log Files

Logfile Path	Purpose	Checking Frequency
<i>/etc/wtmp</i>	Historical login record	weekly
<i>/usr/adm/pacct</i>	Process accounting log file	weekly
<i>/usr/adm/messages</i>	System messages log file	weekly
<i>/etc/ddate</i>	Records date of each backup	monthly or yearly
<i>/usr/spool/micnet/remote/(name)/LOG</i>	Records of network activity with named remote machine	monthly
<i>/usr/spool/uucp/LOGFILE</i>	Records of UUCP job requests, file transfers, system status	monthly
<i>/usr/spool/uucp/.Log/Old/*</i>	Old UUCP log files stored by uudemon.clean	monthly
<i>/usr/spool/lp/log</i>	Record of print requests	weekly

Automatic maintenance of these files is described later in this section.

To clear a log file and retain file permissions:

1. Copy the file to a new filename. For example:

```
cp /usr/adm/messages /usr/adm/messages.old
```

2. You can clear the file by using one of the following commands:

Bourne or Korn shell:

```
>/usr/adm/messages
```

C shell:

```
cat /dev/null > /usr/adm/messages
```

To monitor new information that is currently being appended to system log files, use the **tail(C)** command as follows:

```
tail -f /usr/adm/messages
```

With the **-f** (follow) option, **tail** prints the last ten lines of the file, followed by any lines that are appended to the file between the time **tail** is initiated and killed with the (Del) key.

Automatic Cleaning of Log Files

It is also possible to automate the clearing of log files. Here is an example using several of the log files mentioned previously. This and other files can be cleaned out periodically using, for instance, the **cron(C)** facility.

For example, when added to the */usr/spool/cron/crontabs/root* file, the following line executes a script called */usr/bin/cleanlog* each Friday at 11 P.M.:

```
0 23 * * 5 /usr/bin/cleanlog
```

Figure 4-2 contains a usable *cleanlog* script. You can copy it or modify it to suit the needs of your system. The script should have permissions of 700.

```
:
#
# cleanlog script
#
# Cleans up the following files:
#   - /usr/adm/messages
#   - /etc/wtmp
#   - /usr/spool/lp/log
#   - /usr/lib/cron/log
#
# IFS="{space}{tab}{newline}"
IFS="
"
PATH=/bin:/usr/bin
export IFS PATH

set -- `cat /etc/systemid`
LOCSYS=${1:-$2}

#
# Clean up /usr/adm/messages, saving the last 100 lines
#
cd /usr/adm

tail -100 messages > messages.new
cat messages.new > messages
rm -f messages.new

#
# Clean up /etc/wtmp, saving the old one as /etc/wtmp-
#
mv /etc/wtmp /etc/wtmp-
> /etc/wtmp
chown bin /etc/wtmp
chgrp bin /etc/wtmp
chmod 644 /etc/wtmp

#
# Clean up /usr/spool/lp/log, saving the last 100 lines
#
tail -100 /usr/spool/lp/log > /usr/spool/lp/log.new
cat /usr/spool/lp/log.new > /usr/spool/lp/log
rm -f /usr/spool/lp/log.new

#
# Clean up /usr/lib/cron/log, saving the last 100 lines
#
if [ -f /usr/lib/cron/log ]
then
    tail -100 /usr/lib/cron/log > /usr/lib/cron/log.new
    cat /usr/lib/cron/log.new > /usr/lib/cron/log
    rm -f /usr/lib/cron/log.new
fi

exit 0
```

Figure 4-2 Example Cleaning Script: cleanlog

Maintaining Efficient Filesystem Organization

There are two aspects of filesystem usage that degrade the efficiency of filesystems:

- Disk fragmentation (scattering of available disk space) from constant use and reuse of filesystem blocks.
- Directories growing too large, thus increasing search time for files.

This section explains how to avoid these problems and maintain efficient filesystems.

4

Disk Fragmentation

If your system has been in use for some time, the constant creation and removal of files creates a situation called *disk fragmentation*. This means that the files in the filesystem are written in small pieces on the hard disk. The additional access time needed to reassemble these fragments is significant. This can be reduced by first making a complete backup of all the files in the filesystem and then removing all the files from the hard disk and restoring them from the backup. To do this, see “Reducing Disk Fragmentation” in the “Troubleshooting Your System” chapter in this guide for instructions.

Because the files are completely rewritten on the disk during the restore, each file is written in one piece and fragmentation (and thus access time) is reduced. It is a good idea to perform this action about once a year on a heavily used system and less often on a lightly used system. Be certain that you have complete, accurate, and readable backups before you begin or your files may be lost.

Large Directories

It is wise to avoid directories that are larger than necessary. You should be aware of several size limitations. A directory that contains entries for up to 62 files (plus the required `.` and `..` entries) fits in a single disk block and can be searched very efficiently. A directory can have up to 638 entries and still be viable, as long as it is used only for data storage; anything larger is usually a problem if it is used as a working directory. It is especially important to keep login directories small, preferably one block at most.

If large numbers of files are moved in and out of a directory, as in the case of a spool directory, it can also become oversized. This is because, as a rule, directories never shrink. Even if you delete files until you have less than 62 in your login directory, or less than 638 in a spool directory or other data collection directory, system searches of the directory slow down. This is because the inode numbers for the deleted entries still exist.

To see if a directory is too large, even if it does not seem to have over 62 files in it, go into the directory and use the command `hd . | wc -l`. This tells you the number of lines (inodes) in the inode listing.

The solution to the large directory problem is twofold. Educate users to keep their login directories small, and use `cpio` to back up and restore any directory that has an excess of inode numbers left over from deleted or moved files.

Limiting Login Directory Size

To limit the size of the login directory, each user should create a hierarchy of subdirectories that reflects the use or purpose of the contained files. This can make finding files easier for the user. Emphasize the importance of setting up a simple, clear hierarchy, with only five to 10 directories at each level, so files can be added in the future without having to change the hierarchy or move files around.

Removing Excess Inode Numbers

To remove “shadow” inode numbers (numbers left when files are deleted or moved to subdirectories), use `cpio` to back up and restore an oversized directory. Then use `hd` to list the files in the directory and see if there are any that do not show in a standard listing. These “shadow” files appear at the end of the list, and have null inode numbers; 00 and 00 appear in the first two columns of hex digits. Use the following procedure to remove these empty file slots from the directory. Note that the same procedure can be applied to a spool directory.

Maintaining Efficient Filesystem Organization

1. Move to the directory above the oversized one and create a backup directory. For example, if a user named “tracy” has an oversized home directory, enter:

```
mkdir tracy.old
```

2. Move to the oversized directory and back it up:

```
find . -print | cpio -pdlm ../tracy.old
```

This command copies the directory and all subdirectories, linking files instead of copying them where possible. Confirm that the ownership, group, and permissions of the new directory and files match the original.

3. Use **l** and **hd** to check the backup directory to see that all the files are there and all the shadow inode numbers are gone.
4. Move to the directory above, then delete the oversized directory:

```
rm -rf tracy
```

5. Rename the backup directory to replace the oversized one:

```
mv tracy.old tracy
```

The directory appears the same to the user, but searches of it are quicker. This can improve system performance, especially if the procedure is used on numerous working directories.

Adding Disk Space and Filesystems

If free space is chronically low, it may be to your advantage to expand the system's storage capacity by adding a second hard disk as described in the "Adding Hard Disks" chapter. Once the new disk is mounted, you can use the free space in the new filesystem for your work, or even copy user or system directories to it.

Suppose free space is low on your primary hard disk because one filesystem on it is full. If there are others with free space, or there is unused space on the hard disk, you can change the layout on the primary disk, but this is not as simple as adding a second hard disk. It is always best to plan the layout of your hard disk in advance as described in the *XENIX Installation Guide*.

A chronic shortage of space usually results from having more users on the system than the current hard disk can reasonably handle, or having too many directories or files. In either case, creating a new filesystem on a new hard disk allows some of the users and directories to be transferred from the primary hard disk, freeing a significant amount of space on the existing filesystem and improving system operation.

If you decide to change the number of filesystems on your primary hard disk or to reappportion the disk space among the filesystems, then you must back up your system and reinstall it as described in the *XENIX Installation Guide*. During the installation process, use manual layout control to reappportion your disk space as desired.

Filesystem Integrity

The most important job of the operating system is to maintain the integrity of filesystem data. Actual loss of data is a rare occurrence; XENIX filesystems are very resistant to corruption. This is because a certain amount of redundancy exists in special structures that are invisible to the user. It is these structures that ensure filesystem integrity. For example, when the system suffers a power outage, very little information is lost. Any damage usually affects one or two files, making them inaccessible. In almost all cases, the operating system can repair any damage done to files. In very rare cases, damage causes the entire filesystem to become inaccessible.

The operating system uses the **fsck** (for “filesystem check”) program to repair damaged filesystems. The **fsck** program checks the consistency of filesystems. In cases where the contents of a file are lost (rare), the only way to restore lost data is from filesystem backups. **fsck** is run automatically after an abnormal shutdown on the root filesystem at boot time. The **fsck** status messages look like this:

4

```
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Pathnames
** Phase 3 - Connectivity
** Phase 4 - Reference Counts
** Phase 5 - Check Free List
```

If the system terminated abnormally (power outage), you see other messages that may seem alarming:

```
FREE INODE COUNT WRONG IN SUPERBLK (FIX?)
```

In fact, this kind of message is routine when a system was not shut down properly, and you only have to enter **y** and **fsck** continues its recovery work. This could be done without the system administrator’s intervention, but it is generally better to know what is happening to a filesystem after a problem has occurred.

To discuss the idea of filesystem integrity and how **fsck** functions, it is useful to explain the structure that underlies the simple idea of files, directories, and filesystems. Although it is not vital to understand the principles of file storage, it is helpful to know what the messages like the one above refer to so they will seem less mysterious.

Filesystem Integrity

This section describes some of the most basic principles of XENIX operating systems. The section “Repairing a Filesystem with **fsck**” explains the simple mechanics of using the **fsck** command. The subsections that follow explain the filesystem structures that **fsck** manipulates.

How XENIX Systems Maintain Files

Each filesystem contains special structures that allow the operating system to access and maintain the files and data stored on the filesystem. It is the disruption and repair of these structures that we are concerned with.

The structure of a filesystem is based on the way that hard disks store data. Although the hard disk contains all the data used by the system, it is not stored in neat little locations that correspond to individual files. It is unlikely that you could point to a spot on a hard disk and truthfully say: “My file is stored right there on this part of the disk.” In fact, the data is probably scattered across the disk. The operating system uses a sophisticated addressing scheme to access each of the pieces that a file is broken into and to present the data to the user as a unit.

The data is spread around because the operating system does not really deal with files, but rather with units of data. For example, assume you created a file and it is actually stored on one part of the disk. Then, suppose you edit that file and delete a few sentences here and there. This means that you are now using a little bit less disk space than when you started. This space amounts to a series of gaps in the area where your file was stored. Disk space is a precious commodity and is not wasted. So, those small amounts of memory are allocated to other files. Picture this process on a scale of hundreds of files with scores of users and you have an idea of how files are maintained.

Note

Files consume a minimum of one data block.

How XENIX Systems Maintain Filesystems

A filesystem contains files and directories that are represented by special structures called “inodes” and “data blocks.” These structures make it possible for the operating system to create and keep track of filesystems.

Data blocks A block is a 1024-byte unit of data stored on the disk. A data block can contain either directory entries or file data. A directory entry consists of an inode number and a filename.

Note

Certain utilities differ in the block size reported. Utilities concerned with the structure of the filesystem, including **fdisk**(ADM), **divvy**(ADM), and **fsck**(ADM) use 1024-byte blocks. Most user-level utilities report in 512-byte blocks, including **df**(C) and **du**(C).

4

Inodes An inode can be thought of as a card from a library card catalog. Each inode contains information about a file, just as a card contains information about a book, including its location, its size, the type of file, and the number of directory entries linked to the file. One important point to remember is that an inode does not contain the name of a file; directories contain the actual names. An inode contains the locations of all the data that make up a file so the operating system can collect it all when needed.

Blocks are not just stored on the hard disk. To minimize seeking data on the hard disk, recently used data blocks are held in a cache of special memory structures called buffers. These structures make the operating system more efficient. When enough data is accumulated to write out one or more full disk blocks, the buffer is “flushed” by writing its information to the disk. A minor amount of information is often lost when an outage occurs because recently changed data that has not yet been written to the disk.

With a hard disk filled with data, inodes, directories, files, and blocks cached in memory, how does the operating system keep track of them? The answer is that all these structures maintain sufficient connectivity between files and directories to allow severed connections to be reconstructed.

Filesystem Integrity

One special data block, the “super” block, contains overall information about the filesystem, rather than where a particular piece of a file is located. The super block contains the information necessary to mount a filesystem and access its data. It contains the size of the filesystem, the number of free inodes, and information about free space available.

Information is read from the disk version of the super block when the filesystem is mounted and is maintained and modified in memory as activity takes place on the system. The information is written back to the disk at regular intervals by the **update** command, which is normally executed by the `/etc/rc` scripts when the system is brought up. The **update** command calls the **sync(C)** command every 30 seconds, which forces the memory version of the super block and buffers to be written to disk. If the system crashes and the information stored on the disk is not reasonably up-to-date, the filesystem might be corrupted.

Causes of Filesystem Corruption

Corruption can affect all the structures mentioned in this section. This means that the data or the structures used to locate data can be damaged. This can occur for several reasons:

- | | |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hardware Failure | Hardware failures are rare. The best way of dealing with them is to be sure that recommended diagnostic and maintenance procedures are followed conscientiously. |
| Program Interrupts | It is possible that errors that cause a program to fail might result in the loss of some data. It is not easy to generalize about this because the range of possibilities is so large. |
| Human Error | While it may be painful to admit it, probably the greatest cause of filesystem corruption falls under this heading. There are rules that should be followed when managing filesystems. |

Rules for Checking Filesystems

When checking and repairing filesystems, keep in mind the following rules:

- To make sure users cannot access the filesystem you are planning to unmount, bring the system down to single-user (maintenance) mode before unmounting it.
- Unmount the filesystem with **umount(ADM)** before checking and repairing it.
- Unmount the filesystem, such as one on a floppy disk, before physically removing it.
- If you plan to make the filesystem available, remember to remount the filesystem after checking and repairing it.

Regular filesystem backups represent the best assurance of continued filesystem integrity.

4

Repairing Filesystems with fsck

To check and repair filesystems, use **fsck(ADM)**. The **fsck** command examines the various structures on the disk and attempts to reconcile them. Where possible, **fsck** reestablishes connections and resolves references; it “cleans” the filesystem.

Before repairing a filesystem with **fsck**, shut the system down and bring it up in single-user (maintenance) mode. To do this without completely shutting down the system, use the **su** argument to **shutdown**:

```
/etc/shutdown 10 su
```

The first argument indicates the number of minutes before system shutdown.

Note

The **shutdown** command calls **sync** automatically

Filesystem Integrity

For more information on shutting the system down, see **shutdown**(ADM).

Once the system is in single-user mode, unmount the filesystem with **umount**(ADM). Check and repair the filesystem using the following command:

```
fsck filesystem
```

△ **sysadmsh** users select: Filesystems→Check

where *filesystem* is the name of the special file corresponding to the device name of the filesystem. For example, the */dev/u* device file corresponds to the */u* filesystem.

To check the root filesystem, use the **-rr** argument to **fsck**:

```
fsck -rr /dev/root
```

The system must be in single-user mode before running **fsck -rr** on the root filesystem. This command automatically remounts the root filesystem after checking it. For example, if you bring your system back up in single-user mode after a power failure, use the **fsck** command to check the */u* filesystem before going into multiuser mode. To do this, enter the following command:

```
fsck /dev/u
```

The **fsck** program checks the filesystem and reports on its progress with the following messages:

```
** Phase 1 - Check Blocks and Sizes  
** Phase 2 - Pathnames  
** Phase 3 - Connectivity  
** Phase 4 - Reference Counts  
** Phase 5 - Check Free List
```

If a damaged file is found during any one of these phases, **fsck** asks if it should be repaired or salvaged. Enter **y** to repair a damaged file. You should always allow the system to repair damaged files even if you have copies of the files elsewhere or intend to delete the damaged files.

Note that the **fsck** command deletes any file that it considers too damaged to be repaired. You can elect for **fsck** to make the repair or not. You might choose to have **fsck** ignore an inconsistency because the problem is so severe that either you want to fix it yourself using the **fsdb**(ADM) utility, or you plan to restore your system from backups. If you cannot use **fsdb**, you must allow **fsck** to resolve the inconsistencies or the filesystem may not be usable. (**fsdb** is described in “Repairing a Filesystem When fsck Stops at Size Check” in the “Troubleshooting Your System” chapter of this guide.)

You may need to run **fsck** several times before the entire filesystem is clean. For a complete list of error messages, see the **fsck**(ADM) manual page.

Summary of fsck Phases

The **fsck** program scans and examines each of the structures mentioned earlier. Each phase compares components and checks that these components agree with each other.

4

Phase 1 checks the blocks and sizes. The **fsck** program reads the inode list to determine the sizes and locates the blocks used by each file. Inodes are checked for inode type, zero link counts, inode sizes, and bad or duplicate blocks. (Bad blocks are block values outside the boundaries of a filesystem.) When **fsck** asks whether or not to clear an inode, this means to zero out the bad information in the node. This removes the file or directory that was associated with it. A duplicate block means that two inodes point to the same block on the disk. The **fsck** command attempts to find the original inode along with the duplicate for correction in phase 2.

Phase 2 checks the pathnames. Files removed in phase 1 must then have their directory entries removed. Phase 2 cleans up error conditions caused by improper inode status, out-of-range inode pointers, and directories that point to bad inodes as described earlier. For files with duplicate blocks found in phase 1, **fsck** wants to remove both files (this is one of the few areas where system administrator intervention is useful).

Phase 3 checks for connectivity. Phase 2 removed directories that do not point to valid files. Phase 3 reconnects files that were severed from the directory structure. Any files that are unreferenced but valid are placed in a special directory called *lost+found*. Because the directory was severed, the name of the file is lost and a number is assigned to the file in *lost+found*.

Filesystem Integrity

Phase 4 checks the reference counts. The **fsck** command checks the link count of each entry that survives phases 2 and 3. In some cases, files that were not pointed to under the directory structure, but still have an inode, can be relinked to the filesystem in *lost+found*.

Phase 5 checks the free list. The **fsck** command examines the free-block list maintained by the filesystem and resolves the missing or unallocated blocks allocated or removed earlier. When an inconsistency is detected, **fsck** prompts to rebuild it.

Phase 6 salvages the free list. If specified in phase 5, the system reconstructs a free block list from the altered filesystem.

Automatic Filesystem Check

The operating system sometimes requests a check of the filesystem when you first start it. This usually occurs after an improper shutdown (for example, after a power loss). The filesystem check repairs any files disrupted during the shutdown.

Chapter 5

Maintaining System Security

Introduction 5-1

General Security Practices 5-3

Physical Security 5-3

Access Security 5-3

User Passwords 5-5

Defining Password Length 5-5

Changing a User's Password 5-6

Maintaining File Permissions 5-7

Displaying Permissions 5-7

Changing Permissions 5-10

Changing the File Creation Mask 5-11

Protecting Special Files 5-12

Managing File Ownership 5-14

Changing User Ownership 5-14

Changing Group Ownership 5-15

Forcing a New Password 5-16

Enabling Password Aging 5-16

Setting the Minimum and Maximum for One User 5-17

Disabling Password Aging 5-17

Forcing a New Password Immediately 5-17

Preventing a Password Change 5-18

Displaying Password Aging Information 5-18

Logging Out Idle Users 5-19

Adding Dial-in Password Protection 5-20

Permitting Users to Mount Filesystems 5-22

Monitoring Login Sessions 5-24

Monitoring su Attempts 5-25

Using XENIX Accounting Features 5-26

Starting and Stopping Process Accounting 5-26
Displaying Accounting Information 5-27

Setting up a Restricted Shell 5-30
Creating a Restricted Korn Shell Account 5-30
Creating a Restricted Bourne Shell Account 5-31
Special Uses for the Restricted Shell 5-32

Introduction

This chapter outlines XENIX facilities and strategies for maintaining system security. To be effective, your system security strategies must:

- prevent theft of data on a system and theft of the system itself
- eliminate tampering with data on a system and tampering with the system itself

Both aspects of security are important. You can prevent thefts and eliminate tampering through:

- user education
- administrator education
- management education
- hardware confinement
- software support

For system security to be reliable, users must not do such things as revealing or writing down their passwords. System administrators must follow the same rules. Management must support appropriate efforts to prevent theft and tampering.

Notice that hardware and software are at the bottom of the list; the best software in the world is essentially useless if users give away their passwords. (the “Logging In” chapter of the *XENIX Tutorial* discusses basic security from a user standpoint in the section “Keeping Your Account Secure.”)

This chapter discusses the following topics:

- establishing physical security and access security (passwords)
- maintaining file and directory permissions using the “file creation mask” and the **umask(C)** command
- managing user and group ownership of files
- forcing a user to choose a new password with the **pwdadmin(ADM)** command

Introduction

- logging out idle users with **idleout**(ADM)
- adding dial-in passwords
- allowing a user to mount a filesystem with the **mnt**(C) command
- setting up XENIX process accounting
- creating restricted accounts

General Security Practices

Every system, no matter what its size, should have some form of protection from unauthorized access to the computer, disks, and system files. The following sections suggest ways for a system administrator to protect the system.

Physical Security

You can protect the physical components of the computer, especially system disks, by taking these precautions:

- Keep your hardware secure and locked when not in use.
- Organize and lock up all floppy disks when not in use. Do not store disks with the computer itself.
- Keep disks away from magnetic surfaces, direct sunlight, and severe changes in temperature.
- Write labels on disks with felt-tipped pens, not ballpoint pens.
- Make backup copies of all floppy disks. See the “Using Floppy Disks and Tape Drives” chapter of the *System Administrator’s Guide* for instructions on making backups.
- Keep an accurate and complete system log (in hardcopy, not on the computer).

5

Access Security

You can protect the system from access by unauthorized individuals by taking these precautions:

- Train users to log out of their accounts before leaving their terminals.
- Train users to use the **lock(C)** utility if they leave their terminals, even for a short time.

General Security Practices

- Force logouts on idle terminals with the **idleout**(ADM) command. For more information on **idleout**, see the section “Logging Out Idle Users” later in this chapter.
- Discourage users from choosing passwords that are easy to guess. For guidelines on choosing secure passwords, see the section “User Passwords” later in this chapter.
- Require users to keep their passwords secret at all times. Users should never write their passwords down, send them over electronic mail, or tell them to anyone else in person or by phone.
- Force users to change their passwords regularly with the **pwadmin** command. For more information on **pwadmin**, see the section “Forcing a New Password” later in this chapter.
- Keep the superuser password secret from anyone whose job does not require superuser access, and change the password regularly.
- Restrict access to the system console.
- Restrict access to the XENIX distribution floppies.
- Maintain appropriate permissions on the directories */etc*, */bin*, */usr*, and */dev*. All users should be able to read and execute these directories, but only the super user should have write permission. Also maintain the permissions on system utilities that are reserved for the super user, such as **mkuser**(ADM), **shutdown**(ADM), **haltsys**(ADM), **restore**(ADM), **setuid**(S), and others. Incorrect permissions on these files and directories are a serious lapse in system security. XENIX is distributed with the correct permissions set on all files and directories. Retain these original permissions so that the files and directories do not present a system security problem.

User Passwords

For user passwords to create an effective defense against intrusion into your system, your users must choose passwords that others cannot easily guess. Give your users guidelines for choosing passwords, such as the following:

- Include uppercase and lowercase letters, numerals, escape characters, and punctuation marks in passwords.
- Make passwords memorable enough that users do not need to write them down.
- Do not use people or pet names, place names, or any word from the file */usr/dict/words* as passwords.
- Do not alter normal words in any standard way to create passwords; for example, do not use words spelled backwards or words formed by a well-known code such as “pig latin.”
- Do not simply add an extra character or numeral to normal words to create passwords.
- Make passwords at least six characters long.

5

Words from different languages or strings of short words can make good password choices. The string “4wOmbAt?” has the following features of a good password: it is unusual, it contains letters, numbers, and punctuation, it has capital letters within the word, and it is long enough to prevent easy guessing or decoding using a password-testing program.

Defining Password Length

You can define the minimum password length by setting the **PASSLENGTH** variable in the */etc/default/passwd* file. **PASSLENGTH** specifies the minimum password length when users change their passwords with **passwd(C)**. The default value is 0. To ensure that user passwords are at least six characters long, edit the file */etc/default/passwd* and change the **PASSLENGTH** line to read:

```
PASSLENGTH=6
```

Changing a User's Password

Normally, ordinary users can change the password of their own accounts with the **passwd** command (see the "Logging In" chapter of the *XENIX Tutorial*). Sometimes, however, the superuser needs to change the password for a user. For example, if a user forgets a password, and cannot get into the account to change it, the superuser must change the password. The superuser can change the password of any user (including the superuser account) with the **passwd** command.

To change a password, follow these steps:

1. Log in as the superuser.
2. Enter the following command, where *login_name* is the user's login name:

```
passwd login_name
```

You see the prompt:

```
New password:
```

3. Enter the new password and press **<Return>**. The command does not display the password as you type it, so type carefully. The command then prompts you to enter the password again:

```
Retype new password:
```

Enter the password again and press **<Return>**.

Maintaining File Permissions

Permissions control access to all the files and directories in a XENIX system. In XENIX, ordinary users can access only those files and directories for which they have permission.

There are three different levels of permissions: user, group, and other. User permissions apply to the owner of the file, group permissions apply to users who have the same group ID as the owner, and other (public) permissions apply to all other users.

Displaying Permissions

Any user can display the permission settings for all the files in a directory with the list directory command, `l(C)`. This command lists the permissions along with the name of the file owner, the size (in bytes), and the date and time the file was last modified. The command display has the following format:

```
-rw-rw---- 1 johnd group 11515 Nov 17 14:21 file1
```

The permissions appear as a sequence of 10 characters at the beginning of the display. The sequence has four fields. The first field is the “type” field and contains a single character. The other fields show permissions for user, group, and other; they contain three characters each.

Maintaining File Permissions

Table 5.1 lists the characters in the type field.

Table 5.1
File Type Designators

Character	Meaning
d	directory
-	ordinary file
b	block device I/O file
c	character device I/O file
p	named pipe
s	semaphore
m	shared data (memory) file

Table 5.2 lists the characters in the fields for user, group, and other.

Table 5.2
Permission Designators

Character	Meaning
r	read permission On a file, read permission means that you can copy or display the file. On a directory, read permission means that you can list the files in that directory.
w	write permission On a file, write permission means that you can edit and modify the file. On a directory, write permission means that you can create or remove any file or subdirectory within that directory.
x	execute permission On a file, execute permission means that you can invoke the file as you would a program. On a directory, execute permission means that you can enter that directory with the change directory command, <code>cd(C)</code> .
-	no permissions on the file or directory

Maintaining File Permissions

Below are examples of how permissions can allow and restrict access to files in an office setting where each login group consists of the employees in one department:

- Everyone in the company needs to execute a file but only the owner (probably the super user) should read and edit it. Set the file permissions to:

`-rwx--x--x`

- A file contains information that everyone in the department needs to read. Nobody from other departments should see the information, and only the owner should edit the file. Set the file permissions to:

`-rw-r-----`

- A file contains information that everyone in the department needs to read and edit. Nobody from other departments should have access to the information. Set the file permissions to:

`-rw-rw----`

- A file contains private information that only the owner should see or change. Set the file permissions to:

`-rw-----`

- Everyone in the company needs to enter a directory and list the files, but only one department should create and remove files. Set the directory permissions to:

`drwxrwxr-x`

When you create a file, the XENIX system assigns the following permissions by default:

`-rw-r--r--`

These permissions allow everyone to read the file, but only the user can edit it.

Maintaining File Permissions

When you create a directory, the system assigns the following permissions by default:

```
drwxr-xr-x
```

These permissions allow everyone to enter and read the directory, but only the user can create and remove files and subdirectories within it.

Changing Permissions

Any user who owns a file or a directory can change its permissions with the change mode command, **chmod(C)**. The **chmod** command has the following syntax, where *file* is the name of one or more files or directories that you want to affect:

```
chmod mode file ...
```

△ **sysadmsh** users select: Dirs/File→Modify→Permissions

To specify the *mode*, first indicate whose permissions you wish to change:

```
u user (owner)    g group    o other (everyone else)
```

Next, indicate whether you want to add or remove permissions:

```
+ add    - remove
```

Finally, specify which permissions you wish to change:

```
r read    w write    x execute
```

For example, to change the permissions of the file *iris* from `-rw-rw-r--` to `-rwxrw-r--`, add execute permission for the user as follows:

```
chmod u+x iris
```

To change the permissions of the directory *bulbs* from `drwxrw-rw-` to `drwx-----`, remove read and write permission for group and other as follows:

```
chmod go-rw bulbs
```

If you specify more than one file, separate the filenames with spaces. For example, to change the permissions of the files *crocus* and *tulip* from `-rw-r--r--` to `-rw-rw-r--`, add write permission for group as follows:

```
chmod g+w crocus tulip
```

To change one permission for one set of people and a different permission for another set of people, separate the modes with commas. For example, to change the permissions of the file *jonquil* from `-rw-r--r--` to `-rw-rw----`, add write permission for group and remove read permission for other as follows:

```
chmod g+w,o-r jonquil
```

After using **chmod**, check the results with the **l** command. Use **chmod** again to correct any mistakes.

Changing the File Creation Mask

The file creation mask is a special number, kept by the system, that defines the permissions given to every file and directory created by a user. Initially, the mask has the value 022, which assigns each ordinary file the permissions `-rw-r--r--` and each directory the permissions `drwxr-xr-x`.

5

You can change the mask and the default permissions that the system assigns your files and directories by using the **umask** command. You can execute **umask** as a shell command, but typically you establish individual **umask** defaults for each user by inserting the **umask** command into each user's *.profile* file (for Bourne shell users) or *.login* file (for C shell users).

The **umask** command has the form:

```
umask value
```

Maintaining File Permissions

The *value* is a three-digit number. The three digits represent permissions for user, group, and other, respectively. Each digit defines which permission the system assigns, as shown Table 5.3.

Table 5.3
umask Digits

Digit	Ordinary File Permissions	Directory Permissions
0	read and write	read, write, and execute
1	read and write	read and write
2	read	read and execute
3	read	read
4	write	write and execute
5	write	write
6	no permissions	execute
7	no permissions	no permissions

For example, the following command sets the file creation mask so that all files and directories initially have read and write permission for the user, and no permissions for all others:

```
umask 177
```

Protecting Special Files

You can prevent ordinary users from gaining direct access to the data and program files on the system's hard and floppy disks by protecting the system's special files. The XENIX special files, in the */dev* directory, are used primarily by the system to transfer data to and from the computer's hard and floppy disks, as well as to other devices. However, ordinary users can also use the special files to gain direct access to these devices.

Because direct access bypasses the system's normal protection mechanisms and allows ordinary users to examine and change all files in the system, you must protect the special files to ensure system security.

Initially, the XENIX special files have the correct permissions. If you find that the permissions are wrong, log in as the super user and use the **chmod** command to set appropriate permissions. For example, to prevent any access by ordinary users, set the permissions of special files such as */dev/mem*, */dev/kmem*, */dev/root*, and */dev/usr* to read and write access for the file owner only.

Note

Do not change the permissions for the `/dev/tty` files. These files must remain writable to allow normal logins.

Managing File Ownership

Whenever a user creates a file, the system assigns user ownership of that file to that user. User ownership allows the creator to access the file according to the user permissions. The system also assigns a group ownership to the file, defining which group can access the file according to the group permissions. The file's group is the same as the creator's group.

The system allows only one user and one group to own a file at any one time. (You can display this owner and group with the `l` command.) However, you can change the ownership of a file by using the **chown** and **chgrp** commands.

Changing User Ownership

Any user who owns a file can change the user ownership of that file with the **chown** command. The super user can change the ownership of any file with **chown**. The command has the following form, where *login_name* is the name of the new user, and *file* is the name of the file or directory that you want to change:

```
chown login_name file
```

△ **sysadmsh** users select: Files/Dirs→Modify→Ownership

For example, the following command changes the current owner of the file *projects.june* to johnd:

```
chown johnd projects.june
```

The **chown** command is especially useful when you change the user ID of a user account; for example, when one user takes over all of the projects formerly managed by a previous user.

Changing Group Ownership

The owner of a file can change the group ownership with the **chgrp** command. The super user can change the group ownership of any file with **chgrp**. The command has the following form, where *group_name* is the name of a group listed in the */etc/group* file and *file* is the name of the file you wish to change:

```
chgrp group_name file
```

△ **sysadmsh** users select: Files/Dirs→Modify→Ownership

For example, the following command changes the group ownership of the file *projects.june* to the group named *mgmt*:

```
chgrp mgmt projects.june
```

The **chgrp** command is especially useful when you change the user's log-in group; for example, when the user transfers to a new department.

Forcing a New Password

Because the security of any account depends on its password, system users and system administrators should keep the password as secret as possible. “Password aging” increases password security by forcing users to change their passwords on a regular basis. The **pwadmin**(ADM) command controls password aging.

When you enable password aging on an account, **pwadmin** dates the password for that account and requires the user to choose a new password when a specified maximum number of weeks have passed. Password aging can also require that the user wait a minimum number of weeks before changing the password again. The maximum limit reduces the chance for an unauthorized person to learn a password. The minimum limit prevents a user from choosing a new password and then immediately changing back to the old one.

To use **pwadmin**, you must be logged in as the super user.

Enabling Password Aging

The **-a** option enables password aging on an account. For example, to enable password aging on the account *maryr*, enter:

```
pwadmin -a maryr
```

The variables **MINWEEKS** and **MAXWEEKS** in the file */etc/default/passwd* define the minimum and maximum length of time the user can keep a password. For example, you might set **MINWEEKS=4** and **MAXWEEKS=6**. Then, once you enable password aging for an account, the user of the account cannot keep a password for longer than six weeks, and must use a new password for at least four weeks before changing it again.

You can edit */etc/default/passwd* and change **MINWEEKS** and **MAXWEEKS** to any values that suit your needs, with the following restrictions:

- **MINWEEKS** must be at least 0 and no more than 63
- **MAXWEEKS** must be at least 1 and no more than 63
- **MINWEEKS** must not be greater than **MAXWEEKS**

Setting the Minimum and Maximum for One User

The **-min ... -max** option sets minimum and maximum values for a user that are different from the **MINWEEKS** and **MAXWEEKS** values. For example, if **MINWEEKS** is 4 and **MAXWEEKS** is 6, but you want to allow the account *agathac* to keep passwords for as long as 10 weeks, enter:

```
pwadmin -min 4 -max 10 agathac
```

Note that you must specify both the minimum and the maximum with this command, even though the specified minimum is the same as **MINWEEKS**. You can name any minimum and maximum values, subject to the same restrictions as the values for **MINWEEKS** and **MAXWEEKS**.

Disabling Password Aging

The **-n** option removes the current minimum and maximum password age for an account. For example, to disable password aging for the account *dorothys*, enter:

```
pwadmin -n dorothys
```

The system no longer prompts the user for changes.

Forcing a New Password Immediately

The **-f** option forces a user to change passwords immediately. For example, to force a change on the account *maryr*, enter:

```
pwadmin -f maryr
```

The system prompts the user to supply a new password the next time the user logs in. Use this command if you believe someone has learned an account's password.

Forcing a New Password

Preventing a Password Change

The **-c** option prevents the user of an account from changing the account's password. For example, to prevent a change in the password for account *admin*, enter:

```
pwadmin -c admin
```

This command gives you control of password changes on an account. One situation in which you might want such control is when several users use a group or administrative account. If one user changes the password, the others cannot log in.

Displaying Password Aging Information

The **-d** option displays the current minimum and maximum password age for an account. For example, to see password aging information on the account *dorothys*, enter:

```
pwadmin -d dorothys
```

You see a display similar to the following:

```
Minimum weeks: 4  
Maximum weeks: 6
```

This command does not change the current values.

Logging Out Idle Users

Finding a user logged into the system who has not entered any command or information for a long time can indicate that the user left the terminal and forgot to log out. The **idleout**(ADM) command monitors line activity and logs out any user whose terminal remains idle longer than a specified period of time. You must be logged in as the super user to run **idleout**.

To begin monitoring line activity for the system, enter:

idleout

△ **sysadmsh** users select: System→Configure→Autologout

The **IDLETIME** variable in the */etc/default/idleout* file determines how long a user's terminal can remain idle before the system logs the user out. If the value of **IDLETIME** contains a colon (:), **idleout** calculates the time in hours; otherwise, **idleout** calculates the time in minutes.

You can also specify the acceptable idle time on the command line in the one of the following forms:

idleout *minutes*

or

idleout *hours:minutes*

If you want **idleout** to run automatically when you reboot your system, enter the command name, **idleout**, on a line by itself in the file */etc/rc.d/8/userdef*.

Note

Do not use **idleout** on a system running VP/ix™. VP/ix does not notify XENIX when a key is pressed, so **idleout** will log out a VP/ix user even though the user is active.

Adding Dial-in Password Protection

XENIX lets you define special dial-in passwords on selected tty lines, requiring selected classes of users to enter dial-in passwords before they can access the system. You can store logging information, including the last time of connection, for later use.

Define specific dial-in lines that require passwords in the file */etc/dialups*. The format is one tty device name per line, for example:

```
/dev/tty1A  
/dev/tty5C
```

Create a file named */etc/d_passwd* to contain the actual dial-in passwords. The */etc/d_passwd* file uses the same format as the */etc/passwd* file, except that only the first two fields (user shell and encrypted password) have meaning. The remaining fields, if present, are ignored. A colon (:) separates each field.

A sample */etc/d_passwd* file might be:

```
*:encrypt_pass1:Default dial-in password  
/usr/lib/uucp/uucico::UUCP dial-in password (none)  
/bin/rsh:encrypt_pass2:Restricted shell dial-in password
```

The “*” in the first field of the sample’s first line indicates that the line contains the default dial-in password. Otherwise, the first field of a line in */etc/d_passwd* contains the name of some shell program used in */etc/passwd* (for example, */bin/sh*). Any user who logs in on a tty line listed in */etc/dialups* and who uses a login shell listed in */etc/d_passwd* must enter the dial-in password stored in */etc/d_passwd*.

The second field of a line in */etc/d_passwd* either contains the encrypted dial-in password, or remains empty. If you leave the second field empty, the shell specified in the first field does not require a dial-in password. Use the following procedure to create an encrypted dial-in password for a login shell:

1. Create a new line in */etc/d_passwd* consisting of the name of the desired login shell followed by a colon (:).
2. Create a dummy user in */etc/passwd*, either by running **mkuser**(ADM) or by editing the file directly.

3. Run **passwd(C)** to set the password for the dummy user.
4. Copy the dummy user's encrypted password from */etc/passwd* to the second field of the line you created in */etc/d_passwd*. Add a colon (:) after the encrypted password. You can add a comment after the colon, if you wish.
5. If you ran **mkuser** in step 2, run **rmuser(ADM)** to remove the dummy user's home directory and system mailbox from the system. If you edited the */etc/passwd* file directly in step 2, edit that file again to remove the dummy user's line.

Note

You can also use a dummy user account to create a group password. For more information, see **group(F)** and **newgrp(C)**.

To begin recording login information, create a log file using the following command:

```
touch /usr/adm/lastlog
```

This file records the time of each login and displays the time of last login each time a user logs in. In addition, if this file exists and the user is not currently logged in, you can use the **finger(C)** utility to report the time of last login.

Establish the proper ownership and permissions on these files by entering the following commands:

```
chmod 644 /etc/d_passwd /etc/dialups /usr/adm/lastlog  
chown bin /etc/d_passwd /etc/dialups /usr/adm/lastlog  
chgrp bin /etc/d_passwd /etc/dialups /usr/adm/lastlog
```

You can give the two files in */etc* greater restrictions if you wish, but you must give *lastlog* the permissions specified above.

Permitting Users to Mount Filesystems

Only the superuser can use the **mount(ADM)** command. However, users can mount filesystems with the **mnt(C)** command. The superuser must set up parameters to define which filesystems users can mount. These parameters can include the use of an access password.

Each filesystem must have an entry in the file */etc/default/filesys*. Figure 5-1 contains a sample set of entries.

```

bdev=/dev/root cdev=/dev/rroot mountdir=/ \
desc="The Root Filesystem" rcmount=no mount=no

bdev=/dev/u cdev=/dev/ru mountdir=/u rcmount=yes \
fsckflags=-y desc="The User Filesystem"

bdev=/dev/x cdev=/dev/rx mountdir=/x mount=yes \
rcmount=yes fsckflags=-y desc="The Extra Filesystem"

```

Figure 5-1 Sample */etc/default/filesys* Entries

In simple terms, the entries above determine the following:

Table 5.4
Sample Mount Permissions

<u>Filesystem</u>	<u>When Mounted</u>	<u>Can User Mount?</u>
root	boot time	no
/u	multiuser	no
/x	anytime	yes

To make any non-root filesystem mountable by users, simply add “mount=yes” to the entry for the given filesystem, as demonstrated in the third code example above. In addition, when anyone invokes the **mnt** command without an argument (that is, without specifying a filesystem), the program checks all non-root filesystems to see whether users can mount them. If users can mount the filesystems, **mnt** mounts them. The option “mount=prompt” asks if a user wants to mount each filesystem where a mount is permitted.

Important

Do not include entries for removable media, such as a floppy device, in the */etc/default/filesys* file. Allowing users to mount removable media devices severely compromises system security.

For more information on the **mnt** command, including a complete list of options, refer to the **mnt(C)** page in the *XENIX User's Reference*.

Monitoring Login Sessions

Sometimes you can track a suspected misuse of the system by monitoring login sessions. The `last(C)` command displays login information from the binary file `/etc/wtmp`. (For information on monitoring processes other than login, see the section “Using XENIX Accounting Features” later in this chapter.)

To display a user name, tty line, process ID, time of login, and length of login in hours and minutes for each login, enter the command without arguments:

last

You see a display similar to the following:

User	Line	PID	Login time	Elapsed Time	Comments
agathac	001	5187	Wed Feb 6 14:34	00:01	logged in
acd	T1	4352	Wed Feb 6 14:31	00:22	logged in
agathac	001	4726	Wed Feb 6 13:48	00:38	
amandac	013	4465	Wed Feb 6 13:42	01:10	logged in
dorothys	T1	3888	Wed Feb 6 12:37	00:28	
agathac	001	2599	Wed Feb 5 14:35	01:58	
isaaca	p1	191	Wed Feb 5 13:21	03:30	
isaaca	p0	190	Wed Feb 5 13:20	03:31	

To display the login information without the column headings, enter:

last -h

To display only a certain number of the most recent login sessions, enter the following, where *limit* is the number of lines that you want to see:

last -nlimit

To display information for a specific tty line, enter the following, where *line* is the tty line that you want to monitor:

last -lline

Monitoring su Attempts

If you create the file */etc/default/su* to control the **su(C)** command, you can use the **SULOG** variable to monitor **su** attempts. Set the variable to a log file (in this example, */usr/adm/sulog*), by inserting the following line in */etc/default/su*:

```
SULOG=/usr/adm/sulog
```

The log file that you name records the following information on each **su** attempt: the date and time, the tty line, and the user IDs of the original user and the target user. A line describing a successful attempt contains a plus sign (+); a minus sign (-) indicates an unsuccessful attempt. Figure 5-2 shows a sample log file.

```
SU 02/08 10:06 + tty003 root-maryr
SU 02/08 13:21 - tty000 guest-root
SU 02/09 12:14 + tty001 dorothis-autotmp
SU 02/09 12:50 + tty001 dorothis-autotmp
SU 02/10 16:56 + tty003 root-maryr
```

Figure 5-2 Sample su Log File

Using XENIX Accounting Features

The XENIX system provides a set of commands that allows the system administrator to perform process accounting. Process accounting is a simple way to keep track of the amount of time each user spends on the system. The process accounting commands keep a record of the number of processes (that is, programs) started by a user, how long each process lasts, and other information such as how often the process accesses I/O devices, and how big the process is in bytes.

You can use process accounting to develop a detailed record of system, command, and system resource usage. From this record, you can investigate any suspicious use of restricted accounts or certain commands.

The **accton**(ADM) command starts and stops process accounting. When you run **accton**, the command copies information about each process to the file */usr/adm/pacct*. The **acctcom**(ADM) command displays this information. You must be logged in as the superuser to use these commands.

Starting and Stopping Process Accounting

While you can begin process accounting at any time, typically you start it when you start the system. Start process accounting by entering:

```
accton /usr/adm/pacct
```

This command creates a new file */usr/adm/pacct* and begins to copy process-accounting information to it. If the */usr/adm/pacct* file exists before you start **accton**, the file's previous contents are deleted.

Monitor the size of the */usr/adm/pacct* file, and clean it out regularly to keep it from getting too large. You might want to establish a **cron**(C) job to back up and empty the file periodically.

You can stop process accounting at any time by entering:

```
accton
```

To start process accounting automatically whenever you reboot the system, edit the system startup file */etc/rc.d/6/kinit* as the superuser and remove the number signs (#) that begin the following lines:

```
# mv /usr/adm/pacct /usr/adm/opacct
# > /usr/adm/pacct ; chmod 644 /usr/adm/pacct
# [ -x /etc/accton ] && /etc/accton /usr/adm/pacct
```

These commands start up process accounting and back up the accounting log file */usr/adm/pacct*. The number signs are comment symbols that cause the system to ignore the lines. When you remove the number signs, the system executes the commands every time you reboot the system.

Note

When you start the system after editing */etc/rc*, the system saves the contents of the */usr/adm/pacct* file to the file */usr/adm/opacct*, overwriting any previous contents of */usr/adm/opacct*.

Displaying Accounting Information

5

To see process-accounting information from */usr/adm/pacct*, enter:

acctcom

To see process-accounting information from another file (for example, a file that contains old information), specify the filename as follows:

The **acctcom** command displays the information from */usr/adm/pacct* or from the specified file onto your terminal screen. If you invoke **acctcom** without options, the command displays the following basic information:

- process' program name
- name of the user who invoked the process
- tty associated with the process
- start and stop times for the process
- number of execution seconds in real time and CPU time
- process' mean size in Kilobytes

Using XENIX Accounting Features

You can select the information that you want to see by using one of the **acctcom** options.

To display the basic information about a specific terminal, enter the following, where *line* is the tty (example: *tty1a*) that you want to monitor:

```
acctcom -l line
```

You see each entry in */usr/adm/pacct* (or your chosen file) for the specified terminal.

To display the basic information about a specific command, enter the following, where *command* is the command that you want to monitor:

```
acctcom -n command
```

You see each entry in */usr/adm/pacct* (or your chosen file) for the specified command.

To display the basic information about a specific user, enter the following:

```
acctcom -u user
```

For *user*, you can substitute a user ID, a login name, a number sign (#), or a question mark (?). The # argument displays processes executed by the super user, and the ? displays processes associated with unknown user IDs.

To display the amount of available CPU time that a process consumes, plus the basic information, enter:

```
acctcom -h
```

This command displays the “use factor” or “hog factor,” which is a number that the system generates according to the following formula:

$$(\text{total CPU time})/(\text{elapsed time})$$

This number determines how the system schedules each process for execution. Processes with high use factors use a high percentage of the system resources and are, therefore, scheduled after processes with lower factors.

To display the basic information on processes that exceed a specific use factor, enter the following, where *factor* is the use factor that interests you:

acctcom -H *factor*

This option does not display use factors; you must add the **-h** option if you want to see the use factors of the processes in question.

To display the basic information on processes that exceed a specific operating system CPU time, enter the following, where *time* is the operating system CPU time that interests you:

acctcom -O *time*

The **acctcom(ADM)** manual page describes more options for the command.

Setting up a Restricted Shell

The XENIX restricted Korn shell, **rksh**(C), and restricted Bourne shell, **rsh**(C), limit a user to one shell, one directory, one login environment, and one set of commands, all specified by the system administrator. The system administrator establishes a restricted shell when a user does not need the full power of the Korn shell or Bourne shell and the XENIX utilities.

The restricted Bourne shell is much less secure than the restricted Korn shell. Do not use the restricted Bourne shell to restrict hostile users.

Creating a Restricted Korn Shell Account

Follow these steps to set up a restricted Korn shell account:

1. Log in as the super user.
2. Run **mkuser**(ADM), and select **rksh** as the user's login shell.

Under the restricted Korn shell, the user cannot:

- change directories with **cd**
 - set **PATH**, **SHELL**, or **ENV**
 - execute command names that contain slashes (/)
 - redirect output with **>**, **>|**, **<>**, or **>>**
 - execute any command for which you, the system administrator, do not expressly grant access
3. Customize the default restricted Korn shell version of the *.profile* file, if necessary, and install it in the user's home directory. The default *.profile* file includes the recommended settings for the **SHELL**, **ENV**, and **PATH** variables. If you change these settings, be sure that you do not destroy the restrictions of the shell.
 4. Customize the default restricted Korn shell version of the *.rkshrc* file, if necessary, and install it in the user's home directory.

5. Use **mkdir(C)** to create a *bin* directory in the user's home directory.
6. Decide which commands your restricted user needs. Link those commands from */bin* and */usr/bin* to the *bin* directory in the user's home directory.

Note

The restrictions of this shell take effect only after the *.profile* and *.rshrc* scripts finish executing. If the user tries to interrupt either script, the restricted Korn shell exits.

Creating a Restricted Bourne Shell Account

Follow these steps to set up a restricted Bourne shell account:

1. Log in as the super user.
2. Run **mkuser**, and select *rsh* as the user's login shell. If the user already has an account, edit the file */etc/passwd* to specify */bin/rsh* as the user's login shell.
3. Use **mkdir** to create the restricted directories */restrict* and */usr/restrict*.
4. Decide which commands your restricted users need. Link those commands from */bin* and */usr/bin* to the restricted directories.
5. Use **chown** to make *root* the owner of the user's *.profile* file. (See the section "Changing User Ownership" in this chapter for information on using **chown**.)
6. Use **chmod** to make the user's *.profile* file readable and writable by the file owner (*root*) and readable by group and other. (See the section "Changing Permissions" in this chapter for information on using **chmod**.)

Setting up a Restricted Shell

7. Set the **PATH** variable to the restricted directories by editing the user's *.profile* file to read:

```
PATH=/restrict:/usr/restrict
```

8. Set the **SHELL** variable to control the shell escapes by editing the user's *.profile* file to read:

```
SHELL=/bin/rsh
```

Special Uses for the Restricted Shell

You can establish a restricted account that only allows a user to log into a certain application; for example, SCO FoxBASE+ or SCO Professional. This kind of account is called a "front-end" account. After you create the restricted account, edit the user's *.profile* file to include the following line:

```
exec application
```

Replace *application* with the name of the application that you want the user to run.

You can also establish a front-end account that allows only specified system administration tasks. You can then delegate these tasks to another user. Follow these steps to create a "system administration" shell:

1. Write a shell script or program to perform the chosen system administration tasks.
2. Make a new user account and assign a password.
3. On the line for the new account in */etc/passwd*, set the third and fourth fields, which specify user ID and group ID, to 0.
4. Place the following line at the bottom of the account's *.profile* file:

```
exec adminprogram
```

Replace *adminprogram* with the name of your system administration program.

Chapter 6

Backing Up Filesystems

Introduction 6-1

Strategies for Backups Using sysadmsh 6-2

Using the backup Account for Backups 6-2

Floppy Drive Backups and Large Systems 6-3

Summary of Utilities Accessed 6-3

Preparations for Scheduled Backups 6-4

Creating a Backup Schedule 6-4

Labeling Your Backups 6-9

Keeping a Log Book 6-10

Rotating Backup Media 6-10

Archiving Backup Media 6-10

Performing a Scheduled Backup 6-11

Using Formatted Media 6-11

Starting the Backup 6-12

Performing an Unscheduled Backup 6-15

Getting a Backup Listing 6-18

Restoring Individual Files or Directories From Backups 6-20

Restoring an Entire Filesystem 6-24

Editing /etc/default/filesys and /etc/default/archive 6-29

/etc/default/archive 6-29

/etc/default/filesys 6-30

An Explanation of Backup Levels 6-31

Principles of Incremental Backup Levels 6-31

How the Default and Alternate Schedules Work 6-32

How Backups Restore a Filesystem 6-34

Introduction

The main task of a system administrator is to ensure the continued integrity of information stored on the system. Files and filesystems can be damaged and data lost in the following ways:

- power interruptions (make certain you have a surge protector)
- hardware failures (particularly the hard disk)
- user errors (accidental removal of important files)

The importance of having up-to-date backups cannot be overstated. If your system has a number of active accounts, backups require daily attention. It is difficult to estimate the magnitude of a simple loss of data until an accident occurs and several weeks or months of work is gone in an instant.

A filesystem backup is a copy, on storage media (floppy disks or tape) of the files in the root filesystem and other regularly mounted filesystems (for example, the */u* filesystem). (See the “Managing Filesystems” chapter in this guide for a discussion of filesystems.) A backup allows the system administrator (or an assigned user with the *backup* account password) to save a copy of a filesystem as it was at a specific time.

This chapter explains how to use **sysadmsh**(ADM) to create backups of the root directory and other filesystems, and how to restore files from the backups. (Another utility used for simple backups, **tar**(C), is discussed extensively in “Making Backups” in the “Housekeeping” chapter of the *XENIX Tutorial*. **tar** is not sufficiently sophisticated to perform scheduled backups; it is better suited to archiving groups of files.)

The tools discussed in this chapter present menus with simple options instead of the complex command lines used with the utilities **tar**(C), **cpio**(C), **backup**(ADM), and **restore**(ADM). The key to efficient backups is to save only what has changed from day to day, which (when used with **backup** and **restore**) normally requires extra bookkeeping.

Strategies for Backups Using `sysadmsh`

As system administrator, you should familiarize yourself with this chapter and create a backup schedule as instructed. When this schedule is complete, you have only to insert a media volume and respond to a series of prompts to perform your daily backups.

The primary purpose of the `sysadmsh` filesystem backup selection is to provide a dependable schedule of filesystem backups for systems with many users and large filesystems. The program automatically locates modified files and copies them to backup media. If your system has many users and a large number of files that are modified daily, the “scheduled” backup option uses a predefined schedule to make regular backups. When the Backups selection is invoked, the program presents each task as a menu option. To perform a task, simply choose the appropriate option from the menu and supply any required information.

For backups of an informal nature, `sysadmsh` includes an option for “unscheduled” backups. This allows the system administrator to perform a single, complete backup of a filesystem. (Note this type of backup covers the entire filesystem, not just modified files, and may require a number of storage media volumes.) If you intend to rely on unscheduled backups, be sure and perform one at least once a week.

Using the backup Account for Backups

Always use the `backup` account whenever you make or restore backups. (You must be `root` to restore an entire filesystem.) An ordinary user cannot make backups because they do not have access permissions for all files. If backups are made as `root`, files may be accidentally destroyed because `root` has unrestricted permissions on every file on the system. The `backup` account solves this dilemma by having restricted `root` permissions. Logging in as `backup` takes you directly to the Filesystem Maintenance Menu.

The `backup` account already exists on your system; you set the password when the BACKUP package was installed.

Floppy Drive Backups and Large Systems

If your system has only a floppy drive, backups for large systems with several users can be time-consuming and use a great deal of media. A complete backup of a 20-Mbyte filesystem requires 15 1.2-Mbyte 96tpi diskettes, while a single 450-foot cartridge tape can store more than twice that amount. More importantly, diskettes require the presence of the operator to insert and remove floppies, whereas a single cartridge tape can be inserted and the operator need not remain by the system. If your system has a large number of users and just a floppy drive, you should install a cartridge tape drive, or make complete system backups once per week and warn your users to make individual backups of their own files on a regular basis.

Summary of Utilities Accessed

The `sysadmsh` accesses several utilities during the backup process. You need not be familiar with them. However, should you wish to use advanced options not discussed in this chapter, you will need to know how they are used and which reference pages to read. `sysadmsh` accesses the following utilities:

- `fsphoto(ADM)` is the main utility that controls the automated backup facilities.
- `fsave(ADM)` is the program that interacts with the user to perform the backup.
- `schedule(ADM)` is the backup database that specifies the media to be used, the filesystems to be backed up, and when to do so.
- `backup` and `restore(ADM)`: the actual backup utilities.

The `sysadmsh` Backups selection forms the “user-friendly” layer that isolates the user from the complicated syntax of these programs.

Preparations for Scheduled Backups

The only mandatory requirement for scheduled backups is the creation of a backup schedule. In addition, it is recommended that the system administrator follow the optional procedures for labeling, storing, and logging backups. A detailed explanation of backup levels is included at the end of this chapter in case it is necessary to design a more specialized schedule.

Creating a Backup Schedule

The first step is to create a timetable for backups using the *schedule* file. This file is located in the */usr/lib/sysadmin* directory and contains all the data needed for the system to perform a system backup, including:

- the name of your site or machine
- the media type and drive to be used
- a precise schedule of filesystems to be backed up

The sections that follow explain what changes should be made to the *schedule* file provided with your distribution.

Edit the schedule File

You can edit the *schedule* file with any text editor; make certain you are logged in as *root*. Figure 6-1 shows the default *schedule* file. You can also use the following *sysadmsh* selection to edit the *schedule* file:

Backups→Schedule

sysadmsh uses the *vi*(C) editor by default, but you can set the *SA_EDITOR* environment variable to the editor you prefer. See *environ*(M) or *sh*(C) for an explanation of how to set environment variables. The subsections that follow explain the exact changes you need to make to this file.

```
# SYSTEM BACKUP SCHEDULE
site mymachine
# Media Entries
#
# 96 tpi 1.2 MB floppy 0
# media /dev/rfd096ds15 k 1200 format /dev/rfd096ds15
# 96 tpi 1.2 MB floppy 1
# media /dev/rfd196ds15 k 1200 format /dev/rfd196ds15
# 135 tpi 1.44 MB floppy 0
# media /dev/rfd0135ds18 k 1440 format /dev/rfd0135ds18
# 135 tpi 1.44 MB floppy 1
# media /dev/rfd1135ds18 k 1440 format /dev/rfd1135ds18
# Cartridge tape 1
media /dev/rct0 k 60000 125000 150000 tape erase
# Mini cartridge drive (10MB)
# media /dev/rctmini k 8800 format /dev/rctmini
# Mini cartridge drive (20MB)
# media /dev/rctmini k 17200 format /dev/rctmini
# Mini cartridge drive (40MB)
# media /dev/rctmini k 37500 format /dev/rctmini
# 9-track tape drive
# media /dev/mt0 d 1600 2400 1200 600

# Backup Descriptor Table

# Backup  Vol.  Save for  Vitality  Label
# level  size  how long  (importance)  marker
0        -    "1 year"  critical    "a red sticker"
1        -    "4 months"  necessary  "a yellow sticker"
2        -    "3 weeks"  useful     "a blue sticker"
3        -    "1 week"   precautionary  none

# Schedule Table
#          1 2 3 4 5   6 7 8 9 0   1 2 3 4 5   6 7 8 9 0
# Filesystem  M T W T F   M T W T F   M T W T F   M T W T F
/dev/rroot   0 x 3 x 3   2 x 3 x 3   1 x 3 x 3   2 x 3 x 3
/dev/ru      3 0 3 3 3   3 2 3 3 3   3 1 3 3 3   3 2 3 3 3
```

Figure 6-1 The schedule File

Add the Name of Your Site or Machine

Simply change the *mymachine* entry at the top of the file to the name you wish.

Select the Media Device that Matches Your Configuration

The schedule file differs according to your distribution media; it may not match Figure 6-1. In addition, the default drive is either the cartridge tape or, in the case of XENIX-286 distributions, the floppy drive that matches the media type of your distribution. The pound signs (#) are comment symbols used to “comment out” text so that it is ignored by the

Preparations for Scheduled Backups

program. Note that the default drive is the only one without a comment symbol. If you plan to use a drive other than the default, put a comment symbol in front of the default drive and remove the comment symbol from in front of the drive you wish to use. The remaining drives should remain commented out.

```
# 96 tpi 1.2 MB floppy 0
media /dev/rfd096ds15 k 1200 format /dev/rfd096ds15

# 96 tpi 1.2 MB floppy 1
# media /dev/rfd196ds15 k 1200 format /dev/rfd196ds15
```

Figure 6-2 Default Media Entry

Edit the Backup Descriptor Table

Directly below the media drive lines is the Backup Descriptor table. This table, reproduced in Figure 6-3, describes each backup level in terms of volume size, how long it is to be stored, how important it is, and how it is marked. The default entries should prove useful, but the volume size entries must be edited according to the type of media you are using.

#	Backup level	Vol. size	Save for how long	Vitality (importance)	Label marker
0		-	"1 year"	critical	"a red sticker"
1		-	"4 months"	necessary	"a yellow sticker"
2		-	"3 weeks"	useful	"a blue sticker"
3		-	"1 week"	precautionary	none

Figure 6-3 Backup Descriptor Table

If you are using floppy disks, leave the dashes in the "Vol. size" column as they are. This causes the backup program to take the volume size from the media entry for that device.

If you are using tapes or tape cartridges, replace each dash in the "Vol. size" column with the size (in kilobytes) of the tape volume. If you are using tapes that are all the same size for each backup level, replace each with the size of the tape you are using.

The last column contains label entries that are discussed in "Labeling Your Backups" later in this section.

Edit the Backup Schedule Table

The default schedule assumes that backups are done every day. To make backups more efficient, they are broken into levels. Level 0 is the lowest level backup. A level 0 backup saves everything on the filesystem, while 1, 2, and 3 each save only the files that have changed relative to the last lower-level backup. This concept is illustrated in Figure 6-4 with a stack of toy disks representing each level. Notice that the level 0 disk is the largest backup, and each of the others is progressively smaller. This is because level 1 contains only the files that were changed since the level 0 was done, and so on. This figure also illustrates how these backups would be restored: first the level 0, followed by each of the latest 1, 2, and 3 that were done.

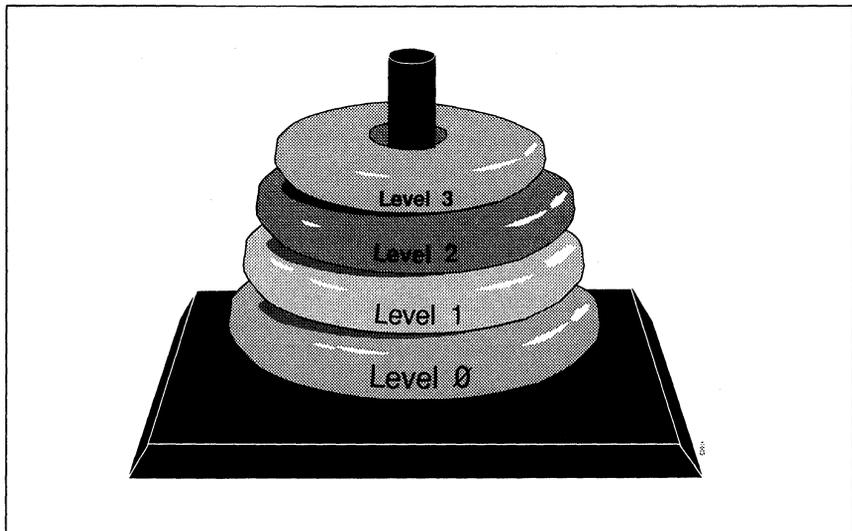


Figure 6-4 Backup Levels

The concept of levels may seem needlessly complex at first, but consider what would happen on a system with a number of large filesystems. If you performed a full backup of each filesystem each night, the process would take hours to perform, bogging the system down in the process. If only the files that changed most recently are saved, backups would be less time-consuming and, depending on the size of your media, consume fewer volumes (see “An Explanation of Backup Levels” at the end of this chapter for a more detailed discussion of backup levels).

Note

XENIX-286 distributions use levels 0, 1, 8, and 9. There is no functional difference between the two numbering schemes.

The example *schedule* file in this chapter includes two filesystems, root and */u*. When you installed the system, you were asked if you had a */u* filesystem, and if you answered yes, the */u* line was left in place. If you responded no, only the root filesystem is in the *schedule* file and the schedule table was altered to backup the root filesystem each day. In the sample file, there is a backup done every other day for the root filesystem and once a day for the */u* filesystem. This is because the */u* filesystem (user accounts) changes much more frequently than the root filesystem, which contains the system files. An “x” means that a backup is not performed on that day for that filesystem.

If you do not have a */u* filesystem, then your user accounts are located in the root filesystem (in the directory */usr*). If this is so, the schedule table is preconfigured to back up the root filesystem. However, if you have added a */u* filesystem, edit the schedule table and remove the # in front of the entry for */dev/ru*, shown in Figure 6-5. This ensures that backups are made of the additional filesystem. If you do not have a */u* filesystem, but you do want daily backups, this entry can also be modified and used for the root filesystem.

#	1 2 3 4 5	6 7 8 9 0	1 2 3 4 5	6 7 8 9 0
# Filesystem	M T W T F	M T W T F	M T W T F	M T W T F
/dev/rroot	0 x 3 x 3	2 x 3 x 3	1 x 3 x 3	2 x 3 x 3
/dev/ru	3 0 3 3 3	3 2 3 3 3	3 1 3 3 3	3 2 3 3 3

Figure 6-5 Backup Schedule Table

Note that the Monday-Friday notation can be misleading; if a backup is postponed or unsuccessful (because of bad media, for example) then that same level backup is attempted again at the next scheduled backup. This offsets the schedule, but does not alter the established sequence of backups. The numbered scale of 1-0 above M-F is more accurate, but less useful to people, who work in day and week units.

In addition, if you add lines for other filesystems, you should take care not to schedule two level 0 backups of large filesystems on the same day; the process is lengthy and may slow your machine significantly.

Labeling Your Backups

It is important to label your backup tapes with meaningful and accurate information. If your backups consist of a pile of haphazardly labeled tapes, it will be difficult to locate data at a later date.

Figure 6-6 is a suggested format for media labels.

Name of computer	Backup level Filesystem Name save until date	Date made
Name of backup person		volume # of #

Figure 6-6 Sample Media Label

The date on the label, and the date from which you calculate the “save until” date, should be the date of the business day covered by the backup. This is to avoid confusion if it becomes necessary to restore information from this tape.

You may have noticed that the *schedule* file has a proposed color-coding scheme for easy reference, as emphasized in Figure 6-7.

#	Backup level	Vol. size	Save for how long	Vitality (importance)	Label marker
0		-	"1 year"	critical	"a red sticker"
1		-	"4 months"	necessary	"a yellow sticker"
2		-	"3 weeks"	useful	"a blue sticker"
3		-	"1 week"	precautionary	none

6

Figure 6-7 Backup Labeling Scheme

If there is more than one tape for a single backup, mark the date label on each volume to indicate the volume number and number of volumes, such as “1 of 2” and “2 of 2” for a two-volume backup. Finally, place a label on the side of the box or enclosure marked with the name of the computer, the filesystem, and the backup level completed.

Keeping a Log Book

It is recommended that a written log book be maintained for each computer. In addition to maintenance information (such as when breakdowns occur and what was done about it), you should record the following information:

Date	Just as with the tape label, this date should be the last day covered by the backup.
Filesystem	This is the name of the device backed up on the current tape.
Backup level	This is the backup level of the current tape.
#Vols	This is the number of tape volumes.
Start/finish time	(Optional.) The time from the start of a backup of a filesystem until the last error check is completed. The times are displayed after the backup is finished. The finish time will often be inaccurate, since you may be out of the room when the backup finishes, and the machine sits idle before you return.

If there are problems with the backup, record these in the log book as well, including any error messages that come to the screen.

Rotating Backup Media

Backup media should be used so that at least 6 to 12 months of media are left on file. The default *schedule* file includes a suggested rotation of 1 week on level 3 backups; 3 weeks on level 2; 4 months on level 1; and 1 year on level 0. This means that if you follow the default schedule, you can safely re-use your level 3 backups after 1 week, and so on.

Archiving Backup Media

All filesystems should be periodically backed up and archived offsite. In the event of a fire or natural catastrophe, the data can later be restored.

Performing a Scheduled Backup

This section describes how to perform a backup using a defined schedule. Do not attempt this until you have edited (or at least examined) the schedule file to make certain that it suits your needs.

The system administrator should schedule backups at times when few (if any) users are on the system. This ensures minimal interference with user processes, and that most recent versions of system files are copied correctly.

A regular schedule of backups requires a good supply of media and adequate storage for them. Level 0 backups should be saved indefinitely; lesser backups should be saved at least two weeks. Media volumes should be properly labeled with the date of the backup and the names of the files and directories contained in the backup. After a backup has expired, the media may be used to create new backups.

Using Formatted Media

If you use media that requires formatting, such as floppy disks or mini tape cartridges, you may wish to format several volumes before you begin. The exact number of volumes depends on the number and size of files to be backed up. For details on how to format your media, see the “Using Floppy Disks and Tape Drives” chapter in this guide. You also have the option to do formatting from the **sysadmin** program.

6

Note

Formatting mini tape cartridges is not recommended. Best results are achieved with preformatted cartridges.

Performing a Scheduled Backup

Starting the Backup

To run your scheduled backup, follow these steps:

1. Log in as *backup*. After any login messages are displayed, you are taken directly to the Filesystem Maintenance Menu:

- ```
1. Perform a scheduled backup
2. Perform an unscheduled backup
3. List the contents of an archive
4. Restore backed up file(s)
5. Restore an entire filesystem
6. Check backup archive integrity
```

```
Enter an option or enter q to quit:
```

2. Enter "1" for a daily backup and press `<Return>`.
3. A menu is displayed that looks like the following:

```
Level 0 backup of filesystem /dev/rroot, 22 Sep 1988
 tape size: 450 feet [or Kb]
 tape drive: /dev/rct0
This tape will be saved for 1 year, and is critical.

M)ounted volume, P)ostpone, C)heck or F)ormat volumes,
 R)etension or H)elp:
```

The media type displayed is the one entered in the **schedule** file. Load a volume, tape or disk, into the selected drive. Enter "m" to tell the program the volume is mounted, and press `<Return>`.

## Performing a Scheduled Backup

4. The system displays the current date and the date of the last backup:

```
Backup date= the epoch
Backing up /dev/rroot to /dev/rct0
```

(If you have not performed a backup previously, the system has no last backup date recorded and “the epoch” is displayed as the previous backup date.) An estimate of the number of volumes necessary is displayed. The system then begins to copy files to the drive. If a volume runs out of space, the program displays the message:

```
Please insert new volume, then press <RETURN>
```

---

### Note

If you are using 5.25-inch floppies for your backups, make certain you close the floppy door before pressing <Return>, or the entire backup will be aborted and you will have to start over.

---

Remove the present volume, insert a new volume, then press <Return>. The program continues to copy files to the new volume. Repeat this step until the program displays the message:

```
DONE
```

5. After the backup is complete, you are advised to select “c” from the main menu to perform a check of the format. This is not a check of the format used in formatted media, but a confirmation of the presence of header information that should be present on each volume of a backup. The following message is displayed:

```
Check critical volumes for format errors
M)ounted first volume, S)kip format check, or H)elp:
```

## Performing a Scheduled Backup

If you wish to have the volume checked, insert the first volume of the backup, enter “m” and press `<Return>`. If you wish to skip checking the volume format, and continue on to the read error check, enter “s” and press `<Return>`.

---

### Note

During verification, you may see messages indicating that a file on the hard disk differs from the backup. This is because certain files can change during the short interval between backup and verification. These messages do not affect the backup and can be ignored.

---

6. Every volume should be checked for read errors, and they must be checked in first-to-last order. (The `restore(ADM)` utility is used to check for errors during reading.) If an error occurs, the backup is declared unsuccessful and is retried from the beginning. The menu appears as follows:

```
M)ounted which volume, E)rror on previous volume, D)one,
S)kip checks, or H)elp:
```

7. If you wish to check each of the backup volumes, you should insert them in order, enter “m” and press `<Return>`. If you wish to skip the check, enter “s” and press `<Return>`.
8. If an error occurs on the last volume checked, discard the suspect media and start the backup again by entering “e” and pressing `<Return>`.
9. When all volumes have been checked and no errors have occurred, enter “d” and press `<Return>` to exit the program

After the backup has been successfully performed, instructions are given on how to label the volumes.

---

## Performing an Unscheduled Backup

Unscheduled backups are intended for making a simple complete backup of a filesystem. It can be used by those who do not wish to use the scheduled backups, or when it is necessary to do an unscheduled backup.

You can create backups on tape or disk. If you use media that requires formatting, such as floppy disks, you may wish to format several volumes before you begin. The exact number of volumes depends on the number and size of files to be backed up. For details on how to format media, see the “Using Floppy Disks and Tape Drives” chapter in this guide. You also have the option to do formatting from within the Filesystem Maintenance Menu.

To create a backup, follow these steps:

1. Log in as *backup*. The Filesystem Maintenance Menu is displayed:

```
1. Perform a scheduled backup
2. Perform an unscheduled backup
3. List the contents of an archive
4. Restore backed up file(s)
5. Restore an entire filesystem
6. Check backup archive integrity
```

```
Enter an option or enter q to quit:
```

2. Enter “2” for an unscheduled backup and press (Return).
3. You see the following menu:

```
1. / - the root filesystem
2. /u -
3. Other
```

```
Select a filesystem to backup
or enter q to return to the main menu:
```

The menu lists all filesystems found in the file */etc/default/filesys* (discussed later in “Editing */etc/default/filesys* and

## Performing an Unscheduled Backup

`/etc/default/archive`). Select the number of the filesystem you wish to backup and press `<Return>`. (If you wish to enter a filesystem that is not listed in `/etc/default/filesys`, select "Other" and you will be prompted for the name.)

4. Next, you are asked to select the media device to be used:

1. Floppy Drive 0 (48dsdd)
2. Floppy Drive 1 (48dsdd)
3. Floppy Drive 0 (96dshd)
4. Floppy Drive 1 (96dshd)
5. Floppy Drive 0 (96dsdd)
6. Floppy Drive 1 (96dsdd)
7. Cartridge Drive (300 ft tape)
8. Cartridge Drive (450 ft tape)
9. Cartridge Drive (600 ft tape)
10. Mini-Cartridge Drive (10MB)
11. Mini-Cartridge Drive (40MB)
12. Other

Select an archive device,  
or enter `q` to return to the main menu:

Select the number that corresponds to the device you wish to use. The devices appearing in this menu are taken from `/etc/default/archive`; you can add entries or even simplify this menu by editing this file (see "Editing `/etc/default/filesys` and `/etc/default/archive`" in this chapter).

---

### Note

Take care when selecting the number of the media device. For example, make certain that you don't select "Floppy Drive 1" when you want "Floppy Drive 0." If you make this error, the backup is aborted and you must start over.

---

5. The following message is displayed:

```
It is important to have plenty of formatted media on hand.
Do you wish to format any media at this time? (y/n)
```

You can format as many volumes as you wish by inserting them into the drive and pressing **(Return)**.

6. Next you see:

```
PERIODIC BACKUP (level 0) - /name FILESYSTEM
```

```
Load the first volume into the drive /dev/name,
press <RETURN> when you are ready,
or enter q to return to the main menu:
```

7. Load a tape or disk volume into the selected drive, and press **(Return)**. The system displays the current date and the date of the last backup. It displays “the epoch” if there has been no backup. The system then begins to copy files to the drive. If a volume runs out of space, the program displays the message:

```
Please insert new volume, then press <RETURN>
```

8. Remove the first volume, insert a new volume, then press **(Return)**. The program continues to copy files to the new volume. Repeat this step until the program displays the message:

```
DONE
```

If you are using floppies, you may need to repeat the last step several times before the backup is complete. You should label each volume as you remove it from the drive. For example, label the first volume “Volume 1,” the second “Volume 2,” and so on.

---

# Getting a Backup Listing

You can keep a record of the files you have backed by selecting item 3: “List the contents of an archive” from the Filesystem Maintenance Menu. The program copies the names of all files from the backup disks to the temporary file */tmp/backup.list*, or to another file of your choice. This listing is useful when you need to recover a file from a backup, and especially convenient if you wish to keep detailed records of the files copied in each backup.

To get the listing, follow these steps:

1. Log in as *backup*

$\Delta$  **sysadmsh** users select: Backups→Create→Filesystems

2. The Filesystem Maintenance menu is displayed:

```
1. Perform a scheduled backup
2. Perform an unscheduled backup
3. List the contents of an archive
4. Restore backed up file(s)
5. Restore an entire filesystem
6. Check backup archive integrity
```

```
Enter an option or enter q to quit:
```

3. Enter “3” and press  $\langle$ Return $\rangle$ . You are prompted for the name of the file in which to place the listing. Enter  $\langle$ Return $\rangle$  if you wish to use the default */tmp/backup.list*. The program prompts you to insert the first backup volume.
4. Load the first volume, then press  $\langle$ Return $\rangle$ . The program automatically reads the filenames off the backup volume and places them in the list file.

To print the backup list on a lineprinter, quit **sysadmin** and enter:

```
lp /tmp/backup.list
```

△ **sysadmsh** users select: Dirs/Files→Print

and press <Return>. To save space after printing the file, you should remove it with the **rm(C)** command.

---

## Restoring Individual Files or Directories From Backups

You can restore individual files or subdirectories from your filesystem backup volumes by invoking **sysadmin** and selecting the fourth item in the Filesystem Maintenance Menu. You will need the complete set of backup volumes containing the latest version of the file or files you wish to restore. If you are restoring a file that has not been changed recently, use the last level 0 backup.

You must use the “full pathname” of the file or files you wish to restore. This pathname is given in the backup listing. If the files are not on the root filesystem, the name of the filesystem must be omitted from the pathname. For example, to restore the file */u/stellar/data* from your */u* backups, you would enter:

**/stellar/data**

To restore a file, follow these steps:

1. Log in as **root** and enter:

**sysadmin**

△ **sysadmsh** users select: Backups→Create→Filesystems

and press <Return>.

## Restoring Individual Files or Directories From Backups

2. When the Filesystem Maintenance Menu appears enter “4” and press `<Return>`. You see:

```
1. Floppy Drive 0 (48dsdd)
2. Floppy Drive 1 (48dsdd)
3. Floppy Drive 0 (96dshd)
4. Floppy Drive 1 (96dshd)
5. Floppy Drive 0 (96dsdd)
6. Floppy Drive 1 (96dsdd)
7. Cartridge Drive (300 ft tape)
8. Cartridge Drive (450 ft tape)
9. Cartridge Drive (600 ft tape)
10. Mini-Cartridge Drive (10MB)
11. Mini-Cartridge Drive (40MB)
12. Other
```

```
Select an archive device,
or enter q to return to the main menu:
```

Enter the number that corresponds to the drive used to create the backup originally.

3. Next, a message is displayed explaining that you can choose to restore the files to their original location (by providing the name of the top-level directory the filesystem is mounted on), or some other directory. This is followed by a prompt for the directory name:

```
Enter a directory name,
enter <RETURN> to choose the current directory,
or enter q to return to the main menu:
```

6

---

### Note

If you respond with the pathname of the original location, the restored files will overwrite any files by the same names in that location. It is important to be sure that the files on the backup are the desired versions of these files. If you are not absolutely sure that your backup contains the preferred version of the files, you should restore them to a temporary location, such as `/tmp`, and compare them with your current files on disk using `diff(C)` or `cmp(C)`.

---

## Restoring Individual Files or Directories From Backups

Enter the name of the directory you want the files restored to.

4. Next, you are prompted:

List the name of each file or directory to be restored.  
Do not include the destination (filesystem or directory) name,  
but use the rest of the full path name.

For example, if the destination is /usr, and the full path  
name is /usr/bin/lpr, then enter /bin/lpr.

Enter a file or directory and <RETURN>,  
enter <RETURN> if the list is complete,  
or enter q to return to the main menu:

Enter the full pathname of the files or directories you wish to  
restore and press <Return> to continue the program.

5. Next, you see:

```
RESTORE FILES - /file(s)
```

Load the first backup volume into drive /dev/*devicename*.  
Press <RETURN> when you are ready,  
or enter q to return to the main menu:

Load volume 1 of the backup set into the drive, then press <Return>.

6. The program displays the inode numbers of the files you have  
given, then prompts for a volume number:

```
/file(s): inode nn
```

```
Specify volume #:
```

7. Remove the first volume and replace it with the *last* volume made  
of the backup set into the drive, enter its number and press <Return>.  
The program searches the volume for the specified files and places  
copies into the specified locations on your hard disk.

## Restoring Individual Files or Directories From Backups

8. The program prompts for volume numbers until all of the files have been found. When each file is found, you see:

```
Extract file filename
```

Continue to feed volumes in reverse order until the first volume made has been loaded and you have returned to the main menu.

---

# Restoring an Entire Filesystem

Restoring an entire filesystem is a last-resort option, used when a non-root filesystem has become corrupted or unreadable. Do not use this option carelessly; all information currently in the target filesystem will be overwritten. You cannot restore the root filesystem using this utility. If your root filesystem has been corrupted and is not bootable, you can restore it by referring to “Restoring a Corrupted Root Filesystem” in the “Solving System Problems” chapter of this guide and using the Emergency Boot Floppy you created at installation time. If you did not create an Emergency Boot Floppy, you must reinstall XENIX as described in the *XENIX Installation Guide*.

To restore an entire non-root filesystem, you must first “remake” the filesystem using **divvy**. This will start you with a clear filesystem. Then you can invoke the “Restore entire filesystem” option.

To restore a filesystem, follow this procedure:

1. Log in as the super-user (**root**) and enter one of the following commands:

*If you are restoring a filesystem on the primary disk (/dev/hd00):*

```
divvy -b 1 -c 1 -p 0
```

*If you are restoring a filesystem on the secondary disk (/dev/hd10):*

```
divvy -b 1 -c 1 -p 1
```

2. You see a table similar to the following, plus the main **divvy** menu:

| Name     | New File System? | # | First Block | Last Block |
|----------|------------------|---|-------------|------------|
| root     | no, exists       | 0 | 0           | 13754      |
| swap     | no, exists       | 1 | 13755       | 15135      |
| u        | no, exists       | 2 | 15136       | 25135      |
|          | no               | 3 | —           | —          |
|          | no               | 4 | —           | —          |
|          | no               | 5 | —           | —          |
| recover  | no, exists       | 6 | 25136       | 25145      |
| d1057all | no               | 7 | 0           | 25546      |

x blocks for divisions, y blocks reserved for the system

```
n[ame] Name or rename a division.
c[reate] Create a new filesystem on this division.
p[revent] Prevent a new filesystem from being created...
s[tart] Start a division on a different block.
e[nd] End a division on a different block.
r[estore] Restore the original partition table.
```

Please enter your choice or 'q' to quit:

3. Enter “c” to recreate a filesystem. You are then prompted for a division number as displayed in column three:

6

```
which division? (0 through 6) --
```

Enter the number corresponding to the filesystem you wish to recreate.

---

### *Note*

You should take extreme care when selecting the filesystem to recreate. However, after quitting out of **divvy**, you can undo any mistakes by selecting “e[xit]” as described in the next step.

---

## Restoring an Entire Filesystem

4. The **divvy** menu is displayed again, but with “yes” in the “New File System?” column. Enter “q” to quit.
5. Next, you are given a final chance to undo your changes before leaving **divvy**:

```
i[nstall] Install the division set-up shown
r[eturn] Return to the previous menu
e[xit] Exit without installing a division table
```

Please enter your choice:

If you made a mistake, enter “e”, otherwise enter “i” to use your changes. You can then quit out of **divvy** entirely. When you quit, your filesystem is rebuilt. You see the message:

```
Making Filesystems
```

6. Next, while still logged in as root, enter the command:

**sysadmin**

**Δ sysadmsh users select: Backups→Restore→Filesystems**

When the Filesystem Maintenance Menu is displayed, select option “5”. You are warned:

```
WARNING: RESTORING A LEVEL 0 BACKUP WILL OVERWRITE YOUR FILESYSTEM.
```

```
Even if a filesystem has been damaged, it may contain valuable
information. It is very important to make sure that the set of
backups you restore from has not also been damaged before restoring
directly onto the damaged filesystem.
```

```
This option should be used to restore a level 0 backup onto a clear,
or newly created, filesystem. The most recent level 3 backup with a
later creation date than the level 0 should then be restored onto it.
```

```
Do you wish to continue? (y/n)
```

Respond “y” if you are certain that this is what you wish to do.

7. Next you are asked to select the filesystem that you wish to restore:

```
1. / - the root filesystem
2. /u -
3. Other

Select a filesystem to restore
or enter q to return to the main menu:
```

If you select “Other”, you are prompted for the device name of the filesystem that you wish to restore.

8. Next you see the archive menu. Select the medium on which your filesystem is backed up, for example, tape or floppy.
9. You are prompted to load the first volume of the backup into the selected drive:

```
RESTORE FILESYSTEM - /name

Load the first backup volume into drive /dev/devicename.
Press <RETURN> when you are ready,
or enter q to return to the main menu:
```

Start with the last complete (level 0) backup, loading each volume in order as prompted.

6

10. You are first given another chance to stop:

```
Last chance before scribbling on /dev/name.
```

If you wish to continue, press <Return>. If you wish to stop, press <Del> and the program exits.

## Restoring an Entire Filesystem

11. The restoration process may take some time. When the restore phase of the operation is complete, you see:

```
End of backup
The restore of /name has been successful.
The filesystem will be checked to insure integrity.
```

Next, **sysadmin** runs a check on the filesystem using **fsck(ADM)**. You see a series of messages like this:

```
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Free List
***** FILE SYSTEM WAS MODIFIED *****
```

12. When the filesystem check is complete, you see:

```
You must remount /dev/name when ready to use the filesystem.
```

13. Now that the restore of your level 0 volumes is complete, you are returned to the Filesystem Maintenance Menu. Repeat steps 6-12 until each of your higher level backups (1, 2 and 3 that were done between the last level 0 and the date when your filesystem was damaged) have been restored. Be sure and restore them in the order they were done, or you will overwrite recent versions of files with older ones.

Your filesystem is now completely restored. You need to mount the filesystem before you can use it. To mount it, enter the following command, substituting *name* for the name of your filesystem and *mountpoint* is the directory upon which it is mounted:

```
mount /dev/name /mountpoint
```

---

## Editing `/etc/default/filesys` and `/etc/default/archive`

The files `/etc/default/filesys` and `/etc/default/archive` are used by `sysadmin` to create the filesystem and archive device menus. Even though each menu provides the option “Other” so that you can use filesystems and devices not described, you should keep these default files up to date as your system changes. These files are also used by other programs, and should be maintained as specified for these programs.

### `/etc/default/archive`

The `/etc/default/archive` file contains a complete set of devices supported by XENIX. Each device, filesystem, or drive in these files is represented by a one line entry which consists of “name=value” pairs, separated by spaces or tabs. For example, the following is a possible entry in `/etc/default/archive`:

```
cdev=/dev/rct0 desc="Cartridge Drive (600 ft tape)" \
blocking=20 size=600 density=20000
```

The value part of “name=value” pair “desc=“Cartridge Drive (600 ft tape)”” contains spaces, therefore it must be surrounded by quotes in order to be interpreted correctly.

**6**

If your system does not use certain drives included in `/etc/default/archive`, or if you add an entry to either file, and then later decide that you don’t need it any longer, rather than deleting the entry, you can place a pound sign (#) at the beginning of the first line for that device, and it will be treated as a comment and ignored. Later, if you need the entry again, you can delete the #.

Entries commented out will not appear in the list of media devices displayed in the “Restore backup file(s)” option of the Filesystem Maintenance Menu.

## /etc/default/filesys

The minimum necessary information about a filesystem is:

- A character device name, (cdev=), or a block device name, (bdev=).
- The mount directory, (mountdir=), which is also used by the **mnt(C)** utility, in addition to **sysadmin**. The name *rcmount*, in the example below is used exclusively by **/etc/rc**.

```
bdev=/dev/root cdev=/dev/rroot mountdir=/ \
desc="The Root Filesystem" rcmount=no mount=no
```

```
bdev=/dev/u cdev=/dev/ru mountdir=/u rcmount=yes \
fsckflags=-y desc="The User Filesystem"
```

---

## An Explanation of Backup Levels

The most straightforward and dependable way to ensure the safety of data is to back up everything on a filesystem at one time. However, filesystems can be large (as much as 200 Mbytes or more), and may take hours to back up. The concept of backup levels (or incremental backups) addresses this problem. The general idea of an incremental backup is to back up only those files that have changed since a previous backup. This can significantly reduce the size and duration of the backup. Consider the following scheme:

|         |                                 |
|---------|---------------------------------|
| Monthly | complete backup                 |
| Weekly  | everything newer than last week |
| Daily   | everything newer than yesterday |

This means that at the end of every month, the entire filesystem is backed up. Each week, the files that have changed since last week are backed up, and each day, any files that have changed since yesterday. If at some point a filesystem is damaged, you would simply restore the last full (monthly) backup, the last weekly backup, and any daily backups that happened just prior to the accident. Thus it is always possible to reconstruct a filesystem from a series of backups.

While this is a simple method to understand, the implementation using incremental backup levels is not.

6

## Principles of Incremental Backup Levels

To make the business of backing up files more efficient, the backup facility uses a progressive series of levels, each of which is based on the last occurrence of a lower-level backup.

| Level | Files Saved                             |
|-------|-----------------------------------------|
| 0     | all files on the filesystem             |
| 1     | files changed since last level 0 backup |
| 2     | files changed since last level 1 backup |
| 3     | files changed since last level 2 backup |

## An Explanation of Backup Levels

The levels serve to subdivide a backup into manageable units. It is important to realize that each backup level creates backups based on the previous (next lowest) level backup. This means that the order of the backups is not significant, but the level number is.

For example, let's assume that the following backups were done for a week:

| Day | Level | Files Backed Up                   |
|-----|-------|-----------------------------------|
| Mon | 0     | all files on filesystem           |
| Tue | 2     | all files changed since Monday    |
| Wed | 1     | all files changed since Monday    |
| Thu | 3     | all files changed since Tuesday   |
| Fri | 2     | all files changed since Wednesday |

This example is illogical, but serves to demonstrate how the levels work. Remember that each of the backups saves the files changed since the next lower-level backup, and that level 0 is the lowest. Therefore, the level 2 on Friday backs up all files changed since the next lowest number, level 1, on Wednesday. The level 2 on Tuesday saves only those files that have changed since the day before, since the only previous lower-level backup is a 0. If all the backup levels except Monday were level 2, each would still back up all files that changed since the level 0 on Monday.

## How the Default and Alternate Schedules Work

The *schedule* file provided with your distribution is optimized for use on systems under moderate use (8 to 10 users with total disk storage of 200 to 400 Mbytes). The default schedule for the root filesystem is similar to that used for the */u* filesystem in the alternate schedule. This is done because a system with a single filesystem (root) has active user accounts and should be backed up each day. A system with a second filesystem (*/u*) for user accounts is backed up each day, while the less active root filesystem is backed up every other day. The alternate schedule is shown in Figure 6-8.

|   |            |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |
|---|------------|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|
| # |            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| # | Filesystem | M | T | W | T | F | M | T | W | T | F  | M | T | W | T | F | M | T | W | T | F  |
|   | /dev/rroot | 0 | x | 3 | x | 3 | 2 | x | 3 | x | 3  | 1 | x | 3 | x | 3 | 2 | x | 3 | x | 3  |
|   | /dev/ru    | 3 | 0 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3  | 3 | 1 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3  |

Figure 6-8 The Default Schedule

### The /u Filesystem

Filesystem */dev/u* is a heavily used resource. Some level of backup is performed every day. This scheme is designed to minimize resources while maximizing safety; if one or more of the backups for that week is lost or goes bad, there is sufficient redundancy to minimize any loss of data.

According to the default schedule, a full (level 0) backup of */dev/ru* occurs at the beginning of the month. (Because a level 0 is done on the root filesystem on Monday, the level 0 for */u* is done on Tuesday.) On Wednesday, a level 3 backup saves just those files on */dev/ru* that have changed since the level 0 backup. By the end of the week far fewer floppies or tapes are used than the number needed for full backups each day. Time is substantially reduced as well. If it is necessary to restore the filesystem to the last recorded state, you would restore the last level 0 backup, followed by each of the most recent lower-level backups that were done since.

Note that each Tuesday, a lower-level backup (0, 1 or 2) occurs that saves everything since the beginning of the month and causes each of the level 3 backups that follow it to be based on that week. This way the level 3 backups do not become too large and redundant.

### The root Filesystem

The root filesystem contains the operating system and other system files. It changes less frequently, so it is not backed up every day unless user accounts are located there. Each Monday, a lower-level backup is done, and level 3 backups are done twice per week. Just as with the */u* filesystem, the level 3 backups are restricted to cover only those files that have changed during that week.

### How Backups Restore a Filesystem

For example, assume you have a hardware failure that ruins the information on the hard disk. Assume it happens on the last Thursday of the month, just before the backup was to be done that evening. You fix the hardware problem and reinstall your system, but how do you restore your backups? Restore the last occurrence of each backup level, in ascending order:

- level 0 (done on the first Tuesday of the month)
- level 1 (done on the third Tuesday)
- level 2 (done on the fourth Tuesday)
- level 3 (done on Wednesday evening)

You would not need to restore the level 2 that was done on the second Tuesday, because the level 1 that followed it covered the same files. The only information that is missing is what was changed during the day on Thursday, just before the crash. This is the primary reason for backups; recovery should be straightforward and with a minimum of loss.

## **Chapter 7**

# **Adding Hard Disks**

---

Introduction 7-1

Before You Start 7-3

    Configuring a Hard Disk 7-3

        ST506/ESDI or OMTI Controllers 7-3

        SCSI Disk 7-4

    Preparing the Hardware 7-4

Installing the Hard Disk 7-6

Adding the New Filesystem(s) 7-18

    Moving User Accounts Off the Primary Hard Disk 7-19



---

## Introduction

When your system suffers from chronic lack of space, you probably need to add a hard disk to give the system extra space for storing user files and directories. The following types of secondary hard disks and controllers are supported:

- ST506/ESDI (standard disk support)
- SMS OMTI 8620 or 8627 controller
- SCSI

---

### *Note*

SCSI and OMTI support is not available on XENIX-286 distributions and the **mkdev hd** menus are slightly different.

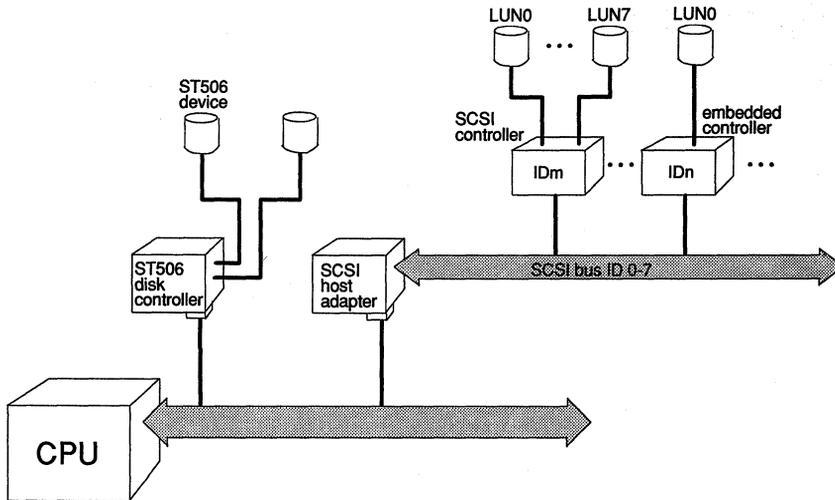
---

Three configurations are possible:

- root disk on a SCSI host adapter with an option to add one SCSI host adapter, each supporting up to seven controllers, and each SCSI controller supporting up to eight devices
- root disk on an ST506/ESDI controller with an option to add one ST506/ESDI controller, each supporting two ST506 or ESDI disks and up to two SCSI host adapters (which can be configured as in the first option)
- root disk on an OMTI controller that supports an additional disk, both of which can be either ESDI or ST506

## Introduction

Figure 7-1 illustrates a configuration of the second type.



**Figure 7-1** ST506/ESDI and SCSI Configuration Examples

A SCSI host adapter (HA) translates signals from the CPU bus to the SCSI bus. A SCSI controller is known by a SCSI ID. A SCSI device is referenced by a logical unit number (LUN).

When you installed the operating system and initialized your root disk, the root disk was configured as the first hard disk on the first controller (for ST506/ESDI or OMTI disks) or first host adapter (for SCSI disks).

Although the basic procedure for adding a disk is common to all types of disks, you occasionally need to perform somewhat different steps based on the type of disk you are installing. Throughout the procedure, the differing steps are clearly indicated.

---

### Note

The SCSI ID is also known as the SCSI controller number or target ID.

---

---

## Before You Start

Before installing an additional hard disk, you must first decide how to configure the disk, then set up and connect the hardware. This section explains the syntax used for the **mkdev hd** command used to configure and add hard disks. You should use this section to determine what command-line options are necessary to configure your disks. When you have chosen the proper syntax, you can proceed to “Installing the Hard Disk,” later in this chapter.

### Configuring a Hard Disk

You need to decide how you want to configure the disk so you can provide that information to the installation utility.

#### ST506/ESDI or OMTI Controllers

To configure an ST506 or ESDI disk with the **mkdev hd** command, you must know which disk controller supports the new disk and whether it is the first or second disk on the controller. The command syntax is:

```
mkdev hd disk controller
```

Numbering of disks and controllers starts at 0. See Table 7.1 .

**Table 7.1**  
**ST506/ESDI and OMTI Commands**

| Controller        | Command             | Disk being added                      |
|-------------------|---------------------|---------------------------------------|
| ST506/ESDI & OMTI | <b>mkdev hd 0 0</b> | first disk on first controller (root) |
|                   | <b>mkdev hd 1 0</b> | second disk on first controller       |
| ST506/ESDI ONLY   | <b>mkdev hd 0 1</b> | first disk on second controller       |
|                   | <b>mkdev hd 1 1</b> | second disk on second controller      |

## Before You Start

### SCSI Disk

SCSI devices are configured differently from standard disk devices. Figure 7-1 shows that SCSI disks are configured according to three values:

- *ID Number*. The number of the controller on the SCSI bus (0-7).
- *Host Adapter Number*. The number of the SCSI adapter card (0 or 1).
- *Logical Unit Number (LUN)*. The number of the device associated with the controller ID (0). This value is meaningless in this context because hard disks have *embedded* controllers; the LUN is always zero and **mkdev hd** does not prompt for this value.

The **mkdev hd** syntax for SCSI disks is as follows:

```
mkdev hd disk host_adapter
```

When a SCSI hard disk is added to a system with a SCSI root disk, *disk* specifies a disk number from 0 to 7; *host\_adapter* specifies the host adapter (0 or 1). For example, to add a second SCSI disk to the first host adapter, you would use:

```
mkdev hd 0 1
```

When a SCSI hard disk is added to a system with an ST506 (standard interface) root disk, to add a disk to the first host adapter, *host\_adapter* must be "2"; to add a disk to the second host adapter, *host\_adapter* must be "3". (In this configuration, "0" and "1" are reserved for ST506 controllers.) For example, to add the first SCSI disk to the first host adapter, you would use:

```
mkdev hd 0 2
```

## Preparing the Hardware

Hard disks that do not have matching entries in your computer's BIOS ROM tables are supported through software. When adding secondary hardware, you must change some of the switch settings on the host adapter, SCSI ID, and disk. "SCSI Guidelines" in the *Release Notes* explains what these settings should be. Check the hardware manual for your hard disk drive and the computer for instructions.

When you change the settings on a SCSI device with an embedded controller, remember to use the SCSI ID number, not the LUN. The LUN on an embedded controller is 0, because it is the first and only device on the controller.

Before adding the new disk, you must know how to connect it to the computer. Connecting the hard disk is explained in the hardware manuals provided with the disk and host adapter.

Make sure the additional drive is formatted and verified, and also passes the manufacturer diagnostics before installing the system. If it does not pass the diagnostic tests, you cannot use it with your system.

---

# Installing the Hard Disk

These are the steps to install another hard disk with one XENIX filesystem and no DOS area:

1. After you have connected the hard disk and booted the system, enter system maintenance mode and use the appropriate form of the **mkdev** command, specifying the required configuration information on the command line:

**mkdev hd** *disk controller or host\_adapter*

△ **sysadmsh** users select: System→Add→HardDisk

---

*Note*

If you have a XENIX-286 distribution, the displays in steps 2 through 4 do not appear; skip to step 5.

- 
2. You see:

```
XENIX System V Hard Disk Initialization
```

```
Note: This kernel is configured to support 6 hard disks.
If your system will have more than 6 disks,
the kernel must be reconfigured.
Please press <RETURN>
```

Press <Return> as instructed.

3. Next, the disk controller menu is displayed:

```
XENIX System V Hard Disk Installation
```

```
What type of disk controller will be supporting this disk?
```

1. ST506/ESDI (standard disk support)
2. OMTI 8620 or OMTI 8627
3. SCSI

```
Enter 1, 2, 3 or enter q to quit:
```

Enter the number that corresponds to the controller you have.

4. If you are adding an OMTI controller, an additional prompt appears:

```
What interface will this hard disk use?
```

- 1) ESDI
- 2) ST506

```
Enter 1 or 2 or enter q to quit:
```

Enter the number that corresponds to the type of interface the hard disk will use.

5. If you are adding a SCSI disk and you did not specify the ID on the **mkdev hd** command line, you see:

```
What ID is this disk? Choose a number between 0 and 7:
```

Supply the ID number as determined earlier under "Configuring a Hard Disk."

6. The following prompt is displayed:

```
During installation you may choose to overwrite all
or part of the present contents of your hard disk.
Do you wish to continue? (y/n)
```

Enter **y** and press **(Return)**.

7. If you have a SCSI controller, you see the following message:

The hard disk installation program will now invoke `/etc/fdisk`.  
Entering 'q' at the following menu will exit `/etc/fdisk`.  
and the hard disk installation will continue.

If you wish to exit the entire installation at this menu,  
press the <DEL> key.

Skip to step 13.

---

### Note

The SCSI installation skips steps 8-12.

---

8. If you have an ST506/ESDI (standard interface) controller, you see the following message and prompt:

The hard disk installation will now invoke `/etc/dkinit`.  
Entering 'q' at the following menu will exit `/etc/dkinit`,  
and the hard disk installation will continue.

If you wish to exit the entire installation at this menu,  
press the <DEL> key.

Hard Disk Drive 1 Configuration

1. Display current disk parameters
2. Modify current disk parameters
3. Select default disk parameters

Enter an option or 'q' to quit:

9. If you have an OMTI controller, you see the following additional message:

Caution: Consult the ESDI installation Release Notes if  
you wish to modify the disk parameters the `/etc/default`  
will display.

Read the section "OMTI Guidelines" in your *Release Notes*.

If you enter **q**, you see the following message:

```
The hard disk installation program will now invoke
two disk preparation utilities: fdisk and badtrk.
Selecting 'q' at the main menu for each utility
will exit that utility and continue with the hard
disk installation.
```

Skip to step 13.

10. The **dkinit** menu is intended for unusual or nonstandard disks. If you have a standard hard disk, one that is supported by your computer hardware or special motherboard ROM, enter **3** followed by **<Return>** to select the default parameters. You should then enter **q** and press **<Return>** to continue the installation; skip to step 13.

---

### Note

If you are not sure if your disk is non-standard, check the default parameters using option **1** of the **dkinit** menu. Calculate the size of your disk in bytes using the following calculation:

$size = cylinders \times heads \times sectors/track \times 512$

In addition, some drives are sold by formatted size, others by unformatted size. The formatted size of a drive is approximately 85% of its unformatted size. Note that the parameters displayed by **dkinit** may not match the drive manufacturer's documentation. Some controllers have optional "translation," "mapping," or "63-sector" modes. If one of those modes was chosen during low-level formatting, XENIX must be initialized with the translated parameters and not those of the physical drive. In all cases, the known size of the drive should approximately match the size calculated above from the disk parameters.

---

If your disk is nonstandard, you must enter information that overrides the ROM disk configuration information, replacing it with the new information. If you are unsure of what parameters to enter for your nonstandard disk, contact your disk manufacturer for this information. The **dkinit** program (called during installation) uses BIOS parameters that should be documented in the hardware manual for your computer.

If you enter **1** or **2**, you see the following display:

| Disk Parameters  | Values       |
|------------------|--------------|
| 1. Cylinders     | <i>value</i> |
| 2. Heads         | <i>value</i> |
| 3. Write Reduce  | <i>value</i> |
| 4. Write Precomp | <i>value</i> |
| 5. Ecc           | <i>value</i> |
| 6. Control       | <i>value</i> |
| 7. Landing Zone  | <i>value</i> |
| 8. Sectors/track | <i>value</i> |

In the actual display, *value* is replaced with the default value for that variable.

---

*Note*

The “Cylinders” value refers to the number of cylinders on the entire hard disk and should not to be confused with the number of cylinders you allocated (or intend to allocate) to a given partition.

---

If you entered a **1**, you now see the first menu again. If you entered a **2**, you are now prompted:

Enter a parameter to modify or 'q' to return to the main menu:

11. Enter any number from 1 to 8 to change the disk parameters, or **q** to return to the previous menu. You see the following:

Enter the new value or <RETURN> to use the existing value:

If you wish to change the value, enter a new value now or press <Return> to use the existing value.

12. After you finish changing the disk parameters, enter **q** to return to the main menu. Next, enter **q** again to save the changes you made. Exiting from **dkinit** by entering **q** overwrites any parameters you changed with the new values. If you wish to restore the default parameters after making modifications, enter **3** from the first menu.
  
13. The installation program next runs the **fdisk(ADM)** utility to partition the hard disk. You can partition your disk to also support DOS on the same hard disk (if you have DOS already installed), or you can use the whole disk for your XENIX system. Be sure you follow the guidelines for DOS partitions found in the "Before You Start" chapter of the *XENIX Installation Guide*.

After a moment, the **fdisk** menu appears on the screen. You see this option list:

- ```

1. Display Partition Table
2. Use Entire Disk for XENIX
3. Create XENIX Partition
4. Activate Partition
5. Delete Partition
    
```

Enter your choice or 'q' to quit:

Select option **1** and press (Return).

If you have never installed an operating system on your disk, you see a table similar to this:

Current Hard Disk Drive: /dev/rdisk/1s0

Partition	Status	Type	Start	End	Size

Total disk size: 1220 tracks (5 reserved for masterboot and diagnostics)

Press <RETURN> to continue

If you have previously installed an operating system on your disk, the **fdisk** table is filled in. DOS is usually displayed as partition number 4.

14. Press `<Return>` to return to the main **fdisk** menu. If you would like the XENIX partition to occupy the whole disk, select option **2**. After you have made your selection, quit out of the **fdisk** menu by entering **q**. If any other operating systems were previously installed on your system, you also see the following warning message:

```
Warning! All data on your disk will be lost!  
Do you wish to continue? (y/n)
```

Enter **y** and press `<Return>` only if you want your XENIX system to occupy the whole disk. This ensures that **fdisk** partitions the whole disk.

Note

Most computers have diagnostic programs that write to the last cylinder of the hard disk. This means that the last cylinder should not be allocated to a partition. The last cylinder is not allocated when you choose option **2** from the **fdisk** menu. If you choose option **3**, you should not allocate the last cylinder of the hard disk to the XENIX partition.

-
15. Press `<Return>`, and you see the main **fdisk** menu. You have now set up the partition(s) on your hard disk. To continue with the next step in the installation procedure, enter **q** and press `<Return>`.

If you have an ST506/ESDI controller, continue with step 16.

If you have a SCSI or OMTI controller, skip to step 24.

Note

Bad tracks are handled automatically by SCSI and OMTI controllers and thus steps 16-23 are omitted.

16. Now you see a menu from the program **badtrk(ADM)**. With the **badtrk** program, you can scan your hard disk for defective tracks. The program maps any flawed locations to good tracks elsewhere on the disk. It also creates a bad track table, which is a list of all the bad tracks on your hard disk.

The main **badtrk** menu looks like this:

```
1. Print Current Bad Track Table
2. Scan Disk (You may choose Read-Only or Destructive later)
3. Add Entries to Current Bad Track Table by Cylinder/Head Number
4. Add Entries to Current Bad Track Table by Sector Number
5. Delete Entries Individually from Current Bad Track Table
6. Delete All Entries from Bad Track Table
```

Please enter your choice or 'q' to quit:

Enter **2**, then press **<Return>**.

17. You see the following submenu:

```
1. Scan entire XENIX partition
2. Scan a specified range of tracks
3. Scan a specified filesystem
```

Please enter your choice or 'q' to quit:

Select option **1**.

18. After you select the area you want scanned, you are given the following choices:

```
1. Quick scan (approximately 7 megabytes/min)
2. Thorough scan (approximately 1 megabyte/min)
```

Please enter your choice or 'q' to quit:

Select option **2**.

19. You are prompted:

```
Do you want this to be a destructive scan? (y/n)
```

Enter **y**. You are warned:

This will destroy the present contents of the region you are scanning.
Do you wish to continue? (y/n)

Enter **y** and press **<Return>**. You see the following message:

Scanning in progress, press 'q' to interrupt at any time.
Destructively scanning track x/y, z% of scan completed

- 20. After you respond to the above prompts, the program scans the active partition of the new disk for flaws. The larger your disk, the longer the scanning process takes, so a very large disk may take a while.

As **badtrk** scans the disk, it displays the number of each track it examines, and the percentage of the disk already scanned. Pressing the **q** key at any time interrupts the scan. If you press **q** to interrupt the scan, you do not need to press **<Return>**. You are then prompted to continue scanning or to return to the main menu.

Whenever **badtrk** finds a defective track, it lists the location of that track using both the sector number and cylinder or head conventions. Defective track information is entered into the table and displayed on the screen. Here is an example of a bad track:

wd: ERROR : on fixed disk ctlr=0 dev=0/47 block=31434 cmd=00000020
status=00005180, sector = 62899, cylinder/head = 483/4

- 21. When the scan is complete, the menu reappears. Select option **1** to see the results of the scan. Your bad track table looks something like this:

Defective Tracks

	Cylinder	Head	Sector Number (s)
1.	190	3	12971-12987

Press **<RETURN>** to continue

Press **<Return>** to return to the main menu.

Note

If there is a flaw in the first few tracks of the XENIX partition, you are returned to the **fdisk** utility (see the previous installation step). Repartition the disk with **fdisk** so that the XENIX partition no longer includes the defective tracks. You have to experiment to determine how many tracks to exclude. Leave these defective tracks unassigned to any operating system. When you leave **fdisk**, **badtrk** is run again and you should scan the disk for further flaws.

This process continues until **badtrk** finds no flaws in the first few tracks.

22. To exit **badtrk**, enter **q** and press <Return>.
23. You are next prompted for the number of tracks to allocate as replacements for those tracks that are flawed. You should allocate at least as many as the recommended number. Enter the number or just press <Return> to use the recommended number that is displayed:

Enter the number of bad tracks to allocate space for
(or press <RETURN> to use the recommended value of *n*):

If you press <Return> and do not enter an alternate value, **badtrk** allocates the recommended number of tracks as replacements. This number is based on the number of bad tracks currently in the table, plus an allowance for tracks that may go bad in the future. If you ever exceed the number of allocated bad tracks, you must reinstall the system. Next, you see a prompt from **divvy(ADM)**. The **divvy** program divides a partition into filesystems. You can create up to seven divisions on a single partition, and name them anything you like.

Note

If possible, try to limit your filesystems to 60-80 megabytes. System maintenance tools work faster and more efficiently on this size filesystem. Remember that a block is 1024 bytes. To convert blocks to megabytes, simply divide by 1000.

24. You see the main **divvy** menu and a display that shows how your disk is divided similar to the one below:

Name	New File System?	#	First Block	Last Block
	no	0	0	25302
	no	1	-	-
	no	2	-	-
	no	3	-	-
	no	4	-	-
	no	5	-	-
	no	6	-	-
hd1a	no,exists	7	0	25546

25303 blocks for divisions, 244 blocks reserved for the system

n[ame] Name or rename a division.
c[reate] Create a new filesystem on this division.
p[revent] Prevent a new filesystem from being created...
s[start] Start a division on a different block.
e[nd] End a division on a different block.
r[estore] Restore the original division table.

Please enter your choice or 'q' to quit:

Each row in the **divvy** table corresponds to a filesystem (also known as a division).

You can divide the partition into as many as seven filesystems. Each filesystem must have the following:

- if the division does not already exist it must first be created using the **c** (create) command.
- a beginning block number, defined by the **s** (start) command.

- an ending block number, defined by the **e** (end) command.
- a filesystem name, defined by the **n** (name) command. Filesystems can have any name you choose. For example, you could name a filesystem *u* (for “user”). This name is in turn used to create the device name (for example, */dev/u*).

Warning

Note that filesystem boundaries must not overlap. For example, filesystem 0 cannot end on the block number where filesystem 1 begins.

Do not change the configuration of filesystem 7; it is reserved for internal use by the operating system.

Do not name a filesystem *usr*; this directory already exists on the root filesystem.

Exit from **divvy** by entering **q**. The program prompts whether to install the new partition table, return to the main menu, or exit the program without installing the partition table. Select option **i** to install the partition table.

For more information, see the **divvy(ADM)** manual page.

25. The system now creates the filesystems on your hard disk. This takes several minutes. You see the following message:

```
Making filesystems
```

26. After creating the new filesystem(s), **mkdev hd** terminates. To make the filesystem(s) accessible, you must follow the instructions in the next section “Adding the New Filesystem(s).”

Adding the New Filesystem(s)

Before leaving system maintenance mode, you must add the new filesystem to the system. The `divvy n[ame]` menu option created the device node for the filesystem (for example: `/dev/u`). The next step is to create the filesystem itself. Using the filesystem `/u` as an example, this is how the actual filesystem is created:

```
mkdev fs /dev/u /u
```

This command does the following:

- Creates a directory `/u` (also known as the mount point).
- Creates the *lost+found* directory for `fsck(ADM)`. Files are then created in the *lost+found* directory, then removed. This allocates inodes for the directory, so that if the filesystem is corrupted and runs out of inodes, `fsck` is still able to recover files.
- Mounts the device (`/dev/u`) on `/u`, and cleans it using `fsck`.
- Removes write permissions on the directory `/u` for group and all other users except root.
- Adds the following line to `/etc/checklist`:

```
/dev/u
```

- Prompts you to decide if the new filesystem is to be automatically mounted, checked, and cleaned at boot time or not.
- Unmounts the filesystem.

Note

Note that your new filesystem does not have to be called `/u`. You can name your filesystems and mount points as you choose. It is recommended that mount points and the corresponding filesystems have the same name to avoid confusion.

To mount or unmount */dev/u* on */u*, use the following two commands respectively:

```
mount /dev/u /u
```

△ **sysadmsh** users select: Filesystems→Mount

```
umount /dev/u
```

△ **sysadmsh** users select: Filesystems→Unmount

Only the super-user can use the **mount** command. The system administrator can permit users to mount specific filesystems (with or without password protection). For more information, refer to “Permitting Users to Mount Filesystems” in the “Maintaining System Security” in this guide.

Moving User Accounts Off the Primary Hard Disk

You can access the files on your new filesystem by first mounting it in the appropriate directory (we used */u* as an example in the preceding section). After you mount the filesystem, all directories and files on it are usable just as any others on the system.

Extending the example set forth in the previous sections, let’s assume that you want to move your user accounts to the new */u* filesystem on your secondary hard disk. The first thing to do is to ensure that any new accounts you add to the system will be placed in the new location. The **mkuser(ADM)** command (used to create new user accounts) reads the default location for user accounts from the file */etc/default/mkuser*. In particular, the variable “HOME” must be changed to reflect the new location.

Edit the file */etc/default/mkuser*. As distributed, the file has an entry that looks like this:

```
HOME=/usr
```

If you wish to use */u*, change the entry accordingly. Now, whenever you run the **mkuser** command to add a new user, that user account will be in */u*. Make certain that the */u* filesystem is mounted before you run the **mkuser** script, or the new user’s directory will not be accessible when */u* is mounted.

If there are already users on the system, and you want to move their accounts to the new filesystem, you can use the **copy(C)** command to

Adding the New Filesystem(s)

copy their accounts to the new filesystem. You must also change the users' entries in */etc/passwd* to reflect the new pathnames of their home directories.

Follow these steps to move user accounts from one filesystem to another:

1. Make sure the new filesystem is mounted and the account in question is not being used. Also, be sure you are logged in as root.
2. Change directories to the top of the current user account directory. If, for example, the user accounts are in */usr*, enter:

```
cd /usr
```

and press **<Return>**.

3. List the contents of this directory:

```
ls
```

In addition to the system directories such as *bin*, *lib*, and so forth, you see a list of account names, for example:

```
gwenl  anaf      julies  billbr
mattb  dariaw    zursch  kamk
banta  jamescb  pj      landy
```

4. Copy each of the user directories as follows:

```
copy -orm name /u/name
```

and press **<Return>**.

5. When finished, enter:

```
cd /u
```

and press **<Return>**. List the new contents of */u* to make sure all of the accounts have been copied correctly.

6. After you are sure that all of the accounts have been completely copied, you can remove the user accounts in the previous user filesystem by removing files, directories and aliases to that location.

7. Change the home directory for each user as listed in */etc/passwd*. An example entry in */etc/passwd* might be:

```
alb:CoHiKNs.:271:104:Al Berry:/usr/alb:/bin/csh
```

You see one such line for every user on your system. Change the field:

```
:/usr/alb:
```

to match the user's new home directory:

```
:/u/alb:
```

Do this for every user whose home directory has changed.

Chapter 8

Tuning System Performance

- Introduction 8-1
 - Kernel Parameters 8-1

- Reallocating Kernel Resources with `configure` 8-3
 - Using the `configure` Command Line 8-5

- Reconfiguring Because of Persistent Error Messages 8-6

- Reconfiguring for Performance 8-7
 - Tradeoffs in Kernel Tuning 8-7
 - Common Resource Needs 8-7
 - Improving Disk Utilization 8-8
 - Sizing the Buffer Cache 8-8
 - Buffer Allocation for XENIX-286 8-9
 - Setting the “Sticky-Bit” (XENIX-286 Only) 8-10
 - Adding Memory (RAM) 8-10
 - Reorganizing Filesystems 8-11
 - SCSI Scatter/Gather I/O 8-11

- Defining Efficient System Usage Patterns 8-12
 - Checking Process Activity with `ps` 8-12
 - Checking User `$PATH` Variables 8-13

- Using `vmstat` to Diagnose System Inefficiency 8-14
 - Memory Usage: Buffers Versus Pages 8-14
 - The `vmstat` Display 8-15
 - Checking Buffer and Page Cache Usage 8-16
 - Checking CPU Usage 8-16

- Summary of Tunable Parameters 8-17
 - Disks and Buffers 8-20
 - Character Buffers 8-22
 - Files, Inodes, and Filesystems 8-22
 - Processes, Memory Management and Swapping 8-23
 - Clock 8-25
 - MultiScreens 8-25
 - Message Queues 8-26
 - Semaphores 8-27

Shared Data 8-28
System Name 8-28
Streams Data 8-28
Event Queues and Devices 8-31
Hardware Dependent Parameters 8-31

Introduction

Your XENIX system is optimized for use with a variety of hardware configurations and as a platform for many applications. The kernel, which lies at the heart of the operating system, controls a number of resources that are constantly being used, released, and recycled. These resources include:

- | | |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| buffers | A cache of in-memory storage units that hold recently used data. (Buffers increase efficiency by keeping this data on hand and decrease disk access.) |
| table entries | A space in system tables that the kernel uses to keep track of current tasks, resources, and events. |
| other parameters | These are other definable values that govern special resources (such as the number of multiscreens available or the quantity of semaphores). |

The use of these resources is defined by certain limits known as *tunable kernel parameters*. These limits can be decreased or extended, sometimes at the expense of other resources. Each resource or limit is represented by a separate kernel parameter. Deciding how to best optimize the use of these resources is known as performance or kernel tuning. This chapter explains how to change these parameters to suit the needs of your system. In addition, general procedures are included that can improve resource usage and system performance.

Kernel Parameters

Kernel parameters are values contained in the XENIX system kernel, which is the core of the operating system. Each time these tunable parameters are changed, their new values are relinked (recompiled) into the kernel so that the new limits will take effect. The **configure(ADM)** utility changes the value of kernel parameters.

Performance tuning is an activity that may need your attention when you first set up your XENIX system. When you bring the system up for the first time, the system is automatically set to a basic configuration that is satisfactory for most sites. This configuration, however, cannot take into account the usage patterns and the behavior of your particular applications. For this reason, the structure of the system allows you to reconfigure it to enhance the performance for your particular application over that of the standard configuration.

Note

We do not recommend adjusting kernel parameters if there is no apparent need to do so.

There are several reasons for reallocating system resources:

- You install additional hardware memory and thus have greater memory resources to allocate.
- Persistent error messages are displayed indicating that certain resources are used up, such as inodes or table entries.
- The system response time is consistently slow, indicating that other resources are too constrained for the system to operate efficiently (as when too little hardware memory is installed).
- Resource usage needs to be tailored to meet the needs of a particular application.

In addition, it is important to determine which resources are being wasted or are inefficiently distributed. Certain tunable parameters are normally adjusted upward when additional memory is installed to allow the system to support more users. However, for a computer used as a high-powered personal computer, or dedicated processor, it may not be necessary to increase kernel tunable parameters when additional memory is installed. In fact, tuning certain parameters normally associated with adding memory to support more users (NBUF, NCLIST, and so on) can actually decrease overall performance. This is because these parameters increase kernel data space requirements, thus making less of the new memory available for user processes. Simply stated, the intended use of your computer and your observations on how well it is performing should be used as a guide in determining the need to adjust tunable parameters.

Specialized applications often require the reallocation of key system resources for optimum performance. For example, users with large databases may find that they need to lock more files simultaneously than the current allocation of file locks permit. Users who have no need for specialized features such as message handling may find that they can get a slight performance boost by deallocating those features.

Reallocating Kernel Resources with `configure`

The `configure` utility is a menu-driven program that presents each resource and prompts for modification.

Note

This section only describes how a parameter change is made; you must read the rest of the chapter to understand why and when a change must be made.

To change any kernel parameter, do the following:

1. Reboot and enter single-user (maintenance) mode.
2. Use `custom(ADM)` to determine if the link kit package (LINK) is installed. If not, use `custom` to install it.
3. After making certain the Link Kit is installed, enter the following commands:

```
cd /usr/sys/conf
./configure
```

4. The `configure` menu is displayed:

1. Disks and Buffers
2. Character Buffers
3. Files, Inodes, and Filesystems
4. Processes, Memory Management and Swapping
5. Clock
6. MultiScreens
7. Message Queues
8. Semaphores
9. Shared Data
10. System Name
11. Streams Data
12. Event Queues and Devices
13. Hardware Dependent Parameters

Select a parameter category to reconfigure by typing a number from 1 to 14, or type 'q' to quit:

The parameters are grouped by category. To locate a parameter, see "Tunable System Parameter Descriptions" at the end of this chapter. Choose a category by entering the number preceding it. The resources in that category are displayed, one by one, each with its current value. Enter a new value for the resource, or to retain the current value, simply press `<Return>`. After all the resources in the category are displayed, `configure` returns to the category menu prompt. Choose another category to reconfigure or exit `configure` by entering `q`.

Note

Note that you must have the package associated with the parameter to make use of the resource associated with it. For example, the `STREAMS` package must be installed for the `STREAMS` parameters to have effect.

-
5. After you finish changing parameters, you must link them into a new kernel. Enter the following commands:

```
./link_xenix  
./hdinstall
```

This assembles each of the kernel modules into a new kernel, which is then moved into place by the `hdinstall(ADM)` command. Linking can take a while.

6. Boot the new kernel with the following command:

```
/etc/shutdown
```

```
Δ sysadmsh users select: System→Halt
```

A boot prompt appears. When you press (Return) to reboot the system, the new kernel is loaded and run.

If problems exist with the new kernel, reboot */xenix.old* by entering **xenix.old** at the Boot: prompt.

Using the `configure` Command Line

configure also has a command-line interface suitable for use by application developers. For instance, a database developer who finds that 70 files rather than 50 files need to be locked simultaneously may provide a shell script to perform the reconfiguration. To find the current value of any configurable resource using the command-line interface, enter:

```
./configure -y RESOURCE
```

where *RESOURCE* is the name of the tunable parameter (in uppercase). To change the value of any resource from the command line, enter:

```
./configure RESOURCE=value
```

This interface is in addition to the interactive one; the same resources are configurable from both interfaces.

The sections that follow describe scenarios for reconfiguring the kernel resources.

Reconfiguring Because of Persistent Error Messages

There are cases when the operating system advises you that system limits are being exceeded. These messages are displayed on the console. Some of the messages are advisory only. Others precede a system panic in which case additional diagnostic messages are printed, and the system “hangs,” requiring you to reboot. The kernel should not be reconfigured because a kernel error message was received once, or even a couple of times, but when a single message persists between system sessions.

If you encounter any of the messages listed in Table 8.1, refer to the descriptions for the appropriate parameters in the section “Tunable System Parameter Descriptions” later in this chapter. If you need to adjust a resource, first try to increase the value by a small amount. If the problem persists, increase it by 100 percent or more of its original value. If the problem is still not solved, more detailed research is required to locate the exact program and sequence that cause the error.

Table 8.1
Error Messages and Associated Tunable Parameters

Kernel Error Message	Parameter
inode table overflow	NINODE
Timeout table overflow	NCALL
File table overflow	NFILE
stropen: out of streams	NSTREAM
stropen: out of queues	NQUEUE

Reconfiguring for Performance

The system is configured so that the greatest quantities of kernel resources are assigned to the most common tasks (such as reading and writing from the disk), without ignoring the more specialized features (such as interprocess communication). This balance can be shifted to conform to individual requirements using the information described in the sections that follow.

Tradeoffs in Kernel Tuning

Systems can support very different usage of resources. A system that supports several users editing small files consumes different resources than a single-user system running a large database. Parameters are adjusted to allow the kernel to operate more efficiently. This often increases the size of kernel data structures. Although this might make one aspect of system operation more efficient, the kernel takes longer to scan larger structures. This implies that increasing certain parameters unnecessarily can actually slow the system down. For example, increasing the parameter NPROC allows the system to maintain a larger list (PROCCess table) of active processes. This can have an adverse effect on the kernel scheduler because it now must repeatedly scan this larger table every time it checks to see which process to run next. Additionally, because the kernel data space requirements increase when table sizes are increased, there is less memory space available for user processes, which can also lower overall performance.

Common Resource Needs

Often your system usage presents you with the need to tune certain parameters for particular circumstances. Refer to the section “Tunable System Parameter Descriptions” in this chapter for details on each parameter before you make any alterations.

Table 8.2
Special Case Tuning Needs

Desired Improvement	Parameters
Improve system performance when additional memory is installed.	NBUF, NHBUF (see "Sizing the Buffer Cache")
Increase system limits when additional memory is installed (to support more users and reduce chances of system problems at times of heavy load, and so forth).	NCALL, NINODE, NFILE, NPROC, NCLIST (also see message, semaphore, and shared memory parameters)
Each user needs to run more processes.	MAXUPRC

Improving Disk Utilization

Disk input and output can cause a bottleneck in system performance. There are certain considerations in tuning the disk subsystem for better utilization.

- Choosing the proper number of buffers
- Setting the sticky bit on selected programs (XENIX-286 only)
- Adding more memory
- Organizing the filesystems to minimize disk activity
- Enabling SCSI scatter/gather (XENIX-386 only with host adapters that support it)

Sizing the Buffer Cache

The system effectively divides available memory between two structures: the disk buffer cache and the page cache. The buffer cache is a series of buffers that hold recently used data in case it is needed again. If a read or a write can be satisfied using the buffer cache instead of the disk, system performance improves because memory operations are much faster than disk operations. The page cache is similar in concept to the buffer cache; disk buffers contain data while pages contain programs. (See the "Using vmstat to Diagnose System Inefficiency" section for an explanation of the page and buffer caches.)

When the **configure** “Disk Buffer” category is selected, the buffer resources NBUF, NSABUF (XENIX-286 only), NHBUF and MAXBUF are displayed in turn.

Note

Buffer allocation differs in XENIX-286; refer to “Buffer Allocation for XENIX-286.”

The NBUF parameter specifies the number of buffers in the system buffer cache. NHBUF specifies the number of hash queues in the buffer cache. Rather than search the entire pool of buffers, the buffer cache is broken into a series of queues that are organized (hashed) by device and block number. The more buffers, the greater chance that data can be found in the buffers without the system having to do a time-consuming disk read. If excessive swapping is evident, you can reduce the number of buffers, which increases the size of the page cache by making more memory available. We also recommend adding as much RAM as practical; swapping is decreased and performance is improved. The value for NHBUF must be a power of 2; in addition, NBUF divided by NHBUF must be approximately 4. After your XENIX system has run for a day or so, you will want to check for excessive swapping activity. If such activity is found, reduce the number of buffers (NBUF and NHBUF) or increase your system memory.

Buffer Allocation for XENIX-286

On 80286-based machines, there is a distinction made between near and far kernel data. NSABUF determines the number of near disk buffers and NBUF determines the number of far disk buffers. (This distinction is meaningless on 80386-based machines and the parameter NSABUF is not used.) The quantity of System Addressable Buffers (NSABUF), represents a tradeoff between buffers and other kernel resources.

A small amount of kernel resource space is available for additional device drivers, but it can be allocated to NSABUF, if desired. Beyond this, NSABUF should only be increased if other kernel resources are not being used.

The basic scheme is to deallocate any unused resources, set MAXBUF and NBUF to equal, large numbers, and after finding an appropriate tradeoff between user memory and external disk buffers, use any remaining near data space space for increasing NSABUF and the few extra buffer headers that the additional buffers will require.

Reconfiguring for Performance

Under XENIX-286, if combined allocated resources consume more than 64K of memory, the `ld` error message:

```
Group "DGROUP" larger than 64Kbytes
```

appears when the shell script `link_xenix` is run. This message does not apply to SCO XENIX-386. If you see this message, you should reduce some of the configurable kernel parameters discussed in this chapter.

Setting the "Sticky-Bit" (XENIX-286 Only)

Setting the sticky bit can reduce the disk traffic of a select group of commands. The term "sticky" refers to making a program stay, or "stick" in memory. The text segments of selected sticky commands are kept in memory or contiguously in swap space, even when the process terminates.

Note

The sticky-bit is only useful on 80286-based machines. With 80386-based machines, recently used pages (with freshly initialized data) are held in the page cache, so there is no need to set the sticky bit. If the bit is set anyway, it is ignored.

The sticky bit is normally useful for very frequently used programs like `ls`, `sh`, and `vi`. Any program used often enough can have the sticky bit set for a significant performance boost. The sticky bit can be set by entering the following command:

```
chmod u+t filename
```

Adding Memory (RAM)

Adding additional memory does not necessarily mean that the kernel needs retuning. Keeping system limits at the default value may deliver optimum performance even when additional memory is installed. If you are installing more memory, see "Adding Multiport Cards, Memory, and Other Bus Cards" in this guide.

Reorganizing Filesystems

As filesystems are used, the blocks of individual member files tend to become physically scattered around the disk(s) and I/O becomes less efficient. This scattering yields poor ordering of blocks within files and poor directory structure. Directories also tend to grow large and increase search time. These problems increase file access overhead and are discussed in “Maintaining Efficient Filesystem Organization” in the “Managing Filesystems” chapter in this guide.

SCSI Scatter/Gather I/O

The SCSI driver includes support for scatter/gather. This is a scheme wherein groups of disk I/O requests are gathered together rather than executed singly. This process increases overall disk efficiency.

To enable scatter-gather support, run the **configure(ADM)** utility and select “13. Hardware Dependent Parameters” and change the **AD_SCATTER** parameter to 1. The boot display indicates whether or not scatter/gather is enabled. A sample follows:

```
%disk 0x330-0x332 11 5 type=ASGE00 unit=0 cyls=512 hds=64 secs=32
```

The characters following “type= ” indicate whether scatter/gather is enabled or disabled. If the string of characters is “ASGE,” then scatter/gather is enabled. If it is “ASGD,” then scatter/gather is disabled. The scatter/gather feature is only supported with certain host adapters; consult your *Release Notes* for details.

Defining Efficient System Usage Patterns

After the kernel and the system activities are tuned, and the filesystems organized, the next step for improving system performance is to perform some housekeeping activities and to check whether prime-time load can be reduced. The person responsible for administering the system should check for the following:

- less important (or even unnecessary) jobs interfering with more important jobs
- the efficiency of user-defined features, such as *.profile* and *\$PATH*

Checking Process Activity with ps

The **ps(C)** command obtains information about active processes. This command gives a “snapshot” picture of what processes are executing, which is useful when you are trying to identify what processes are loading the system. Things will probably change by the time the output appears; however, the entries that you should be interested in are **TIME** (minutes and seconds of CPU time used by processes) and **STIME** (time when process first started). Figure 8-1 contains sample output from the **ps -afe** command.

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
root	22247	21299	0	16:54:41	T1	0:01	msscreen -n 2
johnson	22246	22079	0	16:52:53	T0	0:04	vi file2
root	22285	22247	0	16:56:04	T1	0:06	msscreen -n 2
root	22284	22247	0	16:56:04	T1	0:00	msscreen -n 2
markham	22274	22271	0	16:55:09	p1	1:05	rlogin colossus
markham	22271	22243	0	16:55:07	p1	1:05	rlogin colossus
forbin	22304	21003	0	17:02:51	003	0:00	/usr/bin/mail kuprin
fisher	22298	18505	2	17:02:29	011	3:28	nethack
root	22305	1327	10	17:02:57	T2	0:00	ps -afe

Figure 8-1 Sample Output from **ps -afe**

The “Troubleshooting Your System” chapter explains how to use the **ps** command to perform such tasks as locating “runaway” processes (one that uses progressively more system resources over a period of time while

you are monitoring it). You can also use **ps** to find processes that take a very long time to execute; you can consider using **cron(C)** to execute the such jobs during off-hours. “Using the Job Scheduling Commands: at, cron, and batch” in this guide explains how to do this.

Checking User \$PATH Variables

The **\$PATH** environment variable lists the pathnames of all directories to be searched each time a command is executed. **\$PATH** is defined in a user’s *.login*, *.cshrc*, or *.profile* file. (See **environ(M)** for more information on environment variables.) Before displaying “not found,” the system must search every directory in **\$PATH**. These searches require both processor and disk time. If there is a disk or processor bottleneck, changes here can help performance.

Some things that you should check for in user **\$PATH** variables are:

- path efficiency
\$PATH is read left to right, so the most likely places to find the command should be first in the path (**/bin** and **/usr/bin**). Make sure that a directory does not appear twice in **\$PATH**.
- path length
In general, **\$PATH** should have as few entries as possible.
- large directory searches
Searches of large directories should be avoided if possible. Put any large directories at the end of **\$PATH**.

Using vmstat to Diagnose System Inefficiency

The **vmstat** utility can be a useful tool for examining and tuning system performance. It cannot provide definitive answers on resource tuning issues, but can provide some insight into the internal workings of the system and help diagnose problems related to poor memory and CPU usage.

Note

The **vmstat** utility is only applicable to 80386-based machines and is not available on XENIX-286 distributions.

Memory Usage: Buffers Versus Pages

Specifying the number of system disk buffers (either explicitly using **configure** or implicitly using the defaults), effectively divides the available memory between two pools: the disk buffer pool and the page pool. The page pool contains the programs being run and cached copies of recently-used program pages. If the page pool is much too small for the load imposed on the system, the system will be constantly swapping pages in and out just to keep up with the current processes.

If the page pool is only slightly undersized, the effects will be seen not in swapping overhead but in reduced cache performance when running the same programs repeatedly. This means that sufficient pages are available to effectively handle current processes, but there are none to spare for keeping recently-used pages in memory for potential access savings.

By using **vmstat**, you can determine how many programs have been swapped in and out during a given interval. If excessive swapping is evident, you can redistribute memory allocation to increase the page pool. We also recommend adding as much RAM as practical; swapping is decreased and performance is improved.

The vmstat Display

vmstat is used to take a series of snapshots of how the system is performing, including:

- a summary of the number of processes in various states,
- paging activity,
- system activity,
- CPU cycle consumption.

The **vmstat** display looks like this:

```

procs          paging                      system      cpu
r  b  w  si  so  ch  cm  ffr  swr  sww  rec  shf  shc  cpy  pf  in  sy  cs  us  su  id
2  67  0  0  0  0  0  0  0  0  0  0  0  0  155  68  198  26  72  20  8

```

For the purposes of this procedure, the only categories you are interested in are as follows:

- si** Number of processes swapped in.
- so** Number of processes swapped out.
- ch** Number of page cache hits; the number of pages that were re-used rather than retrieved from disk.
- cm** Number of page cache misses; essentially the number of times the system had to retrieve a page from the hard disk.
- id** The percentage of time the CPU spent doing nothing; idle time.

Checking Buffer and Page Cache Usage

To determine whether you have enough memory in the page pool for the page caching mechanism to work efficiently, do the following:

1. Run a small set of programs over and over in addition to your usual process and user load.
2. Start the **vmstat** command running at fifteen second intervals by entering the following:

```
vmstat 15
```

Watch the columns “ch” (page cache hits) and “cm” (page cache misses) for a few minutes. If you see more misses than hits, you may find that a larger page pool will help performance. (However, an initially poor page cache hit-to-miss ratio is to be expected when running programs for the first time.) If the page pool is too small, the usual solution is to add more memory, but there may be application mixes for which shrinking the buffer cache will achieve the same result.

See “Sizing the Buffer Cache” in the “Improving Disk Initialization” section earlier in this chapter.

Checking CPU Usage

The “id” (percentage idle) column may also provide some insight into system performance. This figure is normally between 40 and 100 percent, even with a large number of active users. When this figure falls consistently below 30%, the chief competition for resources does not involve memory at all; the critical resource is raw processor power. (Run the **ps(C)** command to make certain that the excessive CPU usage is not due to a runaway process that is stealing every spare CPU cycle.)

If you are running a large number of users and you are using non-intelligent serial cards, smart cards can take some of the burden off the CPU.

In addition, you should examine the **/usr/spool/cron/crontabs** directory to see if jobs are queued up for peak periods that might better be run at times when the system is idle. Use the **ps** command to determine what processes are heavily loading the system. Encourage users to run large, non-interactive commands (such as **nroff(CT)** or **troff(CT)**) at off-peak hours. You may also want to run such commands with a low priority by using the **nice(C)** or **batch(C)** commands.

Summary of Tunable Parameters

Tunable system parameters are used to set various table sizes and system thresholds to handle the expected system load. Caution should be used when changing these variables since such changes can directly affect system performance. For the most part, the default tunable parameter values are acceptable for most configurations and applications. If your application has special performance needs, you may have to experiment with different combinations of parameter values to find an optimal set.

Table 8.3 shows the recommended tunable parameter values for a system equipped with different amounts of hardware memory (RAM) installed.

The following notes apply to the Tunable Parameters Table 8.3:

- The value of a few parameters are calculated each time a new kernel (`/xenix`) is generated, unless the value is manually overridden (see note below).
- The default value and the size in bytes for each entry are shown in the table.
- A dash (—) is used in the size information to indicate parameters that set flags in the kernel. Parameters that set flags do not affect the size of the kernel when their values are changed; only the values of the specific flags are changed.
- A dagger (†) indicates a parameter that applies only to XENIX-386 distributions.
- An asterisk (*) indicates a parameter that applies only to XENIX-286 distributions.

Note

The values shown are for XENIX-386 distributions, and may vary in other distributions.

Overriding the calculated value causes the parameter to be set to the new value each time a new kernel (`/xenix`) is generated. When calculated parameter values are overridden, however, any subsequent changes in the hardware configuration (adding new memory for example) require a change of the value that was manually set. This will allow you to optimize the performance of the new configuration.

Table 8.3
Configurable Kernel Parameters

Parameter	RAM Installed				Default Value	Size per Entry in Bytes
	1 MEG	2 MEGS	3-5 MEGS	6-15 MEGS		
NDISK	—	—	—	—	6	—
NBUF	100	250	400	600	CALC.	1024
NSABUF*	—	—	—	—	0	—
NPBUF	4	16	40	60	32	52
NHBUF	32	64	128	256	512	12
MAXBUF	—	—	—	—	1024	—
NCLIST	100	100	160	200	100	72
NEMAP	—	—	—	—	10	—
NSXT	—	—	—	—	1	—
NINODE	100	225	300	400	250	68
NFILE	100	225	300	400	250	12
NMOUNT	8	8	8	8	32	36
NFLOCKS	—	—	—	—	100	—
CMASK	—	—	—	—	0	—
NPROC	60	70	80	100	200	168
MAXUPRC	25	25	30	30	50	—
MEMLIM†	—	—	—	—	100	—
SWPLIM†	—	—	—	—	30	—
CMAPSIZ*	—	—	—	—	CALC.	—
SMAPSIZ*	—	—	—	—	CALC.	—
NTEXT*	—	—	—	—	1	—
NCALL	—	—	—	—	100	—
TIMEZONE	—	—	—	—	480	—
DSTFLAG	—	—	—	—	1	—
NSCRN	—	—	—	—	CALC.	—
NSPTTYS	16	16	16	16	16	—
SCRNMEM	—	—	—	—	CALC.	—
TBLNK†	—	—	—	—	0	—

(Continued on next page.)

Table 8.3
Configurable Kernel Parameters (Continued)

Parameter	RAM Installed				Default Value	Size per Entry in Bytes
	1 MEG	2 MEGS	3-5 MEGS	6-15 MEGS		
MSGMAP	513	513	513	513	513	8
MSGMAX	8192	8192	8192	8192	8192	—
MSGMNB	8192	8192	8192	8192	8192	—
MSGMNI	10	10	10	10	10	53
MSGTQL	60	60	60	60	60	12
MSGSSZ	8	8	8	8	8	1024
MSGSEG	1024	1024	1024	1024	CALC.	8
SEMMAP	21	21	21	21	21	8
SEMMNI	10	10	10	10	10	32
SEMMNU	20	20	20	20	20	8X(SEMUME+2)
SEMMSL	10	10	10	10	10	—
SEMOPM	5	5	5	5	5	8
SEMUME	5	5	5	5	5	8xSEMMNU
SEMVMX	32766	32766	32766	32766	32766	—
SEMAEM	16384	16384	16384	16384	16384	—
SEMMNS	40	40	40	40	40	8
NSDSEGS†	25	25	25	25	25	—
NSDSLOTS†	3	3	3	3	3	—
SHMMAX	131072	131072	131072	131072	131072	—
SHMMIN†	1	1	1	1	1	—
SHMMNI†	25	25	25	25	25	52
SHMSEG†	6	6	6	6	6	12XNPROC
SHMALL†	4096	4096	4096	4096	4096	—
NODE	—	—	—	—	—	—
NQUEUE	—	256	384	384	1	36
NSTREAM	—	32	48	48	1	52
NBLK8192	—	0	0	0	0	4142
NBLK4096	—	0	0	0	0	4142
NBLK2048	—	20	40	40	0	2094
NBLK1024	—	12	32	32	0	1070
NBLK512	—	8	18	18	0	558
NBLK256	—	16	48	48	0	302
NBLK128	—	64	128	128	0	174
NBLK64	—	256	256	256	0	110
NBLK32	—	128	128	128	0	110
NBLK16	—	128	256	256	0	61
NBLK4	—	128	128	128	0	50

(Continued on next page.)

Table 8.3
Configurable Kernel Parameters (*Continued*)

Parameter	RAM Installed				Default Value	Size per Entry in Bytes
	1 MEG	2 MEGS	3-5 MEGS	6-15 MEGS		
NMUXLINK	—	32	48	48	1	12
NSTREVENT	—	256	256	256	1	12
NSTRPUSH	—	9	9	9	0	—
MAXSEPGCNT	—	1	2	2	1	2048
NDBLOCK	—	—	—	—	—	—
NMBLOCK	—	—	—	—	—	—
STRMSGSZ	—	4096	4096	4096	0	—
STRCLSZ	—	1024	1024	1024	0	—
STRLOFRAC	—	80	80	80	0	—
STRMEDFRAC	—	90	90	90	0	—
EVQUEUES	1	3	6	40	8	—
EVDEVS	2	6	12	36	—	16
EVDEVSPERQ	2	3	3	32	3	—
DMAEXCL	—	—	—	—	—	1
KBTYPE	—	—	—	—	—	0
AD_SCATTER	—	—	—	—	0	—
KB_MOUSE	—	—	—	—	0	—

The following is a complete list of configurable parameters, their purpose, and suggested tuning values.

Note

The parameters are listed by category and in the order they are displayed by **configure** instead of alphabetically.

Disks and Buffers

NDISK is the number of hard disk drives attached to the system. The default value is six.

NBUF number of 1K system buffers allocated at boot time. The buffers form a data cache. The data cache is a memory array containing disk file information. Cache hit rate increases with the number of buffers. Cache hits reduce the number of disk accesses and thus may improve overall performance. This parameter is normally in the range of 100 to 600. (See the file */usr/adm/messages* for the value of NBUF; it is determined at boot time and displayed as “i/o bufs.”) The

Summary of Tunable Parameters

number of hash queues (NHBUF) should be increased along with system buffers (NBUF) for optimal performance.

NSABUF The number of buffer cache buffers in near kernel data on 286 kernels. (This parameter only applies to XENIX-286 distributions.)

NPBUF specifies how many physical I/O buffers to allocate. One I/O buffer is needed for each physical read or write active. The default value is 32.

NHBUF specifies how many hash queues to allocate for 1K buffers. These are used to search for a buffer given a device number and block number rather than a linear search through the entire list of buffers. This value must be a power of 2. The NHBUF value must be chosen so that the value NBUF divided by NHBUF is approximately equal to 4. (See the file */usr/adm/messages* for the value of NBUF; it is determined at boot time and displayed as "i/o bufs.")

CTBUFSIZE is the size of the tape (QIC-11, QIC-24) buffer in Kbytes. This should have a value of 32 to 256. It is the size of a static buffer allocated at init time. The following are reasonable values for associated circumstances:

32K	bare minimum: insufficient to stream
64K	minimum to allow streaming (good for systems with little memory) or little tape use (performance is not critical)
96K	a first reduction, if default uses too much memory
128K	default: good tradeoff performance
192K	a first increase, if default provides poor performance
256K	maximum

MAXBUF is the maximum possible number of buffers in the buffer cache. This is the number of buffer description headers in the kernel. Fewer than this number of buffers may actually be autoconfigured by the kernel at boot time, depending on how much core is present. If NBUF is non-zero, then exactly NBUF buffers are configured, and there is no reason for MAXBUF to be

Summary of Tunable Parameters

larger than NBUF. If NBUF is 0, the kernel configures at most MAXBUF buffers automatically. (This parameter only applies to XENIX-386 distributions.)

Character Buffers

The following parameters control various data structure sizes and other limits in base system device drivers.

- NCLIST** specifies how many character list buffers to allocate. Each buffer contains up to 64 bytes. The buffers are dynamically linked to form input and output queues for the terminal lines and other slow-speed devices. The average number of buffers needed per terminal is in the range of 5 to 10. Each entry (buffer space plus header) contains 72 bytes. When full, input and output characters dealing with terminals are lost, although echoing continues.
- NEMAP** specifies the maximum number of I/O translation mappings.
- NSXT** The number of shell-layer sessions.

Files, Inodes, and Filesystems

- NINODE** specifies how many inode table entries to allocate. Each table entry represents an in-core inode that is an active file. For example, an active file might be a current directory, an open file, or a mount point. The file control structure is modified when changing this variable. The number of entries used depends on the number of opened files. The entries are normally in the range of 100 to 400. The default is 250. The value for NINODE pertains directly to the NFILE value. (NINODE is equal to or greater than NFILE).
- NFILE** Specifies how many open file table entries to allocate. Each entry represents an open file. The entry is normally in the range of 100 to 600. The default is 250. The NFILE entry relates directly to the NINODE entry. (NFILE is less than or equal to NINODE.) The NFILE control structure

Summary of Tunable Parameters

operates in the same manner as the NINODE structure. When the file table overflows, the following warning message is displayed on the system console:

```
file table overflow
```

As a reminder, this parameter does not affect the number of open files per process (see the NOFILES parameter).

- NMOUNT** specifies how many mount table entries to allocate. Each entry represents a mounted filesystem. The root (/) filesystem is always the first entry. When full, the **mount** system call returns the error EBUSY. Because the mount table is searched linearly, this value should be as low as possible.
- NFLOCKS** The number of files and records that can be locked at once. The default value is 100.
- CMASK** is the default mask (**umask(S)**) used for file creation.

Processes, Memory Management and Swapping

- NPROC** specifies how many process table entries to allocate. Each table entry represents an active process. The swapper is always the first entry, and */etc/init* is always the second entry. The number of entries depends on the number of terminal lines available and the number of processes spawned by each user. The average number of processes per user is in the range of 2 to 5. When full, the **fork(S)** system call returns the error EAGAIN. This error also generates the “no more processes” screen message. If any significant change is made to NPROC, CMAPSIZ and SMAPSIZ (286 only) should also be changed (see below). The NPROC entry is in the range of 50 to 400. The default is 200.

Summary of Tunable Parameters

MAXUPRC	The number of processes that a single user can run simultaneously. A process attempting to fork when the user already has MAXUPRC processes active will receive the error EAGAIN. The entry is normally in the range of 15 to 30. This value should not exceed the value of NPROC (NPROC should be at least 10% more than MAXUPRC). This value is per user identification number, not per terminal. For example, if 12 people are logged in on the same user identification, the default limit would be reached very quickly. The default value is 50.
MEMLIM	A process may occupy up to this percent of user memory, plus the swap area it can occupy (which is constrained by SWPLIM). (This parameter only applies to XENIX-386 distributions.)
SWPLIM	A process may occupy up to this percent of swap area, plus the memory area it can occupy (which is constrained by MEMLIM). (This parameter only applies to XENIX-386 distributions.)
CMAPSIZ	Table used to hold list of program segments in memory. (This parameter only applies to XENIX-286 distributions.)
SMAPSIZ	Table used to hold lists of program segments being swapped. While CMAPSIZ and SMAPSIZ rarely need to be changed themselves, they should vary with NPROC, each being NPROC * 2 by default. (This parameter only applies to XENIX-286 distributions.)
NTEXT	Maximum text segments (systemwide) (This parameter only applies to XENIX-286 distributions.)

Clock

NCALL specifies how many call-out table entries to allocate. Each entry represents a function to be invoked at a later time by the clock handler portion of the kernel. This value must be in the range of 30 to 250. The default value is 100. Software drivers may use call entries to check hardware device status. When the call-out table overflows, the system crashes and displays the following message on the system console:

Timeout table overflow

TIMEZONE specifies the **timezone** setting referred to in the **ctime(S)** system call. Note that the **timezone** value is a system default timezone and not the value of the **TZ** environment variable.

DSTFLAG specifies daylight savings time, the **dstflag** described for the **ctime(S)** system call.

MultiScreens

NSCRN specifies the maximum number of multiscreens that can be used by the video driver.

NSPTTYS is the number of pseudo-ttys on the system. The default is 16 and the maximum is 32.

SCRNMEM is the number of 1024-byte blocks for console screen saves.

TBLNK controls the console screen saver feature. It is the number of seconds before the screen blanks to save wear on the monitor. **TBLNK** can have a value of 0 to 32767, with zero disabling screen blanks. The default is 0. (This parameter only applies to XENIX-386 distributions.)

Message Queues

The following tunable parameters are associated with interprocess communication messages:

- MSGMAP** specifies the size of the control map used to manage message segments. This value is calculated as $\text{MSGSEG}/2+1$. The default is 513.
- MSGMAX** specifies the maximum size of a message. This value is calculated as $\text{MSGSEG}*\text{MSGSSZ}$. The default is 8192. Although the maximum possible size the kernel can process is 64 Kbytes -1, the limit is 8192.
- MSGMNB** specifies the maximum length of a message queue. This value is calculated as $\text{MSGSEG}*\text{MSGSSZ}$. The default is 8192.
- MSGMNI** specifies the maximum number of message queues system-wide (id structure). The default value is 10.
- MSGTQL** specifies the number of message headers in the system and, thus, the number of outstanding messages. The default value is 60.
- MSGSSZ** specifies the size, in bytes, of a message segment. Messages consist of a contiguous set of message segments large enough to fit the text. The default value is 8. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 Kbytes).
- MSGSEG** specifies the number of message segments in the system. The default value is 1024. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 Kbytes). Default value is set at boot time.

Semaphores

The following tunable parameters are associated with interprocess communication semaphores:

SEMMAP	Specifies the size of the control map used to manage semaphore sets. This value is calculated as $SEMMNS/2+1$. The default is 21.
SEMMNI	Specifies the number of semaphore identifiers in the kernel. This is the number of unique semaphore sets that can be active at any given time. The default value is 10.
SEMMNS	Specifies the number of semaphores in the system. The default value is 40.
SEMMNU	Specifies the number of undo structures in the system. The default value is 20.
SEMMSL	Specifies the maximum number of semaphores per semaphore identifier. The default value is 10.
SEMOPM	Specifies the maximum number of semaphore operations that can be executed per semop(S) system call. The default value is 5.
SEMUME	Specifies the maximum number of undo entries per undo structure. The default value is 5.
SEMVMX	Specifies the maximum value a semaphore can have. The default value is 32766. The default value is the maximum value for this parameter.
SEMAEM	Specifies the adjustment on exit for maximum value, alias semadj . This value is used when a semaphore value becomes greater than or equal to the absolute value of semop(S) , unless the program has set its own value. The default value is 16384. The default value is the maximum value for this parameter.

Summary of Tunable Parameters

Shared Data

The following tunable parameters are associated with interprocess communication shared memory: (These parameters are only valid on 80386-based machines.)

SHMMAX	Specifies the maximum shared memory segment size. The default value is 131072.
SHMMIN	Specifies the minimum shared memory segment size. The default value is 1.
SHMMNI	Specifies the maximum number of shared memory identifiers system wide. The default value is 25.
SHMSEG	Specifies the number of attached shared memory segments per process. The default value is 6. The maximum value is 15.
SHMALL	Specifies the maximum number of in-use shared memory text segments. The default value is 4096.

System Name

NODE	specifies the system name.
------	----------------------------

Streams Data

Note

The STREAMS parameters are only valid when the STREAMS runtime system is installed.

Summary of Tunable Parameters

The following tunable parameters are associated with STREAMS processing.

- NQUEUE** is the number of Streams queues to be configured. Queues are always allocated in pairs, so this number should be even. A minimal Stream contains four queues (two for the Stream head, two for the driver). Each module pushed on a Stream requires an additional two queues. A typical configuration value is $4 * \text{NSTREAM}$.
- NSTREAM** is the number of Stream-head (stdata) structures to be configured. One is needed for each Stream opened, including both Streams currently open from user processes and Streams linked under multiplexers. The recommended configuration value is highly application-dependent, but a value of 32 to 40 usually suffices on a computer for running a single transport provider with moderate traffic.
- NBLK_n** NBLK4 through NBLK4096 control the number of Streams data blocks and buffers to be allocated for each size class. Message block headers are also allocated based on these numbers: the number of message blocks is 1.25 times the total of all data block allocations. This provides a message block for each data block, plus some extras for duplicating messages (kernel functions **dupb()**, **dupmsg()**). The optimal configuration depends on both the amount of primary memory available and the intended application.
- NMUXLINK** is the maximum number of multiplexer links to be configured. One link structure is required for each active multiplexer link (STREAMS **I_LINK ioctl**). This number is application-dependent; the default allocation of 87 guarantees availability of links.
- NSTRPUSH** is the maximum number of modules that may be pushed onto a Stream. This prevents an errant user process from consuming all of the available queues on a single Stream. By default this value is 9, but in practice, existing applications have pushed at most four modules on a Stream.
- NSTREVENT** is the initial number of Stream event cells to be configured. Stream event cells are used for

Summary of Tunable Parameters

recording process-specific information in the **poll** system call. They are also used in the implementation of the Streams **I_SETSIG ioctl** and in the kernel **bufcall()** mechanism. A rough minimum value to configure would be the expected number of processes to be simultaneously using **poll** times the expected number of Streams being polled per process, plus the expected number of processes expected to be using Streams concurrently. The default is 256. Note that this number is not necessarily a hard upper limit on the number of event cells that are available on the system (see **MAXSEPGCNT**).

MAXSEPGCNT is the number of additional pages of memory that can be dynamically allocated for event cells. If this value is 0, only the allocation defined by **NSTREVENT** is available for use. If the value is not 0 and if the kernel runs out of event cells, it will under some circumstances attempt to allocate an extra page of memory from which new event cells can be created. **MAXSEPGCNT** places a limit on the number of pages that can be allocated for this purpose. Once a page is allocated for event cells, however, it cannot be recovered later for use elsewhere. The default value is 1.

STRMSGSZ is the maximum allowable size of the data portion of any Streams message. This should usually be set just large enough to accommodate the maximum packet size restrictions of the configured Streams modules. If it is larger than necessary, a single **write** or **putmsg** can consume an inordinate number of message blocks. The default value of 4096 is sufficient for existing applications.

STRCTLSZ is the maximum allowable size of the control portion of any STREAMS message. The control portion of a **putmsg** message is not subject to the constraints of the minimum/maximum packet size, so the value entered here is the only way of providing a limit for the control part of a message. The default value of 1024 is more than sufficient for existing applications.

STRLOFRAC is the percentage of data blocks of a given class at which low-priority block allocation requests are automatically failed. For example, if **STRLOFRAC** is

Summary of Tunable Parameters

40 and there are forty-eight 256-byte blocks, a low-priority allocation request fails when more than nineteen 256-byte blocks are already allocated. The parameter helps prevent deadlock situations by starving out low-priority activity. The default value of 80 works well for most applications. STRLOFRAC must be greater than or equal to 0 and less than or equal to STRMEDFRAC.

STRMEDFRAC is the percentage cutoff at which medium priority block allocations are failed (see STRLOFRAC). The default value of 90 works well for most applications. STRMEDFRAC must be greater than or equal to STRLOFRAC and less than or equal to 100. (There is no cutoff fraction for high-priority allocation requests; it is effectively 100.)

Event Queues and Devices

- EVQUEUES is the maximum number of open event queues systemwide. The default value is 8.
- EVDEVS is the maximum number of devices attached to event queues systemwide. The default value is 16.
- EVDEVSPERQ is the maximum devices per event queue. The default value is 3.

Hardware Dependent Parameters

- DMAEXCL specifies whether simultaneous DMA requests are allowed. Some computers have DMA chips that malfunction when more than one allocated channel is used simultaneously. For all installations on these computers, DMAEXCL is set to one by default. On computers that do not suffer from this problem, set DMAEXCL to zero to allow simultaneous DMA on multiple channels.
- KBTYPE is set to 0 for XT-type keyboards and 1 for AT-type keyboards.

Summary of Tunable Parameters

AD_SCATTER This parameter is used for scatter-gather support, where groups of disk I/O requests are gathered together rather than executed singly, increasing overall disk efficiency. This feature is available only on certain Adaptec SCSI adapters. To enable scatter-gather, set this parameter to 1. The default value is 0. (This parameter only applies to XENIX-386 distributions.)

KB_MOUSE This parameter is used to define if a keyboard mouse is present. The default value is 0, meaning no keyboard mouse is present. This parameter is controlled by **mkdev mouse** and is not accessed by **configure**. (This parameter only applies to XENIX-386 distributions.)

Chapter 9

Using DOS and OS/2

Introduction 9-1

OS/2 Coexistence 9-2

Partitioning the Hard Disk Using fdisk 9-3
Switching Operating Systems 9-5

Installing a XENIX Partition on a DOS System 9-8

Using a XENIX System and DOS with Two Hard Disks 9-9

Removing an Operating System from the Hard Disk 9-10

DOS Accessing Utilities 9-11
File and Directory Arguments 9-12
User Configurable Default File 9-12

XENIX Systems and DOS On Non-Standard Disks 9-14

Introduction

This chapter explains how you can switch from XENIX to DOS and access DOS files while running XENIX. The XENIX system provides this facility so that XENIX and DOS can coexist on the same system.

Several programs make this coexistence possible. The **dos(C)** utilities allow access to DOS files on diskettes or on the DOS partition on the hard disk. These utilities are discussed later in this chapter. The hard disk is divided into partitions by the **fdisk** program, available in DOS and XENIX versions. The “Partitioning the Hard Disk with **fdisk**” section explains how to use DOS **fdisk** to create a DOS partition and XENIX **fdisk(ADM)** to create a XENIX partition on the same hard disk. Another section discusses installing a XENIX partition on the hard disk along with DOS. There is also a section explaining various booting configurations, for users who mostly use the XENIX system and for users who mostly use DOS.

Note

You must have DOS 3.3 or earlier installed on your system. Extended DOS partitions are not supported. In addition, be sure you follow the guidelines for DOS partitions described in the *XENIX Installation Guide*.

OS/2 Coexistence

Although it may install successfully, OS/2 may not be bootable on your machine, regardless of whether a XENIX partition is present or not; we cannot guarantee that OS/2 will work with your XENIX system. Refer to your computer's hardware documentation to determine if your machine is supposed to run OS/2. If you wish to use OS/2 and/or DOS on the same disk with your XENIX system, you must load them in the following order:

1. DOS (partition must be 32Mbytes or less)
2. XENIX software
3. OS/2

There are no OS/2 tools available (such as the DOS utilities described in this chapter). In addition, you must use **fdisk(ADM)** to switch to or from OS/2.

XENIX **fdisk(ADM)** displays an OS/2 partition as DOS.

Partitioning the Hard Disk Using **fdisk**

Each version of **fdisk** is documented in the respective operating system's manual. Unless otherwise noted, this chapter refers to the XENIX version of **fdisk**(ADM).

Note

XENIX **fdisk** displays DOS partitions as *DOS* while DOS **fdisk** displays XENIX partitions as *Other*.

You can only create DOS partitions using DOS **fdisk**, and only XENIX partitions using XENIX **fdisk**.

Be aware that DOS **fdisk** reports sizes in terms of cylinders, while XENIX **fdisk** reports sizes in terms of tracks. Check your hard disk manual for the number and size of cylinders on your hard disk.

fdisk is interactive, and uses a menu to display your options. Here is the main **fdisk** menu:

- ```
1. Display Partition Table
2. Use Entire Disk for XENIX
3. Create XENIX Partition
4. Activate Partition
5. Delete Partition
```

```
Enter your choice or 'q' to quit:
```

The **fdisk** utility allows you to set up separate areas (partitions) on your hard disk for your operating system. The hard disk is divided into *tracks*. The number of tracks depends upon the size of the hard disk.

A *partition* consists of a group of tracks. One hard disk may contain up to four partitions. Each partition can have a different operating system and associated directories and filesystems.

## Partitioning the Hard Disk Using fdisk

The **fdisk** command allows you to specify a disk partition as “active”. This means that when you turn on (boot) your computer, the operating system installed in the active partition will start running. The XENIX partition must be active when you intend to use your XENIX system.

You can also specify the number of tracks assigned to each partition. The number of available tracks will vary according to the size of your hard disk. Consult your *Release Notes* for the recommended XENIX partition size. The size of the XENIX partition also depends on the number of software packages you want to install. You can install the XENIX system in this space, and have the rest of the space for user files and other software packages. Refer to the **custom(ADM)** manual page for information on how to install and remove software.

**fdisk** allows you to specify where the partition begins and ends. **fdisk** will not allow you to construct overlapping partitions. You do not need to install your XENIX system in the first partition.

You should always start your DOS partition at the beginning of the disk, starting at cylinder 1, not cylinder 0. Because DOS writes the boot block on cylinder 0 very close to the end of the masterboot block, starting your DOS partition on cylinder 0 can cause the DOS partition to become inaccessible after installing XENIX.

Start the XENIX partition at the beginning of the next cylinder on the disk. To find the beginning of the next cylinder, note the ending track number of your DOS partition and start the XENIX partition on the next track number that is a multiple of the number of heads on your hard disk. For example, if you have five heads on your hard disk and your partition ends at track 103, start your XENIX partition at track 105.

When you are running your XENIX system, the device name of the XENIX partition is `/dev/hd0a`. For more information about hard disk device names, see the **hd(HW)** page.

## Partitioning the Hard Disk Using fdisk

One option of **fdisk** tabulates the current state of the partitions (the Display Partition Table option). This option lists, for each partition, whether the partition is active, the first track, the last track, the number of tracks used, and the associated operating system. If you enter the Display Partition Table option and press (Return) to see the partition table, the result will be similar to this:

Current Hard Disk Drive: /dev/hd00

| Partition | Status   | Type  | Start | End  | Size |
|-----------|----------|-------|-------|------|------|
| 1         | Inactive | DOS   | 005   | 398  | 393  |
| 2         | Active   | XENIX | 400   | 1219 | 819  |

Total disk size: 1229 tracks (9 tracks reserved for masterboot and diagnostics).

## Switching Operating Systems

There are three ways to switch to DOS once you have set up separate DOS and XENIX partitions:

- Enter **dos** at the boot prompt,
- Use a floppy diskette that contains the files necessary to boot the DOS operating system, or
- Use **fdisk** to change the current active partition.

We recommend that you use a boot floppy or enter **dos** at your boot prompt to boot the DOS operating system. Booting from a floppy or the boot prompt is generally easier, faster, and safer than constantly using **fdisk** to change active partitions.

When you use the boot prompt or a floppy to boot DOS, the XENIX partition remains active even though you have switched operating systems. When you use **fdisk**, the XENIX partition is inactive until you switch back to it.

## Partitioning the Hard Disk Using fdisk

To use the boot prompt method, enter:

**dos**

at the boot prompt:

```
Boot
:
```

To use a floppy diskette to boot DOS, follow this procedure:

1. Make sure all users are logged off the system.
2. Run **shutdown(ADM)** to shut down the XENIX system. This command makes sure all users know the system is being shut down, terminates all processes, then halts the system.
3. Once the XENIX system has shut down, insert the bootable DOS diskette into the primary (boot) drive.
4. Boot DOS.
5. To get back to the XENIX partition, remove any disks from the floppy drive(s) and press **<Ctrl><Alt>-<Del>**, or the reset key, or turn the computer off, then on. Since the XENIX partition is still active, your XENIX system boots.

Remember that if you have an active XENIX partition and boot DOS from a floppy you can transfer to C: to work with the DOS files.

The other way to change operating systems is to run **fdisk** and change the active partition from the XENIX partition to DOS. Then, after you shut down the system (see the previous steps) DOS boots from the hard disk. You do not need a bootable DOS floppy disk as long as DOS is loaded on the DOS partition of the hard disk.

To switch back to the XENIX partition, run **fdisk** under DOS and make the XENIX partition active. To reboot the XENIX partition, press **<Ctrl><Alt>-<Del>**, or the reset key, or turn the computer off, then on.

Because the XENIX partition must be active for it to operate, you cannot use a bootable floppy to boot the XENIX operating system. This second method is appropriate for an occasional change of the active operating system.

**Table 9.1**  
**DOS Hard Disk Devices**

| XENIX device | UNIX device    |
|--------------|----------------|
| /dev/hd0d    | /dev/dsk/0sd   |
| /dev/rhd0d   | /dev/rdisk/0sd |
| /dev/hd1d    | /dev/dsk/1sd   |
| /dev/rhd1d   | /dev/rdisk/1sd |

The hard disk device names in Table 9.1 reference the DOS partition (note that both XENIX and UNIX DOS device conventions are supported). The entries */dev/hd0d* and */dev/rhd0d* represent the DOS partition of the first hard drive and */dev/hd1d* and */dev/rhd1d* represent the DOS partition of the second hard drive. The hard disk driver determines which partition is the DOS partition.

---

## Installing a XENIX Partition on a DOS System

If you wish to set up your XENIX system on a hard disk which previously contained only DOS, follow these steps:

1. Copy (back up) all the DOS files and directories on the hard disk onto floppies, or whatever backup media you wish to use.
2. If your hard disk has an extended partition installed, or if a utility other than DOS **fdisk** was used to create the DOS partition, you must do a low-level format of the hard disk.
3. Run **fdisk**, under DOS. If there is enough free space for a XENIX partition on your hard disk, (check your *Release Notes*) skip to step 5. Otherwise, delete the DOS partition, then recreate it, leaving enough room on the disk for your XENIX distribution and any other software that you intend to install.
4. Return the DOS files from the backup media to the newly created DOS partition on the hard disk. Keep the backups in case there is an error of some kind, so you will not lose any data.
5. Turn off your computer.
6. Follow the installation procedure outlined in the *XENIX Installation Guide* to install your XENIX distribution.

You will see a message warning that the contents of the hard disk will be destroyed. There is no cause for concern, because you have already backed up the DOS files and transferred them to the new DOS partition. The new partition being created will contain your XENIX system, and the installation process will only write information on the XENIX partition.

7. During the installation procedure, **fdisk** is invoked to partition the hard disk. Use **fdisk** to assign a sufficiently large XENIX partition. Do not use the "Use Entire Disk for XENIX" option.
8. Designate "XENIX" as the active operating system by choosing the "Activate Partition" option under **fdisk**.
9. Finish installing the XENIX distribution.

---

## Using a XENIX System and DOS with Two Hard Disks

Your computer always boots the operating system in the active partition on the first hard disk. The XENIX system must boot from the first hard disk. There are several ways to configure your system if you have two hard disks and want to boot DOS. Two ways are discussed here.

One configuration consists of designating the entire first disk as a XENIX partition. You then use a DOS boot floppy to start DOS and specify:

A> C:

to switch to the DOS area on the second hard disk, where C: is the designation for the second hard disk. This strategy works for some versions of DOS. Early versions recognize only the first hard disk on the system.

---

### *Note*

If you devote a hard disk for use with DOS, the disk must already be configured under DOS. See the “Adding Hard Disks” chapter of this guide for details regarding hard disk configuration.

---

Another method is to maintain a small DOS partition on the first hard disk. The DOS partition is designated the active partition. In this configuration, the computer always boots DOS. This requires changing the active partition to boot the XENIX system from the hard disk.

If you use the entire second disk for DOS, you need only run **mkdev hd** to create device files for the second disk if you plan to use the XENIX DOS utilities (**doscpc**, **dosls**, **doscat**, and so on). If you do not wish to use those utilities to access DOS files on the second hard disk, there is no need to run **mkdev hd**.

---

### *Note*

Be sure to make a backup copy of your boot floppies if you use them to boot your secondary DOS operating system.

---

## Removing an Operating System from the Hard Disk

You may find that you no longer need one of the operating systems installed on your hard disk. If you want to delete an operating system, use the appropriate version of **fdisk**. To delete a XENIX partition, you must use the XENIX version of **fdisk**. To delete a DOS partition, use **fdisk** under DOS. Deleting the partition removes the contents of that partition and leaves unallocated space.

You can then reallocate that space by either adding another XENIX or DOS partition, or enlarging an existing partition. Enlarging a partition requires reinstalling the operating system and (for a XENIX partition) remaking the filesystem on the partition using **divvy(ADM)**.

---

## DOS Accessing Utilities

The DOS accessing utilities are discussed in detail in the **dos(C)** manual page. Note that you must have a bootable, although not active, DOS partition on the hard disk or a DOS floppy in order to use these XENIX commands. For example, you can only transfer a file from a XENIX partition on hard disk to a DOS formatted floppy or to the DOS partition on the hard disk.

You may also be able to use the XENIX **dd(C)** and **diskcp(C)** commands to copy and compare DOS floppies. The XENIX **dtype(C)** can be used to determine if XENIX recognizes the data on the floppy disk as DOS data.

The file */etc/default/msdos* defines which DOS drives (for example, A:, B:, C: ...) correspond to XENIX devices.

---

### *Note*

You cannot execute (run) DOS programs or applications from your XENIX system. SCO VP/ix is used for this purpose.

---

The XENIX system does not record bad tracks in the DOS area of the hard disk. If a bad track develops in the DOS area, an operation such as **doscp** that attempts to access the affected area may fail. If such is the case, the message "Error on fixed disk" is displayed.

With smaller files, it may be possible to copy the files to another location under DOS and then access the copied version of each file.

### Note

When trying to use the DOS utilities to access files on your DOS partition, you may see the error message “bad media byte.” This message indicates that the DOS partition on the hard disk is not bootable. You can make your DOS partition bootable by first backing up the files on the DOS partition, booting DOS from the floppy, and formatting the DOS partition using the command:

**format c: /s**

You should then reinstall your DOS files.

---

## File and Directory Arguments

The file and directory arguments for DOS files take the form:

*device:filename*

The *device* is a XENIX pathname for the special device file where the DOS information is located. If the DOS information is located on a floppy disk in the primary floppy drive, the device name is */dev/fd0:*. The *filename* is a pathname to a DOS file on the device, for example, *:/usr/dosdr/file*. Note that subdirectories are specified with the XENIX “/” convention rather than the DOS “\” convention. If the *device:* argument is not present, the file is assumed to be a XENIX file. The next section explains how to use the DOS devices to define the *device* argument.

## User Configurable Default File

For convenience, the user configurable default file */etc/default/msdos* can define DOS drive names that you can use in place of XENIX special device file pathnames. For example, you can include the following entries in the above file:

```
A=/dev/fd096ds15
B=/dev/fd048ds9
C=/dev/hd0d
D=/dev/hd1d
```

Once you have defined the variables, you can use the drive letter **A:** in place of the special device file `/dev/fd0` (96ds15 by default) when referencing DOS files or directories. For example:

```
/dev/fd0:/john/memos
```

can be replaced with:

```
A:/john/memos
```

In the example, the drive letter **B:** refers to a low density (48ds9) primary floppy drive, and drive letters **C:** and **D:** refer to the DOS partition on a primary or secondary hard disk.

---

### *Note*

If you get the message “cannot open `/dev/hd0d`,” or a similar message, check the user permissions on the special device file involved. As super-user, change the permissions with the **chmod** command. For example:

```
chmod 666 /dev/hd0d
```

gives full read and write permissions to all users for the special device file `/dev/hd0d`, which is the DOS partition on the primary hard disk.

---

---

## XENIX Systems and DOS On Non-Standard Disks

The XENIX system provides support for “non-standard” hard disks. The term “non-standard” refers to hard disks for which there are no correct disk parameter entries in your computer’s ROM.

The correct parameters you specify for your non-standard disk(s) are stored in the masterboot block, which is the first sector of your boot hard disk drive. The hard disk characteristics are specified during XENIX installation and these characteristics are then written out with the rest of the masterboot block. The special masterboot block that comes with your XENIX distribution resets the disk parameters to the specified values no matter which operating system is “Active.” This mechanism provides non-standard disk support for both XENIX and DOS systems.

Although the special masterboot supports non-standard disks under DOS, you cannot use your XENIX system to install DOS on your hard disk. If a non-standard disk is being used, it is assumed that you already have some method to transfer your DOS files to the hard disk.

Unless you are changing the active partition, you should only use the XENIX **fdisk** to manipulate your hard disk partition table. Using DOS **fdisk** or custom **fdisks** provided by hard disk manufacturers after the XENIX system has been installed may disable non-standard disk characteristics, rendering your disk inaccessible.

## **Chapter 10**

# **Administering User Accounts**

---

Introduction 10-1

Adding a User Account 10-2

Creating a Group 10-7

Changing a User's Login Group 10-9

Changing a User ID 10-11

Removing a User Account 10-13



---

# Introduction

User accounts help the XENIX system administrator keep track of the people using the system and control their access to system resources. Ideally, each user should have a user account. Each account has a unique “login name” and “password” with which the user enters the system, and a “home directory” where the user works.

It is the system administrator’s job to create accounts for all users on the system, and maintain these accounts by changing user passwords, login groups, and user IDs when necessary.

This chapter explains how to:

- Add user accounts to the system
- Create a group
- Change an account’s login group
- Change an account’s user ID
- Remove user accounts from the system

The following sections describe each task in detail.

---

# Adding a User Account

You can add a user account to the system with the **mkuser** program. The program creates a new entry in the */etc/passwd* file. This entry contains information about the new user (such as login name and initial password) that the system uses to let the user log in and begin work. The program also creates a home directory for the user, a mailbox for use with the **mail** command, and an initialization file (for example, *.profile* for the Bourne or Korn shells or *.login* for C-shell) containing XENIX commands that are executed when the user logs in.

To create a new user account, follow these steps:

1. Log in as the super user.
2. Enter:

**mkuser**

Δ **sysadmsh** users select: Users→Add

and press the (Return) key. The system displays the following message:

```
Mkuser
Add a user to the system
Do you require detailed instructions? (y/n/q)
```

3. Enter the letter **y** if you want information about the program, otherwise type the letter **n**. Enter **q** (for “quit”) only if you wish to stop the program and return to the system. If you type a “q” to any “(y/n)” prompt, the program will stop and no changes will be made.

- Next, you are prompted for the login name:

```
Enter new user's login name, or enter q to quit:
```

The login name is the name by which XENIX identifies the user. It is usually a short version of the user's actual name, typed in lower-case letters. For example, either "johnd" (a first name and last initial) or "jdoe" (a first initial and last name) is customary for the user John Doe. Enter the new name, and press the `<Return>` key.

- When the program continues, you are prompted for user id:

```
Do you wish to use the next available user id? (y/n/q)
```

If you enter **n**, you are asked to specify the id number you wish to use.

- The program now prompts you for information about the new user's group name and group number:

```
Do you want to use the default group? (y/n):
```

A group name is the name of the group of users to which the new user will belong. Users in a group have access to a common set of files and directories. The group name is optional. If you enter "y", the user's group name will be "group" and the group ID number will be 50.

If you enter "n", the program responds with a list of existing groups and their members:

```
Existing groups are:
```

```
Group "groupers" (50): markham kuprin forbin
Group "malomars" (51): grauber fisher johnson
```

```
Do you want to use one of these groups? (y/n):
```

If you enter either **y** or **n**, you are asked which group you want to use. Enter the name of the group. You may create a new group by entering in the new name.

## Adding a User Account

Next, you are prompted for a group number. The group ID, or number, may be any number from 50 to 30000 that isn't already used for another group.

7. After entering the group name and ID, you are prompted for the initial password.

```
Enter at least 5 characters for the password.
Enter password:
```

The initial password is the password you assign to the new user. The user will use the initial password to enter the account for the first time. Once in the account, the user should create a new password for himself, one that is hard to guess. (See the section "Changing Your Password" in the "Logging In" chapter of the *XENIX Tutorial*.)

Password management is discussed in the "Maintaining System Security" chapter of this guide. The system administrator has great flexibility in defining the length of passwords and how often they must be changed.

8. Enter the password, and press the <Return> key.
9. Next, you are prompted for a shell type. You see a list and brief explanation of the available shells (the menu selections depend on what packages and/or applications installed on your system) and the prompt:

```
ENTER Shell type (1, 2, 3,...) and press RETURN:
```

**sh** is the standard (Bourne) shell. **vsh** is the menu driven "visual" shell, **csh** is the C-shell, **rsh** is the restricted shell, **ksh** is the Korn shell, and **uucp login** is */usr/lib/uucp/uucico* enabling logging in to the system via **uucp**. For more information, see **sh(C)**, **ksh(C)**, **vsh(C)**, **csh(C)**, or **uucico(C)** in the *XENIX Reference*.

10. Enter the desired shell number and press <Return>. After you have entered the shell type, the program prompts you for a comment:

```
Please enter Comment >-----
 >
```

A comment is information about the new user, such as a department name and phone extension. Although, the comment is optional, it is useful if the **finger(C)** command is often used to display information about users. If given, the comment must be no more than 20 characters long, including spaces. It must not contain any colons (:). The example

```
Gregory Burgess, 123
```

shows the recommended form for a comment.

11. Enter the comment. Make sure it is 20 characters or less. If you do not wish to enter a comment, just press **<Return>**.

The program now displays what you have entered and the special user entry that it has created for the new user. This entry is copied to the special system file */etc/passwd*. The entry shows the login name, the encrypted password, the user ID, the group ID, the comment, the user's home directory, and the startup program. Fields in the entry are separated by colons (:). (For a full description of each field, see **passwd(F)** in the *XENIX Reference*.)

The program then gives you an opportunity to change the user name, password, group, or comment:

```
Username is "mavrac", user ID is 2001.
Group name is "group", group number is 50.
Comment field is: Mavra Chang, 123
Shell is "/bin/csh"

Do you want to change anything? (y/n):
```

12. Enter the letter **y** (for “yes”) and press the **<Return>** key, if you wish to change something. Enter **n** (for “no”) and skip to the next step if you wish to complete the new account. (Enter **q**, for “quit”, only if you wish to leave the program and abort the new account.)

## Adding a User Account

If you enter **y**, the program prompts for the field you wish to change:

- ```
1. User name
2. User id (uid)
3. Group
4. Shell
5. Password
6. Comment in /etc/passwd
```

```
Select an item to change, or enter q to quit:
```

Enter the name of the item (field) you wish to change and press **<Return>**. After you have changed a field, you see the complete list of fields and are asked if you wish to make other changes. When you are finished with any changes, the program adds the user.

13. The program displays the messages:

```
Password file updated.
Group file updated.
Home directory /x/name created.
/x/name/.shellfile created.
Test mail sent to user user.
User name added to this system.
```

The program then asks if you wish to add another user to the system.

14. Enter **y** if you wish to add another user. Otherwise, enter **n** to stop the program and return to the super user prompt.

A user can log into a new account as soon as it is created. For details see the "Logging In" chapter of the *XENIX Tutorial*. Be sure to communicate the new user's password and remind them to change it.

Creating a Group

A group is a collection of users who share a common set of files and directories. The advantage of groups is that users who have a common interest in certain files and directories can share these files and directories without revealing them to others. Initially, all users belong to the common system group named “group”, but you can create new groups by modifying the XENIX system file */etc/group* using a XENIX text editor.

To create a new group, you need to choose a group name and a group identification number (group ID). You also need to make a list of the users in the new group. The group name may be any sequence of letters and numbers up to eight characters long, and the group ID may be any number in the range 50 to 30000. Both the group name and ID must be unique, i.e., they must not be the same as any existing group name or ID.

To create a new group, follow these steps:

1. Log in as the super user.
2. Display the contents of the */etc/group* file by entering:

```
cat /etc/group
```

```
Δ sysadmsh users select: Users→Add→Group
```

and pressing the `<Return>` key. The `cat` command displays the contents of the */etc/group* file. The file contains several entries, each defining the group name, group ID, and users for a group. Each entry has the form:

```
group-name::group-ID:users
```

The users are shown as a list of login names separated by commas (.). For example, a typical file may look like this:

```
other:x:1:sergeo, nikkiz
sys:x:2:
group::50:mavrac,nathanb
```

3. Check the */etc/group* file entries to see that the group name and ID you have chosen are unique.
4. If the group name and ID are unique, invoke a text editor and specify */etc/group* as the file to edit.

Creating a Group

5. Locate the last line in the file, then insert the new entry in the form given above. For example, if you wish to create a group named "cpo" with group ID "142" and users "forbin", "markham", and "kuprin", enter:

```
cpo::142:forbin,markham,kuprin
```

6. Exit the editor.

To make sure you have entered the group names correctly, use the **grpcheck(C)** command to check each entry in the */etc/group* file. If the new entry is free of errors, no other changes to the file are required.

You can create any number of new groups. Each group may have any number of members. Furthermore, any user may be a member of any number of groups. Multiple group membership is especially convenient for users who have interests that span a variety of areas.

If a user is a member of several groups, they can gain access to each group by using the **newgrp(C)** command.

Note that new groups can be created while adding a user account with **mkuser(ADM)**.

Changing a User's Login Group

When a user logs in, the system automatically places the user in the proper “login group”. This is the group given by the group ID in the user's */etc/passwd* file entry (see the section “Adding a User Account” in this chapter). You can change the user's login group by changing the group ID. To change the group ID you need the group ID of the new login group, and you need to know how to use a XENIX text editor (see the *XENIX User's Guide*).

To change the group ID, follow these steps:

1. Log in as the super user.

Δ **sysadmsh** users select: Users→Modify→Group

2. Use the **cd** command to change the current directory to the */etc* directory. Enter:

```
cd /etc
```

3. Use the **cp** command to make a copy of the */etc/passwd* file. Enter:

```
cp passwd passwd+
```

4. Invoke a text editor and specify */etc/passwd+* as the file to edit.
5. Locate the desired user's password entry. Each entry begins with the user's login name.
6. Locate the user's group ID number in the user's password entry. It is the fourth field in the entry. Fields are separated by colons (:). For example, the following entry has group ID 50:

```
nathanb:9iKlwp:205:50:Nathan Brazil,122:/usr/nathanb:/bin/csh
```

7. Delete the old group ID and insert the new one. Be sure you do not delete any other portion of the user's password entry.
8. Exit the editor.

Changing a User's Login Group

9. Use the **mv** command to save the old */etc/passwd* file. Enter:

```
mv passwd passwd-
```

10. Use the **mv** command to make the edited file the new */etc/password* file. Enter:

```
mv passwd+ passwd
```

You can make sure you have entered the new login group correctly by using the **pwcheck(C)** command. If the new entry is correct, no other changes to the file are required.

You must not change the group IDs for system accounts such as "cron" and "root". System accounts are any accounts with user IDs less than 200. The user ID is the third field in the password entry.

Note that changing a user's login group does not change the "group ownership" of the files. Group ownership defines which group has access to a user's files. If users in the new group wish to access the user's files, you must change the group ownership with the **chgrp(C)** (for "change group") command.

Changing a User ID

Sometimes it is necessary to change the user ID in a user's account entry to allow a user to access files and directories transferred from other computers. In particular, if a user has different accounts on different computers and frequently transfers files and directories from one computer to another, the user IDs in each of the account entries must be made the same. You can make them the same by modifying the account entries in the */etc/passwd* file.

To change a user ID, follow these steps at every computer for which the user has an account:

1. Log in as the super user.
2. Use the **cd** command to change the current directory to the */etc* directory. Enter:

```
cd /etc
```

3. Use the **cp** command to make a copy of the */etc/passwd* file. Enter:

```
cp passwd passwd+
```

4. Invoke a XENIX text editor and specify */etc/passwd+* as the file to edit.
5. Locate the user's account entry. Each entry begins with the user's login name.
6. Locate the current user ID. The ID is the third field in the entry. For example, the following entry has a user ID of 205:

```
mavrac:9iKlwp:205:50:Mavra Chang,125:/usr/mavrac:/bin/ksh
```

Substitute the new user ID for the old one.

7. Exit the text editor.

Changing a User ID

8. Use the **mv** command to save the old */etc/passwd* file. Enter:

```
mv passwd passwd-
```

9. Use the **mv** command to make the edited file the new */etc/passwd* file. Enter:

```
mv passwd+ passwd
```

No other changes to the file are required.

In most cases, you can change the user ID to the same number as the user's most-used account. But the new number must be unique on every system for which the user has an account. If there is any conflict (for example, if the number already belongs to another user on one of the systems), you must choose a new number. You can choose any number greater than 200. Make certain it is unique, and that you copy it to all systems on which the user has an account.

Once a user's ID has been changed, you must change the "user ownership" of the user's files and directories from the old user ID to the new one. You can do this with the **chown(C)** (for "change owner") command described in the "Maintaining System Security" chapter.

For example, to change the ownership of mavrac's home directory, enter:

```
chown mavrac /usr/mavrac
```

Note that you may use the **find(C)** command described in the "Managing Filesystems" chapter to locate all files and directories with the user's old user ID.

Removing a User Account

It is sometimes necessary to remove a user account from the system. You can remove a user account with the **rmuser**(ADM) program. The program deletes the user's entry from the */etc/passwd* file and removes the user's home directory and mailbox.

Before you can remove the user account, you must remove all files and directories from the user's home directory, or move them to other directories. If you wish to save the files, you can use the **tar**(C) command to copy the files to a floppy disk (see "Making Backups" in the "House-keeping" chapter of the *XENIX Tutorial*.)

To remove a user account, follow these steps:

1. Log in as the super user.
2. Move to the user's home directory. For example, to change to the home directory for user *gregoryb*, you would enter the following command followed by <Return>:

```
cd /usr/gregoryb
```

If your user directories are in another filesystem (such as */u*) you would use it instead of */usr*.

3. Make sure that you have made copies of all important files and directories in the user's home directory.
4. Use the **rm** (for "remove") command to remove all files and directories from the user's home directory. This includes any files that begin with a period (.). Directories can be removed by using the **-r** (for "recursive") option of the **rm** command. For example, the command:

```
rm -r gregoryb
```

removes the directory named *gregoryb* and all files within this directory.

Removing a User Account

5. After removing all files and directories, make sure the user's mailbox is empty. Enter the following command, substituting the user's login name for *login-name*:

```
l /usr/spool/mail/login-name
```

If the mailbox is not empty (zero length), enter the following command:

```
cat /dev/null > /usr/spool/mail/login-name
```

and press the \langle Return \rangle key.

6. When the user's home directory and mailbox are empty, enter:

```
cd /usr
```

and press \langle Return \rangle . If your user directories are contained in another filesystem, use it instead of */usr*. The user's home directory cannot be removed until you have moved to another directory.

7. Enter the following command, followed by \langle Return \rangle :

```
rmuser
```

Δ **sysadmsh** users select: Users \rightarrow Delete

The program displays a message explaining how to remove a user:

```
rmuser
remove a user from the system

Press RETURN when you are ready.
```

The program then prompts you for the login name of the user you wish to remove:

```
Enter name of id to be removed.
```

8. Enter the user's login name. You should now see the message:

```
Removing user name from the system. CONFIRM? (y/n/q):
```

9. Enter **y** (for “yes”) to remove the user from the system. Otherwise enter **n** (for “no”) to stop the removal, or **q** (for “quit”) to stop the program. The program removes the user’s entry from the */etc/passwd* file, the user’s mailbox, *.profile* file, and home directory. The program displays the message:

```
User name removed from the system
```

The program now gives you a chance to remove another user:

```
Do you want to remove another user? (y/n/q):
```

10. Enter **y** to remove another user. Otherwise, enter **n** or **q** to stop the program.

Note that the **rmuser** program will refuse to remove an account that has a system name, such as *root*, *sys*, *sysinfo*, *cron*, *uucp*, or a system ID (user ID below 200). Also, the program cannot remove a user account if the user’s mailbox still has mail in it, or if the user’s home directory contains files other than *.profile*.

Chapter 11

Building a Remote Network with UUCP

Introduction	11-1
What Is UUCP?	11-1
How to Use This Chapter	11-2
What You Need	11-2
UUCP Commands	11-3
User Programs	11-3
Administrative Programs	11-4
UUCP Directories	11-5
UUCP Background Programs	11-6
How UUCP Works	11-6
A Sample UUCP Transaction	11-7
Configuring UUCP on Your System	11-10
Connecting a UUCP Modem	11-10
Variable Rate Modems	11-10
Editing the UUCP Control Files	11-11
Creating Login Accounts for Sites Dialing-in	11-12
Testing the UUCP Connection	11-13
Detailed Descriptions of UUCP Configuration Files	11-15
Adding Entries for Remote Sites to the Systems File	11-15
The Schedule Field	11-16
The Device Field	11-18
The Speed Field	11-18
The Phone Field	11-18
The Login-Script Field	11-19
Creating Login Scripts	11-20
Setting Up Polling	11-22
Limiting Access with the Permissions File	11-22
Structuring Permissions File Entries	11-22
Permissions File Restrictions	11-23
Permissions Options	11-24
Adding Dial-Out Entries to the Devices File	11-31
The Type Field	11-32
The Speed Field	11-33
The Dialer-Token Field	11-33

Structuring Dialer-Token Entries 11-35
Using the Same Port for Dialing In and Out 11-38

Special UUCP Configuration Options 11-39
 Adding Dialers to the Dialers File 11-39
 Using Dialcodes to Create a Portable Systems File 11-39
 Creating Alternate Control Files with Sysfiles 11-39
 Changing uucico Packet Parameters 11-40
 Preventing Unknown Sites from Logging in 11-40
 Linking Micnet Sites with the UUCP Network 11-40
 Connecting Two Local Systems Using a Direct Wire 11-41
 Choosing a Serial Port 11-41
 Connecting a Serial Cable 11-42

Administering Your UUCP System 11-44
 UUCP Maintenance Shell Scripts 11-44
 Generating Log Reports on UUCP Usage: uulog 11-45
 uulog Options 11-45
 Special uulog Files 11-45
 The UUCP Spool Directory 11-46

Complete UUCP Examples 11-49
 Example 1: System gomer 11-49
 /etc/passwd 11-49
 /etc/group 11-49
 /etc/systemid 11-50
 /etc/ttys 11-50
 /usr/lib/uucp/Devices 11-50
 /usr/lib/uucp/Permissions 11-50
 /usr/lib/uucp/Systems 11-51
 Example 2: System dingbat 11-51
 /etc/passwd 11-51
 /etc/group 11-52
 /etc/systemid 11-52
 /etc/ttys 11-52
 /usr/lib/uucp/Devices 11-52
 /usr/lib/uucp/Permissions 11-53
 /usr/lib/uucp/Systems 11-53
 Sample Commands 11-53

UUCP Error Messages 11-55
 ASSERT Error Messages 11-55
 UUCP STATUS Error Messages 11-58

Introduction

This chapter explains how to use UUCP to build a remote network system for your computer using a normal telephone line and a modem.

UUCP can also be used to connect physically dissimilar UNIX or XENIX systems using a direct serial line. In addition, the UUCP system is a practical solution to the problem of two Micnet networks that cannot be connected because of distance or cost of cable. (See the chapter on “Building a Local Network with Micnet” in this guide.)

Note

UUCP is not a terminal emulation program. If you want to use your modem to dial into another computer and log on, you should refer to the “Using Modems” chapter of this guide and follow the instructions for adding modems.

If you plan to do extensive file transfers between physically separated UNIX and XENIX systems, you should set up a UUCP connection.

What Is UUCP?

The UUCP package permits UNIX and XENIX systems to communicate as part of a remote network. The name UUCP is an acronym for “UNIX-to-UNIX Copy.” The UUCP package consists of a group of programs that provide the following capabilities:

- remote file transfer (**uucp**)
- remote command execution (**uux**)
- mail to and from remote sites (via **mail**)

The XENIX system uses the HoneyDanBer implementation of UUCP. Used primarily over phone lines, UUCP can connect with specific remote machines on a demand or scheduled basis, and by either dialing out or allowing other machines to call in.

Introduction

UUCP uses a batch method to manage communications traffic, storing (or “spooling”) requests for later execution when actual contact is made between systems. When UUCP commands are executed, work files and any data files needed are created in */usr/spool/uucp* and its subdirectories. The program **uucico** scans these directory for the instructions contained in any work files and executes them. Although it is possible to execute commands immediately, most systems call other systems according to a daily schedule (usually during the evenings to reduce connection costs).

How to Use This Chapter

This chapter describes how to build a UUCP system and covers both hardware installation and software configuration. There are also sections on routine maintenance and troubleshooting.

The following is a procedural outline of what must be done to set up your UUCP network:

1. Connect and configure a modem.
2. Configure the UUCP software by editing the configuration files.
3. Create login accounts for any sites that will be calling your system.
4. Test your connections with each remote site.

The most important task of configuring UUCP is the editing of several control files that act as the database for UUCP. The next few sections describe the function of these files, and “Configuring UUCP on Your System” later in this chapter explains the information contained in these files.

What You Need

To install a UUCP communications system on your computer, you need:

- at least one RS-232 serial line (or serial port) on your computer to use for UUCP
- the UUCP and MAIL packages extracted from your XENIX system distribution using **custom**(ADM)
- a modem. Supported modems include models by Hayes, Penril, Ventel, Vadic, Rixon, AT&T, and Telebit. You can supply *Dialers* entries or dialer programs for other modems (for best results, use dialer programs).

- a standard telephone jack for access to the telephone system
- a cable to connect the serial port to the modem

UUCP Commands

UUCP programs are divided into two categories: user programs and administrative programs. The paragraphs that follow describe the programs in each category.

User Programs

The user programs for basic networking are in */usr/bin*. No special permission is needed to use these programs, but it is possible to restrict access to the devices they use. These commands are all described in the “Communicating with Other Sites” chapter of the *XENIX User’s Guide*.

- cu** connects your computer to a remote computer so you can be logged in on both at the same time. You can transfer files or execute commands on either computer without dropping the initial link.
- ct** connects your computer to a remote terminal so the user of the remote terminal can log in. The user of a remote terminal can call the computer and request that the computer call it back. The computer then drops the initial link so that the remote terminal’s modem is available when it is called back.
- uucp** copies files from one computer to another. It creates work files and data files, queues the job for transfer, and calls the **uucico** daemon, which contacts the remote computer.
- uupick** retrieves files placed in */usr/spool/uucppublic/receive* when files are transferred using **uuto**.
- uustat** displays the status of requested transfers (**uucp**, **uuto**, or **uux**). It also provides a means of controlling queued transfers.

Introduction

- uuto** copies files from one computer to a public spool directory on another computer in */usr/spool/uucppublic/receive*. Unlike **uucp**, which lets you copy a file to any accessible directory on the remote computer, **uuto** places the file in an appropriate spool directory, and tells the remote user to pick it up with **uupick**.
- uux** creates the work, data, and execute files needed to execute commands on a remote computer. The work file contains the same information as work files created by **uucp** and **uuto**. The execute files contain the command string to be executed on the remote computer and a list of the data files. The data files are those files required for the command execution.

Administrative Programs

Most of the administrative programs, control files, and scripts are in */usr/lib/uucp*. One exception is **uulog**, which is in */usr/bin*.

- uucheck** checks for the presence of basic networking directories, programs, and support files. It also checks the *Permissions*, *Systems*, and *Devices* files for syntax errors.
- uuclean** cleans up the spool directory. It is normally executed from a shell script called **uudemon.clean**, which can be set up to be run by **cron**.
- uulog** displays the contents of a specified computer's log files. Log files are created for each remote computer your computer communicates with. The log files contain records of each use of **uucp**, **uuto**, and **uux**.
- uutry** tests call-processing capabilities and does a moderate amount of debugging. It invokes the **uucico** daemon to establish the communications link.

UUCP Directories

There are three directories associated with UUCP:

- | | |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>/usr/spool/uucp</i> | This is the working directory for UUCP. Work files, lock files, log files, and all UUCP communications traffic are stored here and in subdirectories. |
| <i>/usr/spool/uucppublic</i> | This is the publically readable or writable target directory used for most file transfers. |
| <i>/usr/lib/uucp</i> | Most of the UUCP programs are stored here, as well as the supporting database or control files. The main user programs, including uux and uucp , are found in <i>/usr/bin</i> . |

The */usr/lib/uucp* directory also contains configuration files for UUCP (distinguished by their capitalized names). The most important to understand are:

- | | |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Systems</i> | contains information needed to establish a link to a remote computer, including the name of the connecting device associated with the remote computer, when the computer can be reached, telephone number, login sequence, and password. |
| <i>Permissions</i> | defines the access level granted to computers when they attempt to transfer files or remotely execute commands on your computer. |
| <i>Devices</i> | contains information concerning the port name, speed, and type of the Automatic Call Units (modems), direct links, and network devices. |

UUCP Background Programs

The **uucp** traffic is managed by three *daemons*, or supervisory programs, that run in the background, handling file transfers and command executions. (The daemons can also be executed manually as commands.)

- uucico** selects the device used for the link, establishes the link to the remote computer, performs the required login sequence and permission checks, transfers data and executes files, logs results, and (if requested) notifies the user by mail of transfer completions. When the local **uucico** daemon calls a remote computer, it “talks” to the **uucico** daemon on the remote computer during the session.
- uuxqt** implements remote program execution. It searches the spool directory for execute files (*X.file*) that were sent from a remote computer. When an *X.file* file is found, **uuxqt** opens it to get the list of data files that are required for the execution. It then checks to see if the required data files are available and accessible. **uuxqt** also verifies that it has permission to execute the requested command.
- uusched** schedules the queued work in the spool directory. Before starting the **uucico** daemon, **uusched** randomizes the order in which remote computers are called.

How UUCP Works

When you enter a UUCP command, the program creates a work file and usually a data file for the requested transfer. The work file contains information required for transferring the file(s). The data file is a copy of the specified source file. After these files are created in the spool directory, the **uucico** daemon is started.

The **uucico** daemon attempts to establish a connection to the remote computer. First it gathers the information required for establishing a link to the remote computer from the *Systems* file. This is how **uucico** knows what type of device to use in establishing the link. Next, **uucico** searches the *Devices* file looking for the devices that match the requirements listed in the *Systems* file. After **uucico** finds an available device, it attempts to establish the link and log in on the remote computer.

When **uucico** logs in on the remote computer, the **uucico** daemon is started on the remote computer. The two **uucico** daemons then negotiate the line protocol to be used in the file transfer(s). The local **uucico** daemon then transfers the file(s) that you are sending to the remote computer. The remote **uucico** places the file in the specified pathname(s) on the remote computer. After your local computer completes the transfer(s), the remote computer may send files that are queued for your local computer. The remote computer can be denied permission to transfer these files with an entry in the *Permissions* file. (This is also affected by directory permissions.) If this is done, the remote computer must establish a link to your local computer to perform the transfers. A remote computer can also request files.

If the remote computer or the device selected to make the connection to the remote computer is unavailable, the request remains queued in the spool directory. If set up to run by **cron**, each hour, **uudemmon.hour** starts the **uused** daemon. When the **uused** daemon starts, it searches the spool directory for the remaining work files, generates the random order in which these requests are to be processed, and then starts the transfer process (**uucico**) described in the previous paragraphs.

A Sample UUCP Transaction

The following steps trace the execution of a **uucp** command:

1. A user on a system called *kilgore* wishes to send a copy of the file *minutes.01.10* to a remote system called *obie*. To accomplish this, the user enters the following command:

```
uucp minutes.01.10 obie!/usr/spool/uucpublic
```

Note that the exclamation point needs to be escaped (preceded by a “\”) if the **csh** is used; the Bourne and Korn shells do not require this.

2. A work file is created in the */usr/spool/uucp/obie* directory, *C.obienxxxx*, where *xxxx* is the job number.
3. The **uused** daemon schedules the request for execution by **uucico**.
4. When the execution time is reached, **uucico** first checks the *Systems* file and confirms that *obie* is a recognized system and that a call is permitted at this time.

- Using the information in the *Systems* file, **uucico** next locates the modem device and tty port associated with it as stored in the *Devices* file.
- Using the phone number in the *Systems* file and the modem type from the *Devices* file, **uucico** uses the appropriate modem commands from the *Dialers* file (or runs a dialer program from the */usr/lib/uucp* directory) to connect to the remote system.

Table 11.1

Example UUCP Control Files (sites: *kilgore* and *obie*)

<i>Systems:</i>	obie Any ACU 2400 14081234567 \ --ogin:-BREAK-ogin: nuucp ssword: mavra
<i>Devices:</i>	ACU tty1A - 2400 dialHA24
<i>Permissions:</i>	LOGNAME= ukilgore MACHINE= kilgore \ READ=/usr/spool/uucppublic:/usr/kilgore \ WRITE=/usr/spool/uucppublic:/usr/kilgore \ REQUEST=no SENDFILES=call \ COMMANDS=rmail:rnews:uucp

- uucico** creates a lock file (*LCK..tty1a*) to lock the serial line, and a lock file (*LCK..obie*) to lock the called system in the directory */usr/spool/uucp*.
- uucico** uses the login sequence and password defined in the *Systems* file to log in to *obie*, whose own **uucico** confirms that *kilgore* is recognized before beginning the actual transaction.
- The calling system, *kilgore*, is said to be the *guest*; the called system, *obie*, is said to be the *host*. The host **uucico** checks the local *Permissions* file to confirm that the guest is authorized to transfer the file.
- The guest (*kilgore*) transmits the file in packets that are checked for errors and retransmitted if garbled. During reception, the file is stored in a temporary file (*TM.xxxx*) in the */usr/spool/uucp* directory. When the transfer is complete, the file is moved to the proper destination, in this case */usr/spool/uucppublic/minutes.01.10*.

11. Each machine records its side of the transaction in log files. For example, *obie* would have the exchange recorded in a file called */usr/spool/uucp/.Log/uucp/kilgore*.
12. Unless the host system *obie* has requests of its own, a hangup request is sent, the connection is terminated, and the lock files are removed.

For remote command execution (via **uux**), an execute *X.file* is created in the */usr/spool/uucp* directory. The **uuxqt** daemon scans this directory for work, checks the *Permissions* file to confirm permission to execute the command, then executes it.

Configuring UUCP on Your System

To configure your UUCP system, you must connect a modem and edit a series of files that contain information about, and control the actions of, the UUCP programs. The UUCP control files are in the */usr/lib/uucp* directory. You can modify these files with a standard text editor. The descriptions found in “Detailed Descriptions of UUCP Control Files” provide details on the structure of these files so that you can create more complex configurations than the examples provided.

Note

After configuring UUCP, if you have any problems initiating transactions, see the section on UUCP in the “Troubleshooting Your System” chapter in this guide for helpful information.

Connecting a UUCP Modem

With a modem, you can communicate with computers over standard phone lines. To configure and install a modem, follow the instructions in “Installing and Using Modems” section of the “Adding Terminals and Modems” chapter of this guide.

Variable Rate Modems

Some modems can determine the connection baud rate from the carrier sent by a remote system. These modems inform the local system of the connection baud rate before issuing the Carrier Detect (CD) signal. The Hayes 2400 dialer supplied with UUCP detects different connection baud rates and informs UUCP and **cu** when it exits with a successful connection.

The speed fields in *Devices* and *Systems* can specify a range of baud rates for a connection. If a dialer supports baud rates from 300 to 2400 baud, enter the baud rate range in the speed field of *Devices* as follows:

300-2400

If a dialer or modem does not allow variable baud rates, place a single baud in the speed field. If a remote system supports several different speeds, place the range of baud rates in the speed field of *Systems*. If the remote system connects at a single baud rate, place that number in *Systems*. UUCP passes the intersection of the *Systems* and *Devices* baud rate ranges to the dialer when connecting. If the dialer connects outside of the baud range, it returns a bad baud rate error. Otherwise, it returns the baud rate of the connection.

Editing the UUCP Control Files

This section is concerned with the configuration or control files that act as the UUCP database. A simple configuration is assumed in this section; more detailed descriptions of the UUCP files are found later in this chapter.

To configure the UUCP files, do the following:

1. Set up the *Systems* file on each machine. Use the following format:

```
sitename Any ACU baud phone# -\r\d-ogin:-\K\d-ogin:-\K\d-ogin:
-\K\d-ogin: uucp word: password
```

In the above lines:

<i>sitename</i>	login name for the opposite site
<i>baud</i>	baud rate used for dial-out
<i>phone#</i>	phone number of opposite site
<i>password</i>	password for uucp account at opposite site

Note

Note that the lines will exceed 80 characters in length, but still should be treated as one line each. In other words, do not press (Return) when the text reaches the right side of the screen, press (Return) only when you have finished typing in the line.

Sitenames should be no longer than seven characters and should contain no 8-bit characters (some sites reject both). In addition, they should not contain control characters, escape sequences, or uppercase letters.

Configuring UUCP on Your System

The baud rate should be set to the highest common baud rate between the modems that will be used. In other words, if the modem on machine A is a 2400 baud modem, and the modem on machine B is a 1200 baud modem, then these both should be set for 1200.

Here is a set of example entries for a pair of sites, kilgore and obie, with 2400 baud modems:

site: obie

```
kilgore Any ACU 2400 5551212 -\r\d-ogin:-\K\d-ogin:-\K\d-ogin:-\K\d-ogin: uucp word: Troutster
```

site kilgore

```
obie Any ACU 2400 5551212 -\r\d-ogin:-\K\d-ogin:-\K\d-ogin:-\K\d-ogin: uucp word: mAvraC
```

There are many other specifications that can be included in the *Systems* file; it is discussed in more detail in “Adding Entries for Remote Systems to the Systems File” later in this chapter.

2. Set up the *Permissions* file on each machine. Add the following text to the bottom of the file `/usr/lib/uucp/Permissions` on each machine, leaving a blank line between any text already in the file and the following new text:

```
MACHINE=site LOGNAME=login \  
COMMANDS=rmail:rnews:uucp \  
READ=/usr/spool/uucppublic:/usr/tmp \  
WRITE=/usr/spool/uucppublic:/usr/tmp \  
SENDFILES=yes REQUEST=yes
```

Note that the LOGNAME must be the login name that the site uses. There are many other options that can be included in the *Permissions* file; it is discussed in more detail in “Limiting Access with the Permissions File” later in this chapter.

Creating Login Accounts for Sites Dialing-in

A dial-in site must provide a login entry for the sites that call it. These entries are placed in the `/etc/passwd` file.

A UUCP login entry has the same form as an ordinary user login entry (see “Administering User Accounts” in this guide), but has a special login directory and login program instead of the normal user directory and shell.

Note

“uucp” should not be used as the name of a uucp user; it is the name of the uucp owner/administrator.

To create a UUCP login entry, use the **mkuser**(ADM) program and follow these steps:

1. Choose a new username and a user ID (identification number) for the UUCP login. The name can be any combination of letters and digits that is no more than eight characters long. The user ID must be an integer in the range 50 to 30000.

Make sure the name and ID are unique. A UUCP login entry must not have the same name or ID as any other login entry.

2. To invoke **mkuser** program, enter:

/etc/mkuser

Follow the program menus and prompts to add the account(s) you wish.

3. The **mkuser** program prompts you to enter a password for the new user. Passwords are optional for UUCP logins.
4. Choose option 5 (uucp) as the shell for a UUCP user; this gives **uucico** as a shell, and a home directory in */usr/spool/uucp*.

Testing the UUCP Connection

To test your UUCP connection, follow these steps:

1. If you are using a Hayes 1200 or compatible, make sure the volume switch on the modem is at an appropriate level. You must be able to hear the modem to carry out this test successfully. Refer to your modem reference manual for the location of this switch.

Configuring UUCP on Your System

2. Ensure that the *Systems* file has an entry for the system you intend to call, and that the *Devices* file has a matching entry for *ttynn*.
3. Start the **uutry** program by entering:

```
/usr/lib/uucp/uutry -x6 sitename
```

4. Listen carefully to the modem. You should hear each digit as the number is dialed, then hear a high-pitched signal when the other modem connects, followed by silence.
5. The dialer automatically disconnects any call that it cannot complete. To break out of the shell created by *uutry*, press $\langle \text{Del} \rangle$ or BREAK. This returns control to the terminal while **uucico** continues to run, sending the output to a file in */tmp* with the name of the system called.
6. If the signal is not present, make certain:
 - the modem is connected to the telephone jack
 - the jack is connected to the phone system
 - the correct phone number is in the *Systems* file
7. If you do not hear the modem dial, make certain:
 - the volume switch is up
 - the modem is connected to the correct serial line and that the cable connection is tight
 - the correct tty line is in the *Devices* file
 - the modem's power is on
 - there are no *LCK..* files in */usr/spool/uucp*.
8. The **uucico** daemon only allows one call to a given system every 10 minutes. You can wait before retrying, or remove the file associated with the site you are calling in the directory */usr/spool/uucp/.Status*. (This file maintains the status of the connection, and its presence prevents a call until the retry is scheduled).

Detailed Descriptions of UUCP Configuration Files

This section includes detailed information on the UUCP database files. Sites with specific needs can design files as needed rather than relying on the simple examples included in this chapter.

Adding Entries for Remote Sites to the Systems File

The *Systems* file (*/usr/lib/uucp/Systems*) contains the information needed by the **uucico** daemon to establish a communications link to a remote computer. Each entry in the file represents a computer that can be called by your computer. In addition, the *Systems* file can be configured to prevent any computer that does not appear in this file from logging in on your computer. More than one entry may be present for a particular computer. The additional entries represent alternative communication paths that can be tried in sequential order.

Note

If you are setting up your system as a dial-in only (passive) site that never initiates calls, you only need to add the names of the systems that will be calling you with the keyword “Never” as in this example:

```
guardian Never
```

Each entry in the *Systems* file has the following format (each field must be separated by a space):

```
sitename schedule device speed phone login-script
```

where:

sitename contains the node name of the remote computer. The name can be any combination of letters and digits that is no more than eight characters long.

Detailed Descriptions of UUCP Configuration Files

<i>schedule</i>	is a string that indicates the day-of-week and time-of-day when the remote computer can be called.
<i>device</i>	is the device type that should be used to establish the communications link to the remote computer.
<i>speed</i>	indicates the transfer speed of the device used in establishing the communications link.
<i>phone</i>	provides the phone number of the remote computer for automatic dialers. If you wish to create a portable <i>Systems</i> file that can be used at a number of sites where the dialing prefixes differ (for internal phone systems), refer to “Using Dialcodes to Create a Portable Systems File” under “Special UUCP Configuration Options” later in this chapter.
<i>login-script</i>	contains login information (also known as a “chat script”).

The Schedule Field

The *schedule* consists of three subfields. The first, *day*, is required. The other two, *time* and *retry*, are optional. The syntax is as follows:

day[*time*][:*retry*]

The *day* subfield can contain the following keywords:

Su Mo Tu We Th Fr Sa	for individual days
Wk	for any weekday (Mo Tu We Th Fr)
Any	for anytime
Never	for a passive arrangement with the remote computer. If the <i>schedule</i> field is Never , your computer never initiates a call to the remote computer. (This field is ignored when you set up polling with uudemon.poll2 ; see “Setting Up Polling” for details.) The call must be initiated by the remote computer. In other words, your computer is in a passive mode with respect to the remote computer (see discussion of <i>Permissions</i> file).

The optional *time* subfield should be a range of times in 24-hour clock format, such as 0800-1230. If no *time* is specified, any time of day is assumed to be allowed for the call. A time range that spans 0000 is permitted. For example, **0800-0600** means all times are allowed other than times between 6 am and 8 am.

For example, the following permits calls on Mondays, Wednesdays, and Fridays between the hours of 9 am and noon (the *schedule* field is in bold-face for clarity):

```
grebe MoWeFr0900-1200 ACU D1200 14087672676 \  
ogin: nuucp ssword: Crested
```

You can also specify more than one set of *day* and *time* entries by separating them with commas. This is useful for more complex specifications. The following example allows calls from 5 pm to 8 am, Monday through Thursday, and calls any time Saturday and Sunday. This example would be an effective way to call only when phone rates are low, if immediate transfer is not critical:

```
gorgon Wk1700-0800, SaSu ACU D1200 14087672676 \  
ogin: nuucp ssword: DontLook
```

The optional subfield, *retry*, is available to specify the minimum time (in minutes) before a retry, following a failed attempt. The subfield separator is a semicolon (;). For example, the following is interpreted as “call any time, but wait at least 9 minutes before retrying after a failure occurs”:

```
Any;9
```

By default, UUCP uses a method called exponential backoff to allow retry of failed calls. UUCP does not allow another call to go through until after the retry time has elapsed. This interval expands exponentially as the number of unsuccessful attempts increases. The *retry* field overrides the exponential backoff algorithm. If you set the *retry* field to 9, for example, UUCP allows another attempt to connect 9 minutes after each failure. The *retry* field cannot be set lower than 5 minutes.

UUCP does not automatically try a failed call again. You must have polling set up as described in “Setting Up Polling” in this chapter or manually invoke **uucico**(ADM). Any files not transferred due to a connection failure are transferred at the next successful connection to that system.

Detailed Descriptions of UUCP Configuration Files

The Device Field

The *device* field selects the device type, in most cases an ACU (Automatic Call Unit). For example, the keyword used in the following field is matched against the first field of *Devices* file entries:

```
Systems:   gorgon Any ACU D1200 14087672676 \  
          ogin: nuucp ssword: DontLook  
Devices:   ACU tty2A - 1200 /usr/lib/uucp/dialHA12
```

The Speed Field

This field can contain a letter and speed (for example, C1200, D1200) to differentiate between classes of dialers (refer to the discussion on the *Devices* file, *speed* field). Some devices can be used at any speed, so the keyword **Any** can be used. However, we recommend that you specify the actual range of speeds that can be used. (If **Any** is used in both *Systems* and *Devices* entries, 1200 is assumed.) For example, this field must match the *speed* field in the associated *Devices* file entry:

```
Systems:   gorgon Any ACU D2400-9600 14087672676 \  
          ogin: nuucp ssword: DontLook  
Devices:   ACU tty1A - 2400-9600 /usr/lib/uucp/dialHA96
```

If information is not required for this field, use a hyphen (-) as a place holder for the field.

The Phone Field

This field provides the phone number used for the modem dialer. The phone number is made up of an optional alphabetic abbreviation and a numeric part. If an abbreviation is used, it must be one that is listed in the *Dialcodes* file. (See “Using Dialcodes to Create a Portable Systems File” in this chapter for details.) For example:

```
Systems:   gorgon Any ACU D1200 CA2676 \  
          ogin: nuucp ssword: DontLook  
Dialcodes: CA 9=408767
```

In this string, an equal sign (=) tells the ACU to wait for a secondary dial tone before dialing the remaining digits. A dash in the string (-) instructs the ACU to pause 2 seconds before dialing the next digit.

Note

Do not use the comma (,) from the Hayes command set in a *Systems* file entry when you wish to indicate a pause. Use hyphens instead.

If your computer is connected to a LAN switch or port selector, you can access other computers that are connected to that switch. The *Systems* file entries for these computers do not have a phone number in the *phone* field. Instead, this field contains the token that must be passed on to the switch so it knows which computer your computer wishes to communicate with. (This is usually just the system name.) The associated *Devices* file entry should have a \D at the end of the entry to prevent translation using the *Dialcodes* entry.

The Login-Script Field

The login-script opens communications between modems, plus recognizes and sends proper login and password sequences. The script is given as a series of space-separated fields and subfields of the following format:

expect send

where *expect* is the string that is received, and *send* is the string that is sent when the *expect* string is received.

The *expect* field can be made up of subfields of the following form:

expect[-subsend-subexpect]...

where the *subsend* is sent if the prior *expect* is not successfully read and the *subexpect* following the *subsend* is the next expected string. To make this distinction clear: the send-expect sequence sends a string if the expect string is received, the subsend-subexpect sends only if the prior expect string is not received within 10 seconds.

For example, with "login:--login:", the UUCP program expects "login:". If a "login:" is received, it goes on to the next field. If it does not get "login:", it sends nothing followed by a carriage return, then looks for "login:" again. If no characters are initially expected from the remote computer, the characters "" (null string) should be used in the first *expect*

Detailed Descriptions of UUCP Configuration Files

field. Note that all *send* fields are sent followed by a carriage return unless the *send* string is terminated with a `\c`.

If an *expect* string starts with a dash, it is interpreted as a null *expect* string followed by a *subsend* string. For example, “--login:” sends a carriage return and then expects a “login:”.

The *expect* string need not be complete; only the trailing characters must be specified, as in “ogin:”. This avoids difficulties with login strings that use an uppercase letter as in “Login:” or “Password:”, and also difficulties when the line is shared by dial-in and dial-out.

Creating Login Scripts

This section explains in greater detail how to create a login (chat) script.

Consider the following sample *Systems* file entry:

```
terps Any ACU 1200 18005211980 "" \r ogin:-BREAK-ogin: \  
uucpx word: ichore
```

This is how this script would work during connection:

1. Nothing is expected initially.
2. A carriage return is sent and the script waits for the prompt “ogin:” (login:).
3. If it does not receive “ogin:”, send a BREAK signal.
4. When “ogin:” is finally received, send the login name *uucpx*.
5. When the prompt “word:” (for Password:) is received, send the password *ichore*.

Login (chat) scripts often require some experimentation. There are cases that require one or more BREAK sequences before presenting a login (this is often true with variable speed modems). If you cannot obtain the necessary login sequence from the system administrator for a given site, it is a good idea to connect with the site manually. You can accomplish this using *cu* and find out what must be sent to generate a login prompt. (You

can also connect with a system using a **uutry** for debugging; see the “Troubleshooting Your System” chapter for details.) There are several escape characters that cause specific actions when sent during the login sequence, some of which correspond to keystrokes; these should be included in the script where necessary. See Table 11.2.

Table 11.2
Login (Chat) Script Escape Sequences

Character	Description
<code>\N</code>	sends a null character (ASCII NUL).
<code>\b</code>	sends or expects a backspace character.
<code>\c</code>	if at the end of a string, suppresses the carriage return that is normally sent. Ignored otherwise.
<code>\d</code>	delays two seconds before sending or reading more characters.
<code>\p</code>	pauses for approximately $\frac{1}{4}$ to $\frac{1}{2}$ second.
<code>\E</code>	starts echo checking. (after this sequence is used, whenever a character is transmitted, the system waits for the character to be received before doing anything else.)
<code>\e</code>	turns echo check off.
<code>\n</code>	sends or expects a newline character.
<code>\r</code>	sends or expects a carriage-return.
<code>\s</code>	sends or expects a space character.
<code>\t</code>	sends or expects a tab character.
<code>\\</code>	sends or expects a <code>\</code> character.
<code>EOT</code>	sends EOT (end of transmission or <code><Ctrl>d</code>)
<code>BREAK</code>	sends a BREAK signal.
<code>\K</code>	same as BREAK.
<code>\ddd</code>	collapses the octal digits (ddd) into a single character whose value is the ASCII character represented by that number (for example: <code>\007</code>).
<code>""</code>	expects a null string.

Setting Up Polling

Use **uudemon.poll2** to set up polling. To run **uudemon.poll2**, you need an entry for the daily daemon and an entry for the hourly daemon in the */usr/spool/cron/crontabs/root* file as follows:

```
0 0 * * * uudemon.poll2 -d
0 * * * * uudemon.poll2
```

The **-d** flag refers to the daily daemon. The above example has the daemon run at midnight. You can change time the daemon runs by altering the second field using a 24-hour clock. The hourly daemon has no flags.

To establish the hours and days that **uudemon.poll2** runs, you create two files: */usr/lib/uucp/Poll.hour* and */usr/lib/uucp/Poll.day*. These files contain the systems to be polled and the times and days they are polled. A sample *Poll.hour* file follows:

```
hanna 12 1 3
raven 2 6 10w
```

If the hour is followed by a “w”, **uudemon.poll2** calls the site only if there is work to be done.

A sample *Poll.day* file follows:

```
hanna 1 3 6
raven 1 2 3 4 5
```

The days of the week are integers where Sunday is 0.

Limiting Access with the Permissions File

If other machines will be dialing into your system, the *Permissions* file (*/usr/lib/uucp/Permissions*) specifies the permissions that remote computers have with respect to login, file access, and command execution. There are options that restrict the remote computer’s ability to request files and its ability to receive files queued by the local site. Other options specify the commands that a remote site can execute on the local computer.

Structuring Permissions File Entries

Each entry is a logical line with physical lines terminated by a \ to indicate continuation. Entries are made up of options delimited by spaces. Each option is a name-value pair in the following format:

```
name=value
```

Note that no spaces are allowed within an option assignment.

Comment lines begin with a crosshatch sign (#) and they occupy the entire line up to a newline character. Blank lines are ignored (even within multi-line entries).

There are two types of *Permissions* file entries:

LOGNAME specifies the permissions that take effect when a remote computer calls your computer.

MACHINE specifies permissions that take effect when your computer calls a remote computer.

In this way it is possible not only to define permissions for sites calling your system, but permissions for when your site calls other machines.

Permissions File Restrictions

When using the *Permissions* file to restrict the level of access granted to remote computers:

- A machine cannot have more than one LOGNAME entry.
- Any site that is called whose name does not appear in a MACHINE entry, has the following default permissions or restrictions:
 - Only local send and receive requests are executed.
 - The remote computer can send files to your computer's */usr/spool/uucppublic* directory.
 - The commands sent by the remote computer for execution on your computer must be one of the default commands, usually **rmail**.

Note

When a remote machine calls you, unless you have a unique login and password for that machine, you do not know if the machine is who it claims to be.

Permissions Options

This section describes each option, specifies how they are used, and lists their default values.

REQUEST

specifies whether the remote computer can request to set up file transfers from your computer. When a remote computer calls your computer and requests to receive a file, this request can be granted or denied. The following string specifies that the remote computer can request to transfer files from your computer:

```
REQUEST=yes
```

The following string specifies that the remote computer cannot request to receive files from your computer:

```
REQUEST=no
```

The no value is the default value. It is used if the REQUEST option is not specified. The REQUEST option can appear in either a LOGNAME (remote calls you) entry or a MACHINE (you call remote) entry.

SENDFILES

specifies whether your computer can send the work queued for the remote computer. When a remote computer calls your computer and completes its work, it may attempt to take work your computer has queued for it.

The following string specifies that your computer can send the work that is queued for the remote computer as long as it is logged in as one of the names in the LOGNAME option:

```
SENDFILES=yes
```

This string is mandatory if your computer is in a passive mode with respect to the remote computer.

The following string specifies that files queued in your computer be sent only when your computer calls the remote computer:

```
SENDFILES=call
```

The call value is the default for the SENDFILE option. This option is only significant in LOGNAME entries because MACHINE entries apply when calls are made out to remote computers. If this option is used with a MACHINE entry, it is ignored.

READ and WRITE

specify the various parts of the filesystem that **uucico** can read from or write to. The READ and WRITE options can be used with either MACHINE or LOGNAME entries.

The default for both the READ and WRITE options is the *uucpublic* directory as shown in the following strings:

```
READ=/usr/spool/uucppublic  
WRITE=/usr/spool/uucppublic
```

The following strings specify permission to access any file that can be read by UUCP.

```
READ=/ WRITE=/
```

The value of these entries is a colon-separated list of pathnames. The READ option is for requesting files, and the WRITE option for depositing files. One of the values must be the prefix of any full pathname of a file coming in or going out.

Detailed Descriptions of UUCP Configuration Files

To grant permission to deposit files in */usr/tmp* as well as the public directory, the following values would be used with the WRITE option:

```
WRITE=/usr/spool/uucppublic:/usr/tmp
```

It should be pointed out that if the READ and WRITE options are used, all pathnames must be specified because the pathnames are not added to the default list. For instance, if the */usr/news* pathname was the only one specified in a WRITE option, permission to deposit files in the public directory would be denied.

You should be careful with which directories you make accessible for reading and writing by remote systems. For example, you probably do not want remote computers to be able to write over your */etc/passwd* file so */etc* should not be open to writes.

Note

READ and WRITE options do not affect the actual permissions of a file or directory. For example, a directory with permissions of 700 only permits the owner to access it, and cannot be read or written by UUCP, no matter what access options are defined in the *Permissions* file.

NOREAD and NOWRITE

specify exceptions to the READ and WRITE options or defaults. The following strings would permit reading any file except those in the */etc* directory (and its subdirectories—remember, these are prefixes) and writing only to the default */usr/spool/uucppublic* directory:

```
READ=/
WRITE=/usr/spool/uucppublic
NOREAD=/etc
NOWRITE=/etc
```

NOWRITE works in the same manner as the NOREAD option. The NOREAD and NOWRITE options can be used in both LOGNAME and MACHINE entries.

CALLBACK

specifies in LOGNAME entries that no transaction takes place until the calling system is called back. There are two examples of when you would use CALLBACK. From a security standpoint, if you call back a machine you can be sure it is the machine it says it is. If you are doing long data transmissions, you can choose the machine that is billed for the longer call.

The following string specifies that your computer must call the remote computer back before any file transfers take place:

```
CALLBACK=yes
```

The default for the CALLBACK option is:

```
CALLBACK=no
```

The CALLBACK option is rarely used. If two sites have this option set for each other, a conversation never gets started.

COMMANDS

specifies the commands in MACHINE entries that a remote computer can execute on your computer. This affects the security of your system; use it with extreme care.

The **uux** program generates remote execution requests and queues them to be transferred to the remote computer. Files and a command are sent to the target computer for remote execution. Note that COMMANDS is not used in a LOGNAME entry; COMMANDS in MACHINE entries define command permissions whether you call the remote system or it calls you.

The default command that a remote computer can execute on your computer is:

```
COMMANDS=rmail
```

If a command string is used in a MACHINE entry, the default commands are overridden. For instance, the following entry overrides the COMMAND default so that the computers *owl*, *raven*, *hawk*, and *dove* can now execute **rmail**, **rnews**, and **lp** on your computer:

```
MACHINE=owl:raven:hawk:dove \  
COMMANDS=rmail:rnews:lp
```

Detailed Descriptions of UUCP Configuration Files

Full pathnames of commands can also be used. For example, the following command specifies that command **rmail** uses the default path:

```
COMMANDS=rmail:/usr/lbin/rnews:/usr/local/lp
```

The default paths for your computer are */bin*, */usr/bin*, and */usr/lbin*. When the remote machine specifies **rnews** or */usr/lbin/rnews* for the command to be executed, */usr/lbin/rnews* is executed regardless of the default path. Likewise, */usr/local/lp* is the **lp** command that is executed.

Including the **ALL** value in the list means that any command from the remote computer specified in the entry is executed. If you use this value, you give the remote computer full access to your computer. So, be careful; this allows far more access than normal users have.

The following string illustrates two points:

```
COMMANDS=/usr/local/bin/lc:ALL:/usr/local/lp
```

1. The **ALL** value can appear anywhere in the string; and the pathnames specified for **lc** and **lp** are used (instead of the default) if the requested command does not contain the full pathnames for **lc** or **lp**.
2. The **VALIDATE** option should be used with the **COMMANDS** option whenever potentially dangerous commands like **cat** and **uucp** are specified with the **COMMANDS** option. Any command that reads or writes files is potentially dangerous to local security when executed by the UUCP remote execution daemon (**uuxqt**).

VALIDATE

is used in conjunction with the **COMMANDS** option in **LOG-NAME** entries when specifying commands that are potentially dangerous to your computer's security. It provides a certain degree of verification of the caller's identity. The use of the **VALIDATE** option requires that privileged computers have a unique login or password for UUCP transactions. An important aspect of this validation is that the login or password associated with this entry be protected. If an outsider gets that information, that particular **VALIDATE** option can no longer be considered secure. (**VALIDATE** is merely an added level of security to the **COMMANDS** option, though it is a more secure way to open command access than **ALL**.)

Careful consideration should be given to providing a remote computer with a privileged login and password for UUCP transactions. Giving a remote computer a special login and password with file access and remote execution capability is like giving anyone on that computer a normal login and password on your computer. Therefore, if you cannot trust someone on the remote computer, do not provide that computer with a privileged login and password.

The following LOGNAME entry specifies that if one of the remote computers that claims to be *eagle*, *owl*, or *hawk* logs in on your computer, it must have used the login *uucpfriend*:

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

As can be seen, if an outsider gets the *uucpfriend* login or password, masquerading is trivial.

VALIDATE increases security by linking the MACHINE entry (and COMMANDS option) with a LOGNAME entry associated with a privileged login. This link is needed because the execution daemon is not running while the remote machine is logged in. In fact, it is an asynchronous process with no knowledge of what machine sent the execution request. Therefore, the real question is how does your system know where the execution files came from?

Each remote computer has its own *spool* directory on your computer. These spool directories have write permission given only to UUCP programs. The execution files from the remote computer are put in its spool directory after being transferred to your computer. When the **uuxqt** daemon runs, it can use the spool directory name to find the MACHINE entry in the *Permissions* file and get the COMMANDS list. If the computer name does not appear in the *Permissions* file, the default list is used.

Detailed Descriptions of UUCP Configuration Files

The following example shows the relationship between the MACHINE and LOGNAME entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \  
COMMANDS=rmail:/usr/local/bin/lc \  
READ=/ WRITE=/  
  
LOGNAME=uucpz VALIDATE=eagle:owl:hawk \  
REQUEST=yes SENDFILES=yes \  
READ=/ WRITE=/
```

The COMMANDS option line shows that remote mail and */usr/local/bin/lc* can be executed by remote users.

In the MACHINE entry, you must make the assumption that when you want to call one of the computers listed, you are really calling *eagle*, *owl*, or *hawk*. Any files put into one of the *eagle*, *owl*, or *hawk* spool directories is put there by one of those computers. If a remote computer logs in and says that it is one of these three computers, its execution files are also put in the privileged spool directory. You should validate that the computer has the privileged login *uucpz*.

Entries for OTHER Systems

You may want to specify different option values for machines or logins that are not mentioned in specific MACHINE or LOGNAME entries. This may occur when there are many computers calling in that have the same set of permissions. The special name OTHER for the computer name can be used in a MACHINE or LOGNAME entry as follows:

```
MACHINE=OTHER \  
COMMANDS=rmail:/usr/local/bin/lc  
  
LOGNAME=OTHER \  
REQUEST=yes SENDFILES=yes \  
READ=/usr/spool/uucppublic \  
WRITE=/usr/spool/uucppublic
```

All options that can be set for specific machines or logins can be used with the OTHER value, although the use of the VALIDATE option makes little sense.

Combining MACHINE and LOGNAME Entries

It is possible to combine MACHINE and LOGNAME entries into a single entry where the common options are the same. For example, the following two entries share the same REQUEST, READ, and WRITE options:

```
MACHINE=eagle:owl:hawk REQUEST=yes \  
  READ=/ WRITE=  
  
LOGNAME=uucpz REQUEST=yes SENDFILES=yes \  
  READ=/ WRITE=/
```

These two entries can be merged as follows:

```
MACHINE=eagle:owl:hawk REQUEST=yes \  
LOGNAME=uucpz SENDFILES=yes \  
  READ=/ WRITE=/
```

Adding Dial-Out Entries to the Devices File

The *Devices* file (*/usr/lib/uucp/Devices*) contains information for all the devices that can be used to establish a link to a remote computer. Devices are Automatic Call Units, direct links, or network connections. This file works closely with the *Dialers*, *Systems*, and *Dialcodes* files. Before you make changes in any of these files, you should be familiar with them all. A change to an entry in one file may require a change to a related entry in another file.

Each entry in the *Devices* file has the following format:

type ttyline dialerline speed dialer-token

where:

- | | |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>type</i> | contains one of two keywords (direct or ACU), the name of a Local Area Network switch, or a system name. |
| <i>ttyline</i> | contains the device name of the port associated with the <i>Devices</i> entry. For example, if the automatic dial modem for a particular entry was attached to the <i>/dev/tty1A</i> line, the name entered in this field would be <i>tty1A</i> . |
| <i>dialerline</i> | is useful only for 801 type dialers, which do not contain a modem and must use an additional line. Unless you have an 801 dialer, simply enter a hyphen (-) as a placeholder. |

Detailed Descriptions of UUCP Configuration Files

speed is the speed or speed range of the device. Can also contain an indicator for distinguishing different dialer classes.

dialer-token contains pairs of dialers and tokens, each representing a dialer and an argument to be passed to it. The *dialer* portion can be the name of an automatic dial modem, or **Direct** for a direct link device.

The Type Field

This field contains one of two keywords (**Direct** or **ACU**), the name of a Local Area Network switch, or a system name:

Direct indicates a direct link to another computer or a switch for **cu** connections.

ACU indicates that the link to a remote computer is made through an Automatic Call Unit. This modem can be connected either directly to your computer or indirectly through a Local Area Network (LAN) switch.

LANswitch can be replaced by the name of a LAN switch. **micom** and **develcon** are supplied with caller scripts in the *Dialers* file.

sysname indicates a direct link to a particular computer. (*sysname* is replaced by the name of the computer.) This means that the line associated with this *Devices* entry is for a particular computer in the *Systems* file.

For example, the keyword **gorgon** used in the *type* field of the *Devices* file is matched against the third field of the *Systems* file entry:

```
Devices:   gorgon tty1A - 1200 hayes1200
```

```
Systems:  gorgon Any ACU 1200 14087672676 ogin: nuucp \  
            ssword: DontLook
```

The Speed Field

In most cases, this is simply the speed of the device, if the keyword **ACU** or **Direct** is used in the *type* field. However, *speed* can contain a letter and a speed (for example, C1200, D1200) to differentiate between classes of dialers (Centrex or Dimension PBX). This is necessary because many larger offices may have more than one type of telephone network: one network may be dedicated to serving only internal office communications, while another handles the external communications. It is necessary to distinguish which lines are used for internal communications and which are used for external communications. The keyword used in the *speed* field of the *Devices* file is matched against the fourth field of the *Systems* file entries, for example:

```
Devices:  ACU tty1A - D1200 hayes1200

Systems:  gorgon Any ACU D1200 3251 ogin: nuucp \
          ssword: DontLook
```

Some devices can be used at any speed, so the keyword **Any** can be used in the *speed* field. If **Any** is used, the line matches any speed requested in a *Systems* file entry. If this field is **Any** and the *Systems* file *speed* field is **Any**, the speed defaults to 1200 bps. If a device can be used at a range of speeds, then the speed field can specify this range (for example, 1200-9600 or D1200-9600). This is preferable to the use of **Any**.

The Dialer-Token Field

This field has the following format:

```
dialer [ token dialer token ... ]
```

For a direct line, this field contains simply the word **direct**, and no token is required.

For a simple connection to a dialer, this field contains the name of the dialer, and the token is omitted; by default it is taken from the phone number field of the *Systems* file entry.

For a dialer or a network dataswitch, this field contains the name of an entry found in the *Dialers* file (**develcon** and **micom** are examples of network data switches). Other dialer types are supported by binaries instead of *Dialers* entries. (Support for 801-type dialers is provided through use

Detailed Descriptions of UUCP Configuration Files

of separate lines for data and the dialer. See the *Devices* file for details.) UUCP recognizes a dialer as a binary if the name begins with a “/” or there is an executable file by that name in */usr/lib/uucp*.

Table 11.3 lists dialer types that are available as *Dialers* entries.

Table 11.3
Dialers File Entries

Dialer Type	Modem or Data Switch
Direct	direct line; no dialer
Penril	Penril modem
Hayes	Hayes modem (or compatible)
Ventel	Ventel 212+ modem
Vadic	Racal Vadic 3451 modem
LANswitch	network switch described in type field
Hayes1200	Hayes Smartmodem 1200
Hayes2400	Hayes Smartmodem 2400
Develcon	Develcon network dataswitch
Micom	Micom network dataswitch
Rixon	Rixon Intelligent Modem
ATT4000	AT&T Programmable 300/1200 Modem Model 4000
ATT2212c	AT&T DATAPHONE II 2212C Modem
ATT2224	AT&T DATAPHONE II 2224 Modem
NLS	Network Listener Service

Note

For best results, dialer programs are preferred over *Dialers* entries. The following entry is an example of an entry using a dialer binary:

```
ACU ttynn - 300-2400 /usr/lib/uucp/dialHA24
```

The TLI and TLIS dialer types are currently not available.

Table 11.4 lists the binary types available in *usr/lib/uucp*.

Table 11.4
Dialer Binaries

Binary File	Modem
dialHA12	Hayes Smartmodem 1200 or compatible
dialHA24	Hayes Smartmodem 2400 or compatible
dialHA96V	Hayes Smartmodem 9600 or compatible
dialMUL	Multitech Multimodem 224 EH
dialVA3450	Racal Vadic 3451 modem
dialVA96	Racal Vadic 9600 modem
dialTBIT	Telebit Trailblazer Modem

The source is provided for these dialer binaries; you can adapt and compile your own dialers if desired. See the section “Dialing Out from Your Computer” in the “Using Modems” chapter of this guide for details.

Structuring Dialer-Token Entries

The *dialer-token* can be structured four different ways, depending on the device associated with the entry:

- Simple modem connection

If an automatic dialing modem is connected directly to a port on your computer, the *dialer-token* field of the associated *Devices* file entry only has one pair. This pair would normally be the name of the modem. This name matches the particular *Devices* file entry with an entry in the *Dialers* file. Therefore, the *dialer* field must match the first field of the following *Dialers* file entry:

```
Devices: ACU tty1A - 1200 ventel
```

```
Dialers: ventel =&-% "" \r\p\r\c $ <K\T%\r>\c ONLINE!
```

Detailed Descriptions of UUCP Configuration Files

Notice that only the *dialer* portion (**ventel**) is present in the *dialer-token* field of the *Devices* file entry. This means that the *token* to be passed on to the dialer (in this case the phone number) is taken from the *Phone* field of a *Systems* file entry. (\T is implied; see the last item, "Modems used with a local network switch.") Backslash sequences are described later.

- Direct links

If a direct-link is established to a particular computer, the *dialer-token* field of the associated entry contains the keyword **direct**. This is true for both types of direct link entries, **direct** and *sysname* (refer to discussion on the *type* field).

- Local network switches

If a computer that you wish to communicate with is on the same local network switch as your computer, your computer must first access the switch and the switch can make the connection to the other computer. In this type of entry, there is only one pair. The *dialer* portion matches a *Dialers* file entry, as shown in the following example:

```
Devices:  develcon tty13 - 1200 develcon \D
Dialers:  develcon "" "" \pr\ps\c est:\007 \E\D\e \007
Systems:  obie Any ACU 1200 obie \
          --ogin:-BREAK-ogin: nuucp ssword: mavra
```

As shown, the *token* portion is \D, which indicates that it is retrieved from the *Systems* file without translation. The *Systems* file entry for this particular computer contains the token in the *phone* field; this is normally reserved for the phone number of the computer (refer to *Systems* file, *phone* field). The \D ensures that the contents of the *phone* field is not interpreted as a valid entry in the *Dialcodes* file.

- Modems used with a local network switch

If an automatic dialing modem is connected to a switch, your computer must first access the switch and the switch makes the connection to the automatic dialing modem. This type of entry requires two *dialer-token-pairs*. The following *dialer* portion of each pair (fifth and seventh fields of entry) are used to match entries in the *Dialers* file:

```
Devices:   ACU tty14 - 1200 develcon vent ventel

Dialers:   develcon "" "" \pr\ps\c est:\007 \E\D\e \007
           ventel =&-% "" \r\p\r\c $ <K\T%\r>\c ONLINE!
```

In the first pair, **develcon** is the switch and **vent** is the token that is passed to the **develcon** switch to tell it which device to connect to your computer. This token would be unique for each LAN switch because each switch can be set up differently. Once the modem is connected, the second pair is accessed, where **ventel** is the dialer and the token is retrieved from the *Systems* file.

The following are two escape characters that can appear in the *dialer-token* field:

- \T indicates that the *Phone* field should be translated at this stage, using the *Dialcodes* file. This escape character is normally placed in the *Dialers* file for each caller script associated with an automatic dial modem (penril, ventel, and so on). The translation does not take place until the caller script is accessed.
- \D indicates that the *Phone* field should not be translated using the *Dialcodes* file. If no escape character is specified at the end of a *Devices* entry, \D is assumed by default when a *Dialers* script is to be used (which can itself contain a \T to translate the number). \T is assumed if a built-in or dialer binary is to be used (because there is then no later opportunity to translate the number).

Using the Same Port for Dialing In and Out

It is possible to dial in and out on the same line without enabling/disabling the line or running a special version of **getty**. All that is necessary is to first create an entry for a line in the *Devices* file (dial-out) with ACU in the type field and then an entry in *etc/ttyS* (dial-in) for the same line. When access to a dial-out line is requested on a shared port, **getty** runs a special program, **uchat**, that automatically reinitializes the port when the call is complete. **uchat** uses special dialer scripts found in the *Dialers* file that begin with an ampersand. This means there are actually two entries for some dialers. For example, the dialer for the Hayes Smartmodem 2400 (or compatible) consists of two entries: **hayes2400** and **&hayes2400**, the latter of which is used when reinitializing a shared port to dial-in. In the case of the dialer binaries in */usr/lib/uucp*, these programs are automatically invoked with the **-h** (hangup) switch that reinitializes the port to dial-in.

Special UUCP Configuration Options

This section contains several options that are used for special circumstances and can be ignored in most cases.

Adding Dialers to the Dialers File

The *Dialers* file (*/usr/lib/uucp/Dialers*) specifies the initial conversation that must take place on a line before it can be made available for transferring data. New entries can be added to this file if your modem does not appear in the file. See the **dialers(F)** manual page for more information.

A modem that is used for dialing in and out may require a second *Dialers* entry. This is to reinitialize the line to dial-in after it has been used for dial-out. The name of the dial-in version of a dialer must begin with an ampersand. For example, the *Dialers* file contains a **hayes2400** and a **&hayes2400** entry.

Using Dialcodes to Create a Portable Systems File

The *Dialcodes* file (*/usr/lib/uucp/Dialcodes*) contains the dial-code abbreviations that can be used in the *Phone* field of the *Systems* file. This feature is intended primarily for those who wish to create a standard *Systems* file for distribution among several sites that have different phone systems and area codes. As such, the *Dialcodes* file is probably not necessary for most sites. See the **dialcodes(F)** manual page for more information.

Creating Alternate Control Files with Sysfiles

The */usr/lib/uucp/Sysfiles* file lets you assign different files to be used by **uucp** and **cu** as *Systems*, *Devices*, and *Dialers* files. See the **sysfiles(F)** manual page for more information.

Changing uucico Packet Parameters

An added feature is the ability to change two specialized parameters contained in the **uucico** program without having to recompile the source. These parameters are:

- windows** specifies the size of window that the sliding-window protocol should use.
- pktime** is the number of seconds **uucico** should wait before giving up and re-transmitting the packet being sent.

These values are modified via the **scopatch** command. See the **scopatch(ADM)** manual page for more information.

Preventing Unknown Sites from Logging in

The script **remote.unknown** is executed when a site whose name is not recognized dials in to your system. It logs the conversation attempt and fails to make a connection. If you wish to allow such “unknown” systems to log in to your system, you can change the permissions of this file so it cannot execute and your system accepts any communication requests. To do so, enter the following commands while logged in as *root*:

```
cd /usr/lib/uucp
chmod 000 remote.unknown
```

Linking Micnet Sites with the UUCP Network

To use a UUCP system with your **micnet** network, follow these steps:

1. Add the following entry to the *aliases* file of the computer on which the UUCP system is installed:

```
uucp:
```

2. For all other computers in your site, add the following entry to the *aliases* file:

```
uucp:machine-name:
```

where *machine-name* is the name of the computer on which the UUCP system is installed. This longer form of entry can also be used on the computer on which the UUCP system is installed.

You can test the UUCP system by mailing a short letter to yourself by means of another site. For example, if you are on the site *chicago*, and there is another **micnet** site named *seattle* in the system, then the following command sends mail to the *seattle* site, then back to your *chicago* site, and finally to the user *johnd* in your **micnet** network:

```
mail seattle!chicago!johnd
```

Note that a UUCP system usually performs its communication tasks according to a fixed schedule and may not return mail immediately.

Connecting Two Local Systems Using a Direct Wire

This section describes how to install a direct wire between two computers. To connect two computers with a direct wire, you need to do the following:

- Choose a serial port on each machine.
- Connect a serial wire (RS-232) between the two machines, using the chosen serial ports.
- Decide which machine is the dial-in site and which is the dial-out site. The dial-out site calls and logs in to the dial-in site.

Choosing a Serial Port

On each machine, you must choose the RS-232 serial port (*/dev/tty nn*) you want to use. If there are no ports available, you must install a new serial line or make one available by removing any device connected to it. If you remove a terminal, make sure no one is logged in.

Find the name of the device special file associated with the line. (Refer to the “XENIX Directories and Special Device Files” chapter in this guide.) The device name should have the form:

```
/dev/tty $nn$ 
```

where *nn* is the number of the corresponding line. For example, */dev/tty1a* usually corresponds to COM1. You need the name of the actual line for later steps.

Special UUCP Configuration Options

The serial port should be owned by *uucp*. To make sure the line is owned by *uucp*, enter this command:

```
chown uucp /dev/tty $nn$ 
```

where *nn* is the number of the corresponding line.

In addition, the *Devices* file entry on each machine should look like this:

```
Direct  tty $la$  - 9600 direct
```

Connecting a Serial Cable

You connect two computers together using an RS-232 cable. The actual pin configurations sometimes vary between machines.

Typically, the cable should connect pins 2, 3, and 7 on one computer to the same pins on the second computer. Sometimes the cable must be *nulled*, which means that pin 2 on one machine is connected to pin 3 on the other, and vice versa. Because the connections can vary, check the hardware manuals for each computer to determine the proper pin connections.

Testing a Connection

For this section, *tty2a* is used as the example serial port for both machines.

To test the wire connection between two machines:

1. Disable the serial lines on each machine. On each computer, enter the command:

```
disable /dev/tty2a
```

Be sure to disable the modem control line as well:

```
disable /dev/tty2A
```

2. Attach one end of the serial wire to one of the machines. Attach the other end to the standard data port of a terminal.

3. Enter this command at your computer:

```
(stty 9600; date ) < /dev/tty2a > /dev/tty2a
```

tty2a is our example serial line, and the `date` command provides sample output.

You should see the output of the `date` command appear on the terminal screen. Repeat this procedure on the other machine.

If this does not work, check the following:

- The wire is plugged in properly at each end.
- The continuity of the wire is fine.
- The terminal is configured correctly (baud rate, parity, and so on).
- The serial line is disabled.
- You are using the correct pin numbers.

Note

An unterminated serial cable can cause serious system problems. Do not leave serial cables dangling.

Administering Your UUCP System

This section discusses the various shell scripts that supervise and maintain UUCP. Consult the section on “Administration and Maintenance Commands” for details on all commands available to the system administrator. Included is an extended description of the */usr/spool/uucp* work directory and a special subsection on troubleshooting.

UUCP Maintenance Shell Scripts

There are several aspects of system operation that are governed by shell scripts running as daemons:

- checking the UUCP directory for work (**uudemon.hour**)
- polling of sites that are passive or do not originate calls (**uudemon.poll2**)
- sending of status information to the UUCP administrator (**uudemon.admin**)
- cleaning of the UUCP spool directory (**uudemon.clean**)

These scripts must be set up by the system administrator. See **uudemon(ADM)** for complete instructions.

Note

You should not run **uudemon.hour** if you are using polling with **uudemon.poll2**.

Generating Log Reports on UUCP Usage: uulog

The **uulog** program displays log information on UUCP usage according to remote machine. All usage of the programs **uucp**, **uuto**, and **uux** are logged in special log files, one per machine.

uulog Options

The **uulog** command has the following options:

Option	Description
-f <i>system</i>	displays the last entry or entries of the <i>system</i> file transfer log.
-s <i>system</i>	displays the <i>system</i> file transfer information.
-x <i>system</i>	displays the uuxqt log file for the given system.
-number	specifies the <i>number</i> of lines displayed by the -f option.

For example, to print the last 10 lines of *chicago*'s file-transfer log, you would enter:

```
uulog -fchicago -10
```

Special uulog Files

During execution of the **uulog** program, the files from the following directories are examined:

Directory	Description
<i>/usr/spool/uucpl.Log/uucicol*</i>	directory used for queries by the uucp program
<i>/usr/spool/uucpl.Log/uuxqt*</i>	directory used for queries by the uuxqt program

The UUCP Spool Directory

The following is a comprehensive discussion of all files and subdirectories of the UUCP spool directory. These files are created in spool directories to lock devices, hold temporary data, or keep information about remote transfers or executions.

TM. (temporary data file)

These data files are created by UUCP processes under the spool directory (that is, */usr/spool/uucp/system*) when a file is received from another computer. The *system* directory has the same name as the remote computer that is sending the file. The names of the temporary data files have the format:

TM.pid.ddd

where *pid* is a process-ID and *ddd* is a sequential three-digit number starting at 0.

When the entire file is received, the *TM.pid.ddd* file is moved to the pathname specified in the *C.sysnxxx* file (discussed below) that caused the transmission. If processing is abnormally terminated, the *TM.pid.ddd* file may remain in the *system* directory. These files should be automatically removed by **uuclean**.

LCK. (lock file)

Lock files are created in the */usr/spool/uucp* directory for each device in use. Lock files prevent duplicate conversations and multiple attempts to use the same calling device. The names of lock files have the format:

LCK..str

where *str* is either a device or computer name. These files may remain in the spool directory if the communications link is unexpectedly dropped (usually on computer crashes). The lock files are ignored (removed) after the parent process is no longer active. The lock file contains the process ID of the process that created the lock. The lock file is always named using the lowercase (non-modem control) suffix to avoid possible conflicts if the same line is specified both modem-control and non-modem-control. For example, the lock on */dev/tty1A* is named *LCK..tty1a*.

C. (work file)

Work files are created in a spool directory when work (file transfers or remote command executions) is queued for a remote computer. The names of work files have the format:

C.systemnxxxx

where *system* is first five characters in the name of the remote computer, *n* is the ASCII character representing the grade (priority) of the work, and *xxxx* is the four-digit job sequence number assigned by UUCP. Work files contain the following information:

- full pathname of the file to be sent or requested
- Full pathname of the destination or `~user/filename`; “~” is shorthand for `/usr/spool/uucppublic` and must be included if the full pathname is not used.
- user login name
- list of options
- name of associated data file in the spool directory. If the **uucp -c** or **uuto -p** option was specified, a dummy name (*D.0*) is used
- mode bits of the source file
- remote user’s login name to be notified upon completion of the transfer

D. (data file)

Data files are created when it is specified in the command line to copy the source file to the spool directory. The names of data files have the following format:

D.systemxxxxyyy

where *system* is the first five characters in the name of the remote computer, *xxxx* is a four-digit job sequence number assigned by **uucp**. The four-digit job sequence number may be followed by a subsequence number, *yyy*, that is used when there are several *D.* files created for a work (*C.*) file.

X. (execute file)

Execute files are created in the spool directory prior to remote command executions. The names of execute files have the following format:

X.systemnxxx

where *system* is the first five characters in the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxx* is a four-digit sequence number assigned by UUCP. Execute files contain the following information:

- Requester's login and computer name
- Name of file(s) required for execution
- Input to be used as the standard input to the command string
- Computer and file name to receive standard output from the command execution
- Command string
- Option lines for return status requests

Complete UUCP Examples

This section includes two complete working examples of a UUCP system and the database files.

Example 1: System *gomer*

The following system (*gomer*) has:

- 1200 baud modem on tty2B
- direct connection to system (*poker*) on tty2d for call-out only
- three valid uucp logins:

nuucp	The public login for email. No password required.
ubarn	The on-site login for system (<i>poker</i>).
upay4	The private login for email and file transfers.

All lines beginning with # are comments and are not required. Most examples are partial listings and may contain other entries. Micnet is not installed. The modem answers at 1200 baud first and is set up for both call in and out.

/etc/passwd

```

uucp:NOLOGIN:4:4:Uucp admin:/usr/lib/uucp:
nuucp: :201:4:public:/usr/spool/uucpllogins/nuucp:/usr/lib/uucp/uucico
upay4:VaJnpqO:202:4:private:/usr/spool/uucppublic:/usr/lib/uucp/uucico
ubarn:bHnxWt:203:4:poker:/usr/spool/uucppublic:/usr/lib/uucp/uucico
```

/etc/group

```

uucp:x:4:uucp,nuucp,ubarn,upay4
```

Complete UUCP Examples

/etc/systemid

```
gomer
```

/etc/ttys

```
13tty2B
03tty2b
03tty2D
1mtty2d
```

/usr/lib/uucp/Devices

```
# 300-1200 baud hayes 1200 baud modem.
# The Direct tty2b entry is for programming the modem
ACU    tty2B - 300-1200 dialHA12
Direct tty2b - 300-1200 direct
poker  tty2d - 9600    direct
```

/usr/lib/uucp/Permissions

```
# Public uucp login for mail only.
# Can send mail, transfer files to/from uucppublic, and get
# a directory (ls) listing.
LOGNAME=uucp    MACHINE=OTHER \
  COMMANDS=rmail:ls:uucp \
  READ=/usr/spool/uucppublic:/usr/tmp \
  WRITE=/usr/spool/uucppublic:/usr/tmp \
  SENDFILES=yes  REQUEST=yes
# Private uucp login for mail and file transfer.
# Only dingbat, ogre, grinch, gomer, and blitzen can use this login.
LOGNAME=upay4   VALIDATE=dingbat:ogre:grinch:gomer:blitzen \
  COMMANDS=rmail:ls:uucp:who:uux \
  READ=/ WRITE=/ \
  NOREAD=/etc \
  SENDFILES=yes  REQUEST=yes
# Local trusted connection to gomer
# Only gomer can use this login.
LOGNAME=ubarn   VALIDATE=gomer \
  COMMANDS=ALL \
  READ=/ WRITE=/ \
  SENDFILES=yes  REQUEST=yes
```

/usr/lib/uucp/Systems

```
# local calls
dingbat Any ACU 1200 4444444 ogin:--BREAK-ogin:--BREAK-ogin: \
  uubig word: wetrot
# long distance (evening calls only)
grinch Any1800-0700 ACU 1200 18888888 "" \r ogin:--BREAK-ogin: \
  -BREAK-ogin:nuucp
uunet Any1800-0700 ACU 1200 17031111111 ogin:--BREAK-ogin: \
  -BREAK-ogin:xytpq sword: grm5q
# systems that call in as nuucp (for mail) but NOT call out.
daboss Never
sales Never
guru2 Never
```

Example 2: System dingbat

The following system (*dingbat*) has:

- 2400 baud modem on tty1A
- two valid uucp logins:

nuucp	The public login for email. No password required.
uubig	The private login for email and file transfers.

All lines beginning with # are comments and are not required. Most examples are partial listings and may contain other entries. Micnet is not installed. The modem answers at 2400 baud first and is set-up for both call in and out.

/etc/passwd

```
uucp:NOLOGIN:4:4:Uucp admin:/usr/lib/uucp:
nuucp:vGoPnyuX:201:4:public:/usr/spool/uucpllogins/nuucp:/usr/lib/uucp/uucico
uubig:VXcooTwi:202:4:private:/usr/spool/uucppublic:/usr/lib/uucp/uucico
```

Complete UUCP Examples

/etc/group

```
uucp:x:4:uucp,nuucp,uubig
```

/etc/systemid

```
dingbat
```

/etc/ttys

```
13tty1A  
0mtty1a
```

/usr/lib/uucp/Devices

```
# 300-2400 baud hayes 2400 baud modem.  
# The Direct entry is for programming the modem  
ACU tty1A - 300-2400 dialHA24  
Direct tty1a - 300-2400 direct
```

/usr/lib/uucp/Permissions

```

# Public uucp login for mail only.
# Can send mail, transfer files to/from uucppublic, and get
# a directory (ls) listing.
LOGNAME=nuucp      MACHINE=OTHER \
  COMMANDS=rmail:ls:uucp \
  READ=/usr/spool/uucppublic:/usr/tmp \
  WRITE=/usr/spool/uucppublic:/usr/tmp \
  SENDFILES=yes   REQUEST=yes
# Private uucp login for mail and file transfer.
# Only ogre, grinch, ... can use this login.
LOGNAME=uubig     VALIDATE=ogre:grinch:gomer:blitzen \
  COMMANDS=rmail:ls:uucp:who:uux \
  READ=/  WRITE=/ \
  NOREAD=/etc \
  SENDFILES=yes   REQUEST=yes

```

/usr/lib/uucp/Systems

```

# local calls
gomer Any ACU 2400 3333333 ogin:-BREAK-ogin:-BREAK-ogin: \
  upay4 word: dryrot
# long distance (evening calls only)
grinch Any1800-0700 ACU 2400 18888888 "" \r ogin: \
  -BREAK-ogin:-BREAK-ogin: nuucp
# systems that call in as nuucp (for mail) but NOT call out.
daboss Never
damgr Never
guru2 Never

```

Sample Commands

Send mail to another system and have it send the mail back.

```

mail othersystem!mysystem!mylogin (Bourne/Korn shell)
mail othersystem!\!mysystem!\!mylogin (C-shell)

```

Print your full mail address on this system.

```

echo "'uuname -l\!'logname'"

```

Complete UUCP Examples

Display the systems you can call.

```
uuname
```

Forcing a call to another system and save the debug output in background.

```
/usr/lib/uutry -x7 sitename
```

UUCP Error Messages

This section lists the error messages associated with UUCP. There are two types of error messages. ASSERT errors are recorded in the */usr/spool/uucpl.Admin/errors* file. STATUS errors are recorded in individual machine files found in the */usr/spool/uucpl.Status* directory.

ASSERT Error Messages

When a process is aborted, ASSERT error messages are recorded in */usr/spool/uucpl.Admin/errors*. These messages include the filename, SCCS ID, line number, and the text listed in these messages. In most cases, these errors are the result of filesystem problems. Use “errno” (when present) to investigate the problem. If “errno” is present in a message, it is shown as () in Table 11.5.

Table 11.5
ASSERT Error Messages

Error Message	Description or Action
CAN'T OPEN	An open() or fopen() failed. Check for the presence of the file and incorrect permissions.
CAN'T WRITE	A write() , fwrite() , fprint() , or so on failed. Check for the presence of the file and incorrect permissions.
CAN'T READ	A read() , fgets() , or so on failed. Check for the presence of the file and incorrect permissions.
CAN'T CREATE	A create() call failed. Check permissions.
CAN'T ALLOCATE	A dynamic allocation failed.
CAN'T LOCK	An attempt to make a LCK (lock) file failed. In some cases, this is a fatal error.
CAN'T STAT	A stat() call failed. Check for the presence of the file and incorrect permissions.

(Continued on next page.)

Table 11.5
ASSERT Error Messages (*Continued*)

Error Message	Description or Action
CAN'T CHMOD	A chmod() call failed. Check for the presence of the file and incorrect permissions.
CAN'T LINK	A link() call failed. Check for the presence of the file and incorrect permissions.
CAN'T CHDIR	A chdir() call failed. Check for the presence of the directory and incorrect permissions.
CAN'T UNLINK	An unlink() call failed.
WRONG ROLE	This is an internal logic problem.
CAN'T MOVE TO CORRUPTDIR	An attempt to move some bad <i>C.</i> or <i>X.</i> files to the <i>/usr/spool/uucp/.Corrupt</i> directory failed. The directory is probably missing or has wrong modes or owner.
CAN'T CLOSE	A close() or fclose() call failed.
FILE EXISTS	The creation of a <i>C.</i> or <i>D.</i> file is attempted, but the file exists. This occurs when there is a problem with the sequence file access. Usually indicates a software error.
No uucp server	A TCP/IP call is attempted, but there is no server for UUCP.
BAD UID	The <i>uid</i> cannot be found in the <i>/etc/passwd</i> file. The filesystem is in trouble, or the <i>/etc/passwd</i> file is inconsistent.
BAD LOGIN_UID	Same as previous message.
ULIMIT TOO SMALL	The ulimit for the current user process is too small. File transfers may fail, so transfer is not attempted.

(Continued on next page.)

Table 11.5
ASSERT Error Messages (*Continued*)

Error Message	Description or Action
BAD LINE	There is a bad line in the <i>Devices</i> file; there are not enough arguments on one or more lines.
FSTAT FAILED IN EWRDATA	There is something wrong with the Ethernet media.
SYSLST OVERFLOW	An internal table in <i>gename.c</i> overflowed. A big or strange request was attempted.
TOO MANY SAVED C FILES	Same as previous message.
RETURN FROM fixline ioctl	An ioctl, which should never fail, failed. There is a system driver problem.
BAD SPEED	A bad line speed appears in the <i>Devices</i> or <i>Systems</i> file (“Class” field).
PERMISSIONS file: BAD OPTION	There is a bad line or option in the <i>Permissions</i> file.
PKCGET READ	The remote machine probably hung up. No action need be taken.
PKXSTART	The remote machine aborted in a non-recoverable way. This can generally be ignored.
SYSTAT OPEN FAIL	There is a problem with the modes of <i>/usr/lib/uucp/.Status</i> , or there is a file with bad modes in the directory.
TOO MANY LOCKS	There is an internal problem!

(Continued on next page.)

Table 11.5
ASSERT Error Messages (*Continued*)

Error Message	Description or Action
XMV ERROR	There is a problem with some file or directory. It is likely the spool directory, because the modes of the destinations were supposed to be checked before this process was attempted.
CAN'T FORK	An attempt to fork and exec failed. The current job should not be lost, but attempted later (uuxqt). No action need be taken.

UUCP STATUS Error Messages

Status error messages are messages that are stored in the */usr/spool/uucp/.Status* directory. This directory contains a separate file for each remote machine that your system attempts to communicate with. These individual machine files contain status information on the attempted communication, whether it was successful or not. Table 11.6 lists the most common error messages that can appear in these files.

Table 11.6
STATUS Error Messages

Error Message	Description or Action
OK	Things are OK.
NO DEVICES AVAILABLE	There is currently no device available for the call. Check to see that there is a valid device in the <i>Devices</i> file for the particular system. Check the <i>Systems</i> file for the device to be used to call the system.
WRONG TIME TO CALL	A call was placed to the system at a time other than what is specified in the <i>Systems</i> file.
TALKING	Self-explanatory.

(Continued on next page.)

Table 11.6
STATUS Error Messages (*Continued*)

Error Message	Description or Action
LOGIN FAILED	The login for the given machine failed. It could be a wrong login or password, wrong number, a very slow machine, or failure in getting through the <i>dialer-token</i> script.
CONVERSATION FAILED	The conversation failed after successful startup. This usually means that one side went down, the program aborted, or the line (link) was dropped.
DIAL FAILED	The remote machine never answered. It could be a bad dialer or the wrong phone number.
BAD LOGIN/MACHINE COMBINATION	The machine called us with a login or machine name that does not agree with the <i>Permissions</i> file. This could be an attempt to masquerade!
DEVICE LOCKED	The calling device to be used is currently locked and in use by another process.
ASSERT ERROR	An ASSERT error occurred. Check the <i>/usr/spool/uucp/Admin/errors</i> file for the error message and refer to the section "ASSERT Error Messages."
SYSTEM NOT IN <i>Systems</i>	The system is not in the <i>Systems</i> file.
CAN'T ACCESS DEVICE	The device tried does not exist or the modes are wrong. Check the appropriate entries in the <i>Systems</i> and <i>Devices</i> files.
DEVICE FAILED	The open of the device failed.
WRONG MACHINE NAME	The called machine is reporting a different name than expected.
CALLBACK REQUIRED	The called machine requires that it calls your system.

(Continued on next page.)

Table 11.6
STATUS Error Messages (*Continued*)

Error Message	Description or Action
REMOTE HAS A LCK FILE FOR ME	The remote site has a LCK file for your system. They could be trying to call your machine. If they have an older version of UUCP, the process that was talking to your machine may have failed leaving the LCK file. If they have the new version of UUCP and they are not communicating with your system, then the process that has a LCK file is hung.
REMOTE DOES NOT KNOW ME	The remote machine does not have the node name of your system in its <i>Systems</i> file.
REMOTE REJECT AFTER LOGIN	The login used by your system to log in does not agree with what the remote machine was expecting.
REMOTE REJECT, UNKNOWN MESSAGE	The remote machine rejected the communication with your system for an unknown reason. The remote machine may not be running a standard version of UUCP.
STARTUP FAILED	Login succeeded, but initial handshake failed. Check communication parameters: data word size, parity, stop bits, and so on.
CALLER SCRIPT FAILED	This is usually the same as DIAL FAILED. However, if it occurs often, suspect the caller script in the <i>Dialers</i> file. Use uutry to check.

Chapter 12

Building a Local Network with Micnet

Introduction 12-1

Planning a Network 12-2

Choosing Machine Names 12-2

Choosing a Network Topology 12-2

Drawing a Network Topology Map 12-3

Network Connection Strategy 12-4

Assigning Lines and Speeds 12-5

Choosing Aliases 12-7

Building a Network 12-9

Creating the Micnet Topology and Alias Files 12-9

Saving the Micnet Files 12-14

Restoring Micnet Files 12-15

Starting the Network 12-16

Testing a Micnet Network 12-17

Checking the Network Connections 12-17

Using the LOG File to Locate a Problem 12-18

Stopping the Network 12-20

Modifying the Micnet Network 12-20

Using a UUCP System 12-21

Introduction

A Micnet network allows communications between two or more independent XENIX systems. The network consists of computers connected by serial communications lines (that is, RS-232 ports connected by cable). Each computer in the network runs as an independent system, but allows users to communicate with the other computers in the network through the **mail**, **rnp**, and **remote** commands. These commands pass information such as mail, files, and even other commands, from one computer to another.

12

It is the system manager's task to build and maintain a Micnet network. The system manager decides how the computers are to be connected, makes the actual physical connections, then uses the **netutil** program to define and start the network.

This chapter explains how to plan a network and then build it with the **netutil** program. In particular, it describes how to do the following:

- Choose machine names.
- Draw the network topology map.
- Assign serial lines.
- Create the Micnet files.
- Distribute the Micnet files.
- Test the Micnet network.

Note

Use a null-modem cable to connect Micnet machines. Pins 2, 3, and 7 must be connected with pins 2 and 3 crossed and 7 connected straight-through.

Planning a Network

To build a Micnet network, the **netutil** program requires that you provide the names of the computers that are to be in the network, a description of how the computers are to be connected, a list of the serial lines to be used, the names of the users who use the network, and what aliases (if any) they are known by.

To keep the task as simple as possible, you should take some time to plan the network and make lists of the information you are required to supply. To help you make these lists, the following sections suggest ways to plan a network.

Choosing Machine Names

A Micnet network requires that each computer in the network have a unique “machine name.” A machine name helps distinguish each computer from other computers in the network. It is best to choose machine names as the first step in planning the network. This prevents confusion later on, when you build the network with the **netutil** program.

A machine name should suggest the location of the computer or the people who use it. You can also use any name you wish. The name must be unique and consist of letters and digits. The Micnet programs use only the first eight characters of each name, so be sure those characters are unique.

The **netutil** program saves the machine name of a computer in a */etc/systemid* file. One file is created for each computer. After you have built and installed the network, you can find out the machine name of the computer you are using by displaying the contents of this file.

Choosing a Network Topology

The network topology is a description of how the computers in the network are connected. In any Micnet network, there are two general topologies from which all topologies can be constructed. These are “star” and “linear.”

In a star topology, all computers are directly connected to a central computer. All communications pass through the central computer to the desired destination.

In a linear topology, the computers form a chain, with each computer directly connected to no more than two others. All communications pass down the chain to the desired destination.

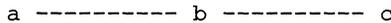
A network can be strictly star, strictly linear, or a combination of star and linear topologies. The only restriction is that no network can form a ring. For example, you cannot close up a linear network by connecting the two computers at each end.

The kind of topology you choose depends on the number of computers you have to connect, how quickly you want communications to proceed, and how you want to distribute the task of passing along communications. A star topology provides fast communication between computers, but requires both a large portion of the central computer's total operation time and a large number of serial lines on the central computer. A linear topology distributes the communications burden evenly, requiring only two serial lines per computer, but is slow if the chain is very long (communication between computers can take several minutes). Often a combination of star and linear topologies makes the best network. In any case, make the choice you think is best. If you discover you have made a wrong choice, you can change the network at any time.

Drawing a Network Topology Map

A network topology map is a sketch of the connections between computers in the network. You use the map to plan the number and location of the serial lines used to make the network.

You can make the map while you work out the topology. Simply arrange the machine names of each computer in the network on paper, then mark each pair of computers you wish to connect with serial lines. For example, the topology map for a linear network of three computers might look like this:



As you draw, make sure that there is no more than one connection between any two computers in the network. Furthermore, make sure that no rings are formed (a ring is a series of connections that form a closed circle). Multiple connections and rings are not permitted.

Network Connection Strategy

Once you have made the topology map, you can decide which serial ports to use to connect the computers. Because every connection between computers in the network requires exactly two ports (one on each computer) and one serial wire, you need to be very careful about assigning the lines.

Make a list of the serial ports (also called TTY ports) available for use on each computer in the network. You can display a list of the serial ports on a computer by examining the file */etc/ttys*. A port is available if it is not connected to any device such as a terminal or modem.

For example, in our above topology, computer *b* has two network connections, one to computer *a* and one to computer *c*. You would need to allocate two tty ports on computer *b*, one for computer *a* and one for computer *c*.

There is a restriction on tty names that can be used with Micnet. No tty port number can be duplicated across the network. Most computers have similar tty numbering schemes (tty1a, tty2a, tty3a, and so forth.) This means that on every computer in your network, the serial ports have the same names. For example, if you connect tty1b on computer *a* to tty5d on computer *b*, you cannot then assign tty1b on computer *c* to any other machine on your network. The reason for this restriction is that Micnet programs do not distinguish between different computers on the network and you must specify the ports used on each computer for communication. Therefore, if the program finds two ttys of the same name in the topology file, the network cannot operate correctly. We recommend the following strategy to allow you to manipulate Micnet connections easily.

We suggest that you choose a range of numbers that your system does not use and make those nonexistent tty ports into links for Micnet. We recommend using tty numbers beginning with 40 or 50. There is no upper limit and you can have as many of these "virtual" ttys as you need in the topology file. For example, because no two computers can use the same tty name in the topology, state in the topology file that computer *a* uses *tty50* to connect to *tty51* on computer *b*. Because these ttys do not exist on any of your systems, you can be certain that the names are not duplicated across the network.

To make these new virtual tty names correspond to actual ttys on your computer, you use the **ln(C)** command to link the filenames you have given in your topology to real ttys on your system. When you use **ln(C)** to link the two filenames, you are telling the operating system that the actual tty is now also known by the new name you give it. When you list the contents of the */dev* directory, you see both names, and both names access the same physical device.

For an example use of `ln(C)`, if you plug the wire into `tty1a` on computer *a* and the other end of the wire goes into `tty5d` on computer *b*, you give the following command on computer *a*:

```
ln /dev/tty1a /dev/tty50
```

Then, when Micnet sends information to `/dev/tty50`, the information actually goes out through `tty1a`. Similarly, on computer *b* link `/dev/tty5d`, where you have actually connected the wire, to the virtual tty called `tty51` as follows:

```
ln /dev/tty5d /dev/tty51
```

This allows you to connect the wires physically to whatever ports are available without concern for duplicating connections. For example, if the only port available on computer *c* was `/dev/tty1a`, you would not have to change your system configuration to use that name in the network topology.

This strategy is also useful when a port fails for some reason. Instead of having to make and distribute a new network topology, you merely change the link from your virtual tty (`tty50`, `51`, `52`, and so on.) to some other physical tty (`tty2c`, for example) and your network operates correctly.

Assigning Lines and Speeds

Follow these steps to create and assign your ttys and ports:

1. Using the topology map and the strategy outlined above, assign one (and only one) available tty to each network connection for each computer. List both the actual and the virtual ttys you plan to use. For example, if computer *a* has only one available serial line (`tty1a`) and you plan to use the virtual tty name `tty50`, then the entry in the topology map should look like this:

```

      a ----- b ----- c
      tty1b
      (tty50)

```

Planning a Network

2. Repeat step 1 for all computers in the topology map. Make sure that each connection is assigned a line and a virtual tty and that no two connections on any given computer have the same virtual tty number. When finished, the map should look like this:

a	-----	b	-----	c
tty1b		tty2a	tty3a	tty1b
(tty50)		(tty51)	(tty52)	(tty53)

Note

For an example of a star topology, imagine a cartwheel. One computer is the hub of the wheel and the spokes of the wheel are connections to the other network computers. The center machine in a star topology is often called the “hub” machine. The hub machine must have enough available serial ports to allow a connection to every machine on the network.

If a computer does not have enough available serial ports to meet its needs, you can make the lines available by removing the devices already connected to them. If you cannot remove devices, you must redraw your topology map.

3. Using the topology map, assign a serial-line transmission speed for each computer pair. The speed must be within the normal range for serial lines (typically 110 to 9600). Transmission speeds are a matter of preference. In general, a higher speed means a smaller amount of time to complete a transmission, but a greater demand on system’s input and output capabilities. In some cases, transmission speeds are a matter of hardware capabilities. Some hardware is not capable of transmission speeds greater than 1200 baud. For this reason, 1200 is the recommended speed when first installing Micnet. You can then increase the speed if you find the hardware can support it.

- After the topology map is completely filled in, make a list of all computer pairs, showing their machine names, serial lines, and transmission speeds. You use this list when installing the network. Here is the topology map showing the default transmission speeds:

```

computer          computer          computer
  a          ---1200--- b ---1200---          c
  tty1b          tty2a  tty3a          tty1b
  (tty50)          (tty51)  (tty52)          (tty53)
    
```

Here is a sample list of computer pairs from the above topology:

```

a (tty50) to b (tty51) at 1200 baud
b (tty52) to c (tty53) at 1200 baud
    
```

- Go to each computer on the network and give the commands to link each virtual tty to its physical counterpart on that machine. For example, the commands should have the form:

In /dev/actual-tty /dev/virtual-tty

Choosing Aliases

Once you have decided how to connect the computers in the network, you can choose aliases for users in the network. An alias is a simple name that represents both a location (computer) and a user. Aliases are used by the **mail** command to allow you to refer to specific computers and users in a network without giving the explicit machine and user names. Although not a required part of the network, aliases can make the network easier to use and maintain.

There are three kinds of aliases: standard, machine, and forward. A standard alias is a name for a single user or a group of users. A machine alias is a name for a computer or an entire network (called a site). A forward alias is a temporary alias for a single user or group of users. A forward alias allows users who normally receive network communications at one computer to receive them at another.

When you build a network with the **netutil** program, you are asked to provide standard aliases only. (You can incorporate machine and forward aliases into the network at your leisure.) Each standard alias must have a unique name and a list of the login names of the users it represents. You may choose any name you wish as long as it consists of letters and numbers, begins with a letter, and does not have the same spelling as the login names. The name should suggest the user or group of users it represents. The login names must be the valid login names of users in the network.

Planning a Network

To help you prepare the aliases for entry during the **netutil** program, follow these steps:

1. Make a list of the user aliases (that is, the aliases that refer to just one user) and the corresponding login names of each user.
2. Make a separate list of the group aliases (that is, the aliases that refer to two or more users) and the login names or user aliases (from the first list) of the corresponding users. A group alias may have any number of corresponding users.

Note that there are a number of predefined group aliases. The name **all** is the predefined alias for all users in the network. The machine names of the computers in the network are predefined aliases for the users on each computer. Do not use these names when defining your own aliases.

Building a Network

You build a network with the **netutil** program. The program allows you to define the machines, users, and serial lines that make up the network.

To build a network, you must first create the Micnet files that define the network and then transfer these files to each computer in the network. After each computer receives the files, you can start the network and use it to communicate between computers.

12

The following sections describe how to build the network.

Creating the Micnet Topology and Alias Files

The Micnet files are created with the Install option of the **netutil** program. The Install option asks for the names, aliases, and serial lines of each computer in the network. As you supply the information, it automatically creates the files needed for each computer. These files can then be transferred to the other computers in the network with the Save and Restore options of **netutil**. This means you can build the entire network from just one computer.

To use the **install** option, follow these steps:

1. Log in as the super user.
2. Enter:

```
netutil
```

Now press the <Return> key. The program displays the network utility menu. The Install option is the first item in the menu.

3. Enter **1**, and press <Return>. If the program display already exist, you see the following message:

```

  _____
  | Compiling new network topology
  | Overwrite existing network files? (yes/no)?
  |_____

```

Enter **y** and press the <Return> key to overwrite the files. The existing network files must be overwritten to create the new network. The first time you install the network, these files contain default

Building a Network

information that need not be saved. If you install the system a second time or expand the system, it may be wise to save a copy of these files before starting the Install option. The files can be saved on a floppy or a hard disk with the Save option described later in this chapter.

Once you have entered *y*, the program displays the following message:

```
Enter the name of each machine
      (or press RETURN to continue installation).
Machine name:
```

4. Enter the machine name and press the `<Return>` key. You can enter more than one name on a line by separating each with a comma or a space. After you have entered all the names, press the `<Return>` key to continue to the next step. The program displays the names you entered and asks if you wish to make changes.
5. Enter *y* (for “yes”) if you wish to enter all the names again. Otherwise, enter *n* (for “no”) or just press the `<Return>` key to move on to the next step. If you enter *n*, the program displays the message:

```
For each machine, enter the names of the machines
      to be connected with it
Machine name:
      Connect to:
```

6. Using the list of machine pairs you created when planning the network, enter the machine names of the computers connected to the given computer. You can enter more than one name on a line by separating each name with a comma (,) or a space. When you have entered the machine names of all computers connected to the given computer, press the `<Return>` key. The program prompts for the names of the computers connected to the next computer.
7. Repeat step 6 for all remaining computers. As the program prompts for each new set of connections, it shows a list of the machine names it already knows to be connected with the current computer. You need not enter these names. The program automatically checks for loops. If it finds one, it ignores the machine name that creates the loop and prompts for another.

Finally, when you have given the connections for all computers in the network, the program displays a list of the connections and asks if you wish to make corrections.

8. Enter **y** if you wish to enter the connections again. Otherwise, enter **n** to move to the next step. If you enter **n**, the program displays the message:

```
For each machine pair, enter the tty name and tty speeds
For the a <==> b machine pair.
Tty on name:
```

9. Using the list of serial-line assignments you created when planning the network, enter the serial-line name or number (for example, tty14) for the first computer in the pair and press the **<Return>** key. The program displays the message:

```
Tty on name:
```

10. Enter the serial-line name for the second computer in the pair and press the **<Return>** key. The program displays the message:

```
Speed:
```

11. Enter the speed (for example, 1200) and press the **<Return>** key. The program asks for the serial lines and transmission speed of the next pair.
12. Repeat steps 9 and 10 for all remaining machine pairs. When you have given serial lines and speeds for all pairs, the program displays this information and asks if you wish to make corrections.
13. Enter **y** if you wish to enter the serial lines and speeds again. Otherwise, enter **n**, to exit the program. move to the next step.

The program displays the message:

```
Enter the names of users on each machine:
```

```
For machine name:
```

```
Users on name:
```

14. Enter the login name of a user on the given computer, then press the `<Return>` key. You can enter more than one name on a line by separating each name with a comma (,) or a space. When you have entered all names for the given computer, press the `<Return>` key. The program displays the names of the users on the computer and asks if you wish to make corrections.
15. Enter *y*, if you wish to enter the user names again. Otherwise, enter *n*. If you enter *n*, the program prompts you for the names of the users on the next computer.
16. Repeat steps 14 and 15 for all remaining computers. After you have entered the names of users for every computer, the program prompts you to enter any aliases:

```
Do you wish to enter any aliases? (yes/no)?
```

17. Enter *y*, if you wish to enter aliases. Otherwise, enter *n*, to complete the installation. If you enter *y*, the program displays the message:

```
Each alias consists of two parts, the first is the alias name,  
the second is a list of one or more of the following:
```

```
    valid user names  
    previously defined aliases  
    machine names
```

```
Aliases:
```

18. Using the list of aliases you created when planning the network, enter the name of an alias and press the `<Return>` key. The program displays the message:

```
Users/Aliases:
```

19. If the alias is to name a single user, enter the login name of that user and press the `<Return>` key. The program then prompts for another alias.

If, on the other hand, the alias is to name several users, enter the login names of the users. If one or more of the users to be named by the alias are already named by other aliases, enter the aliases instead of the login names. If all the users on one computer are to be named by the alias, enter the machine name instead of the login names. In any case, make sure that each item entered on the line is separated from the next by a comma (,) or a space. If there are more items than can fit on the line, enter a comma after the last item on that line and press the `<Return>` key. You can then continue on the next line. After all names and aliases have been entered, press the `<Return>` key. The program then prompts you for another alias.

20. Repeat steps 18 and 19 for all remaining user aliases in your list. When you have given all aliases, press the `<Return>` key. The program displays a list of all aliases and their users and asks if you wish to make corrections.
21. Enter `y`, if you wish to enter all aliases again. Otherwise, enter `n` to complete the installation.
22. Once you direct **netutil** to complete the installation, it copies the information you have supplied to the network files, displaying the name of each file as it is updated.

Once the files are updated, you can use the Save option to copy the Micnet files to floppy disk.

Saving the Micnet Files

You can save copies of the Micnet files on backup media (floppy disk) or hard disk with the Save option of the **netutil** program. Saving the files allows you to transfer them to the other computers in the network. Before you can save the files to a floppy you need to format a floppy disk (see the section “Formatting Floppy Disks” in “Using Floppy Disks and Tape Drives” chapter of this guide). Saving the files to the hard disk enables you to use **uucp(C)** to transfer the files to other machines.

To save the files, follow these steps:

1. Log in as the super user.
2. Enter:

netutil

Press the `<Return>` key. The program displays the network utility menu.

3. Enter **2**, and press the `<Return>` key. The program displays the message:

```
Save to /dev/fdx (yes/no)?
```

where *x* is a drive number.

4. If you wish to use the specified disk drive, insert a blank, formatted floppy disk into the drive, close the floppy door, then enter “yes”, and press the `<Return>` key. If you do not wish to use the drive, enter “no,” and press the `<Return>` key. The program displays a prompt asking you for the filename of the disk drive (or file) you wish to use. Insert a blank, formatted disk into your chosen drive, wait for the drive to accept the disk, then enter the filename of the drive. The name of the default backup device (disk drive) is specified in the file */etc/default/micnet*. This device can be changed depending on system configuration.

In either case, the program copies the Micnet files to the floppy disk.

5. Remove the floppy disk from the drive. Using a soft tip marker (do not use a ball-point pen), label the disk “Micnet disk.”

As soon as all files have been copied, you can transfer them to all computers in the network.

Restoring Micnet Files

The last step in building a Micnet network is to copy the Micnet files from the Micnet disk to all computers in the network. Do this with the Restore option of the **netutil** program. For each computer in the network, follow these steps:

1. Log in as the super user.
2. Enter:

netutil

Press the `<Return>` key. The program displays the network utility menu.

3. Enter **3**, and press the `<Return>` key. The program displays the message:

```
Restore from /dev/fdx (yes/no)?
```

where x is the number of a drive.

4. If you wish to use the specified disk drive, insert the Micnet disk into the drive, wait for the drive to accept the disk, then enter “yes” and press the `<Return>` key. If you do not wish to use the drive, enter “no” and press the `<Return>` key. The program displays a prompt asking you for the filename of the disk drive you wish to use. Insert the Micnet disk into your chosen drive, wait for the drive to accept the disk, then enter the filename of the drive.

In either case, the program copies the network files to the appropriate directories, displaying the name of each file as it is copied. Finally, the program displays the message:

```
Enter the name of this machine:
```

5. Enter the machine name of the computer you are using and press the `<Return>` key. The program copies this name to the new `/etc/systemid` file for the computer. If necessary, it also disables the serial lines to be used on the computer, preparing them for use with the network.

When the files are copied, you can start the network with the Start option.

Starting the Network

Once the Micnet files are transferred to a computer, you can start the network with the Start option of the **netutil** program. The Start option starts the Micnet programs, which perform the tasks needed to communicate between the computers in the network.

To start the network, follow these steps for each computer in the network:

1. Log in as the super user.
2. Enter:

netutil

Press the `<Return>` key. The system displays the network utility menu.

3. Enter **4**, and press the `<Return>` key. The program searches for the */etc/systemid* file. If it finds the file it starts the network. If it does not, it prompts you to enter the machine name of the computer and then creates the file. The program also asks if you wish to log errors and transmissions. In general, these are not required except when checking or testing the network. When starting the network for the first time, enter **n** in response to each question and press the `<Return>` key.

Once the network has started, you can move to the next computer and start the network there.

Note that, for convenience, you can let each computer start the network automatically whenever the system itself is started. Simply include the following command in the system initialization file, */etc/rc.d/7/udaemons*, of each computer:

netutil start

To add this command, use a text editor as described in the section "Changing the */etc/rc* Scripts" in the "Customizing System Startup" appendix in this guide. You can add the **-x** or **-e** options to this command line if you wish to log transmissions or errors. Even if you do not use these options, Micnet copies a login and logout message to the system *LOG* file each time you start and stop the network. This means you need to periodically clear the file. See the section "Clearing Log Files" in the "Managing Filesystems" chapter in this guide.

Testing a Micnet Network

After you have started a network for the first time, you should test the network to see that it is properly installed. In particular, you must determine whether or not each computer is connected to the network.

To test the network, you need to know how to use the `rcp(C)` command (see the “Mail” chapter in the *XENIX User's Guide*). The following sections explain how to test the network and how to correct the network if problems are discovered.

Checking the Network Connections

You can make sure that all computers are connected to the network by mailing a short message to **all** (the alias for all users in the network) with the `mail` command and copying a short file to all the machines in the network with the `rcp` command. Follow these steps:

1. Choose a computer.
2. Log in as the super user.
3. Use the `rcp` command (see the “Communicating with Other Sites” chapter of the *XENIX User's Guide*) and the `all` alias to mail the message:

```
rcp /etc/motd machine:/tmp/test
```

where *machine* is the name of the other machine. Repeat this command for each node in your Micnet network to all users in the network.

4. Check the `/tmp` of each machine in the network to see if the file was received. To check the mailboxes, log in as the super user at each computer and enter the following command:

```
mail -u login-name
```

where *login-name* is the user's login name.

Testing a Micnet Network

The network is properly installed when all machines have received the file. If one or more computers fail to receive the file, the computers are not properly connected to the network. To fix the problem, you need to locate the computer that has failed to make a connection. The next section explains how to do this.

Using the LOG File to Locate a Problem

You can locate a problem with connections by examining the *LOG* files on each computer in the network. The *LOG* files contain records of the interaction between each pair of computers. There are two *LOG* files for each pair of computers (one file on each computer). The *LOG* files on any given computer are kept in subdirectories of the */usr/spool/micnet* directory. Each subdirectory has as its name the *machine-name* of the other computer in the pair. You can examine the contents of a *LOG* file by entering:

```
cat /usr/spool/micnet/remote/machine-name/LOG
```

then press the `<Return>` key. The *machine-name* must be the name of a computer that is paired with the computer you are using.

Each *LOG* file should contain a “startup message” that lists the name of each computer in the pair, and the serial line through which the pair is connected. It also shows the date and time at which the network was started. The message should look like:

```
daemon.mn: running as MASTER
Local system: a
Remote system: b, /dev/tty52
Tue Sep 24 22:30:35 1985
```

A startup message is added to the file each time the network starts successfully. If the message is not present, one or more of the network files and directories cannot be found. Make sure that you have used the Restore option to transfer all the network files to the computer. Also, make sure that the */etc/systemid* file contains the correct machine name for the given computer.

Each *LOG* file contains a “handshake” message if the connection between the computer pair was established. The following message is added to the file on a successful connection:

```
first handshake complete
```

If the message is not present, make sure that the network was started on the other computer in the pair. The network must be started on both computers before any connection can be made. If the network is started on both computers but the handshake message does not appear, then the serial line may be damaged or improperly connected. Check that the ttys (and their virtual ttys) are disabled. Check the serial line to make sure that the cable is firmly seated and attached to the correct RS-232 connectors on both computers. If necessary, replace the cable with one known to work.

If both the startup and handshake messages appear in the *LOG* file but the network is still not working, then there is a problem in transmission. You can create a record of the transmissions and errors encountered while transmitting by restarting the network and requesting Micnet to log all transmissions and errors. Just enter *y* (for “yes”) when the Start option asks if you wish to log errors or transmissions.

Error entries contain the error messages generated during transmission. Each message lists the cause of the error and the subroutine that detected the error. For example, the following message shows that the *rsync* subroutine received a bad response (character 68 hexadecimal) from the other computer:

```
rsync: bad Probe resp: 68
```

You can use this information to track down the cause of the problem. One common problem is stray information being passed down the serial line by electronic noise. Make sure that the serial line’s cable is properly protected against noise (for example, that the cable does not lie near any electric motor, generator, or other source of electromagnetic radiation). Also make sure the cable is in good condition.

Transmission entries contain a record of normal transmissions between computers. Each entry lists the direction, byte count, elapsed time, and time of day of the transmission. For example, the following entry shows that 29349 bytes were received (*rx*) at 16:22:

```
rx: 29349b 2:22 @16:22
```

The elapsed time for the transmission was 2 minutes and 22 seconds. You can use the records to see if messages are actually being transmitted.

Stopping the Network

You can stop the network with the Stop option of the **netutil** program. This option stops the Micnet programs, stopping communication between computers in the network.

To stop the network, follow these steps on each computer in the network:

1. Log in as the super user.
2. Enter:

netutil

Press the `<Return>` key. The program displays the network utility menu.

3. Enter **5**, and press the `<Return>` key. The program stops the network programs running on the computer.

Modifying the Micnet Network

You can modify a Micnet network at any time by changing one or more of the Micnet files. You can reinstall the network with the **netutil** program. For very small changes (for example, correcting the spelling of a machine name), you can modify the Micnet files directly with a text editor. The files and their contents are described in detail in the Miscellaneous (M) section of the manual pages.

Before making any changes to a file, a copy should be made. You can make a copy with the **cp** command. You can replace an old file with the updated file using the **mv** command. Once one or more files are changed on one computer, the files must be transferred to the other systems in the network using the Save and Restore options. These options can only be used after you have stopped the network.

Using a UUCP System

You can send and receive mail from other Micnet sites by installing a UUCP system on one computer in your site. A UUCP system is a set of programs that provide communication between computers using ordinary telephone lines.

To use a UUCP system with your Micnet network, follow these steps:

1. Install a UUCP system on one computer in the Micnet site. Installation of a uucp system requires a modem and the UUCP software provided with the XENIX *Operating System*. See the “Building a Remote Network with UUCP” chapter in this Guide for complete details.
2. Add the entry:

```
uucp:
```

to the *aliases* file of the computer on which the UUCP system is installed.

3. For all other computers in your site, add the entry:

```
uucp:machine-name:
```

to the *aliases* file. The *machine-name* must be the name of the computer on which the UUCP system is installed. One can also use the longer form of entry on the computer on which the uucp system is installed.

You can test the **uucp** system by mailing a short letter to yourself via another site. For example, if you are on the site “chicago”, and there is another Micnet site named “seattle” in the system, then the command:

```
mail seattle!chicago!johnd
```

sends mail to the “seattle” site, then back to your “chicago” site, and finally to the user “johnd” in your Micnet network. Note that a UUCP system usually performs its communication tasks according to a fixed schedule, and may not return mail immediately.

Chapter 13

XENIX Directories and Special Device Files

Introduction 13-1

XENIX Directories 13-2

- The Root Directory 13-2
- The */bin* Directory 13-2
- The */dev* Directory 13-2
- The */etc* Directory 13-3
- The */lib* Directory 13-5
- The */mnt* Directory 13-5
- The */usr* Directory 13-5
- The */tmp* Directory 13-6
- The */shlib* Directory 13-6

Log Files 13-7

Special Device Files 13-9

- Special Filenames 13-9
- Block Sizes 13-10
- Gap and Block Numbers 13-10
- Terminal and Network Requirements 13-11

Introduction

This chapter lists the most frequently used files and directories on a XENIX system. Many of these files and directories are required for proper operation and must not be removed or modified. The following sections briefly describe each directory.

This chapter also contains information needed to create device nodes relating to filesystems and terminals. For a full description of the special files mentioned here, see the manual pages in the Hardware Dependent (HW) section.

XENIX Directories

The following subsections discuss each of the main directories of the operating system.

The Root Directory

The root directory (*/*) contains the following system directories:

<i>/bin</i>	XENIX command directory
<i>/dev</i>	device special directory
<i>/etc</i>	additional program and data file directory
<i>/lib</i>	programming library directory
<i>/mnt</i>	mount directory (reserved for mounted filesystems)
<i>/usr</i>	user service routines (may contain user home directories)
<i>/tmp</i>	temporary directory (reserved for temporary files created by programs)
<i>/shlib</i>	shared libraries used by COFF programs

All of the above directories are required for system operation.

The root directory also contains a few ordinary files. Of these files, the most notable is the */xenix* file, which contains the XENIX kernel image.

The */bin* Directory

The */bin* directory contains the most common XENIX commands, that is, the commands likely to be used by anyone on the system.

The */dev* Directory

The */dev* directory contains special device files that control access to peripheral devices. All files in this directory are required, and must not be removed. There are several subdirectories to the */dev* directory. Each of these subdirectories holds special device files related to a certain type of device. For example, the */dev/dsk* directory contains device files for floppy and hard disks. The operating system supports both UNIX and

XENIX device naming conventions. Where appropriate, the files in the */dev/dsk* directories are linked to the device files that exist in */dev*. You can access the same device through the file in */dev* or the file for the same device in a subdirectory of */dev*.

Table 13.1 contains a partial list of devices.

Table 13.1
/dev Device Nodes

UNIX Device	XENIX Device	Name
<i>/dev/console</i>	same	system console
<i>/dev/rdsk/*</i>	<i>/dev/r*</i>	raw devices
<i>/dev/dsk/0s0</i>	<i>/dev/hd00</i>	entire disk on drive 0
<i>/dev/dsk/0s1</i>	<i>/dev/hd01</i>	first disk partition on drive 0
<i>/dev/dsk/0s2</i>	<i>/dev/hd02</i>	second disk partition on drive 0
<i>/dev/dsk/1s0</i>	<i>/dev/hd10</i>	entire disk on drive 1
<i>/dev/dsk/1s1</i>	<i>/dev/hd11</i>	first disk partition on drive 1
<i>/dev/dsk/1s2</i>	<i>/dev/hd12</i>	second disk partition on drive 1
<i>/dev/dsk/f0d9dt</i>	<i>/dev/fd048ds9</i>	360K floppy drive 0
<i>/dev/dsk/f0q9dt</i>	<i>/dev/fd096ds9</i>	720K floppy drive 0
<i>/dev/dsk/f0q15dt</i>	<i>/dev/fd096ds15</i>	1.2MB floppy drive 0
<i>/dev/dsk/f0q9dt</i>	<i>/dev/fd0135ds18</i>	1.44MB floppy drive 0
<i>/dev/lp</i>	same	lineprinter
<i>/dev/kmem</i>	same	kernel virtual memory
<i>/dev/mem</i>	same	physical memory
<i>/dev/null</i>	same	null device
-	<i>/dev/rct0</i>	QIC tape device
-	<i>/dev/rft0</i>	QIC-40 tape device
-	<i>/dev/rctmini</i>	minicartridge tape device
<i>/dev/root</i>	same	root file structure
<i>/dev/swap</i>	same	swap area
<i>/dev/tty</i>	same	terminals

The */etc* Directory

The */etc* directory contains miscellaneous system program and data files. All files are required, but many can be modified. The data files in the directory */etc/rc.d* contain initialization commands run by the */etc/rc* script when the system goes into multiuser mode. (The */etc/rc* directories are discussed extensively in Appendix A, "Customizing System Startup.")

The data files in the directory */etc/default* contain default information that is used by system commands (see **default(F)**). The following data files can be modified. No files should be removed. See Table 13.2.

Table 13.2
/etc/default Files

File	Utility
<i>/etc/default/archive</i>	sysadmin(ADM) backup default information
<i>/etc/default/backup</i>	backup(ADM) default information
<i>/etc/default/boot</i>	boot(ADM) information
<i>/etc/default/cron</i>	cron(C) default logging information
<i>/etc/default/dumpdir</i>	dumpdir(C) default information
<i>/etc/default/filesys</i>	sysadmsh(ADM) default filesystem data
<i>/etc/default/format</i>	format(C) default information
<i>/etc/default/idleout</i>	idleout(M) default information
<i>/etc/default/lang</i>	default locale information
<i>/etc/default/lock</i>	lock(C) default information
<i>/etc/default/login</i>	login(M) default information
<i>/etc/default/lpd</i>	lp(C) default information
<i>/etc/default/man</i>	man(C) online man page default information
<i>/etc/default/mapchan</i>	mapchan(M) default information
<i>/etc/default/micnet</i>	micnet(M) default information
<i>/etc/default/mkuser</i>	default mkuser(ADM) account information
<i>/etc/default/msdos</i>	DOS disk info (A:, B:,....) for dos(C)
<i>/etc/default/passwd</i>	passwd(C) default information
<i>/etc/default/restore</i>	restore(C) default information
<i>/etc/default/su</i>	su(C) default information (note that you must create this file yourself)
<i>/etc/default/tape</i>	tape(C) default device information
<i>/etc/default/tar</i>	tar(C) default device information
<i>/etc/default/usemouse</i>	usemouse(C) default information

The */lib* Directory

The */lib* directory contains runtime library files for C and other language programs. The directory is required for software development.

The */mnt* Directory

The */mnt* directory is an empty directory reserved for mounting removable filesystems.

The */usr* Directory

The */usr* directory consists of several subdirectories that contain additional XENIX commands and data files. It is also the default location of user home directories.

The */usr/bin* directory contains more XENIX commands. These commands are used less frequently or are considered nonessential to XENIX system operation.

The */usr/include* directory contains header files for compiling C programs.

The */usr/lib* directory contains more libraries and data files used by various XENIX commands.

The */usr/spool* directory contains various directories for storing files to be printed, mailed, or passed through networks.

The */usr/tmp* directory contains more temporary files.

The */usr/adm* directory contains data files associated with system administration and accounting. In particular, the */usr/adm/messages* file contains a record of all error messages sent to the system console. This file is especially useful for locating hardware problems. For example, an unusual number of disk errors on a drive indicates a defective or misaligned drive. Because messages in the file can accumulate rapidly, the file must be deleted periodically.

The */tmp* Directory

The */tmp* directory contains temporary files created by XENIX programs. The files are normally present when the corresponding program is running, but may also be left in the directory if the program is prematurely stopped. You can remove any temporary file that does not belong to a running program.

The */shlib* Directory

The */shlib* directory contains shared library routines used by programs written for UNIX System V/386.

Log Files

A variety of directories contain log files that grow in size during the normal course of system operation. Many of these files must be periodically cleared to prevent them from taking up valuable disk space. (See the section on “Clearing Log Files” in the “Managing Filesystems” chapter.) Table 13.3 lists the files (by full pathname) and their contents.

Table 13.3
System Log Files

13

Filename	Description
<i>/etc/ddate</i>	records date of each backup.
<i>/usr/adm/pacct</i>	records accounting information; grows rapidly when process accounting is on. (See accton(ADM) and acctcom(ADM) .)
<i>/usr/adm/messages</i>	records error messages generated by the system when started. (See messages(M) .)
<i>/etc/wtmp</i>	records user logins and logouts. (See login(M) .)

(Continued on next page.)

Table 13.3
System Log Files (Continued)

Filename	Description
<i>/usr/adm/sulog</i>	records each use of the su command; grows only if option is set in the <i>/etc/default/su</i> file. You must create <i>/etc/default/su</i> . (See su(C) .)
<i>/usr/lib/cron/cronlog</i>	records each use of the at(C) and cron(C) commands.
<i>/usr/spool/micnet/remote*/LOG</i>	records transmissions between machines in a Micnet network. The (*) must be the name of a remote machine connected to the current machine.
<i>/usr/spool/uucp/.Log/utility/sitesname/*</i>	logs UUCP commands used over a UUCP network. The <i>utility</i> and <i>sitesname</i> are the name of the UUCP utility and the name of the remote site, respectively.
<i>/usr/spool/uucp/.Log/.Old/*</i>	stores old log files placed in this directory by the uudemon.clean shell script.

Special Device Files

Many of the filesystem maintenance tasks described in this guide require the use of special filenames, block sizes, and gap and block numbers. The following sections describe each in detail.

Special Filenames

A special filename is the name of either the device special block or character I/O file, which corresponds to a peripheral device such as a hard or floppy disk drive. These names are required in such commands as **mkfs(ADM)**, **mount(ADM)**, and **df(C)** to specify the device containing the filesystem to be created, mounted, or searched.

Table 13.4 lists the special filenames and corresponding devices for hard and floppy disk drives on a typical computer.

Table 13.4
Disk Device Filenames

Filename	Disk Drive
<i>/dev/fd0</i>	floppy drive 0
<i>/dev/fd1</i>	floppy drive 1
<i>/dev/hd00</i>	entire hard disk
<i>/dev/root</i>	root filesystem
<i>/dev/u</i>	user filesystem

Block Sizes

The block size of a disk is the number of blocks of storage space available on the disk, where a block is 1024 bytes of storage. Most commands report disk space in terms of 512 byte blocks, in particular **df(C)**, **du(C)**, **ls(C)**, **lc(C)**, and **find(C)**. A 500-byte file on a 1024-byte block filesystem is reported as using 2 blocks by these utilities, as the file uses one system block that is equivalent to two 512-byte blocks. The size of a 40-megabyte hard disk in 1024-byte blocks is 39168. Note that some of the blocks on the disk are reserved for system use and cannot be accessed by user programs. The block size of a typical floppy disk depends on the total storage capacity of the disk, as given by the manufacturer.

Gap and Block Numbers

The gap and block numbers are used by the **mkfs(ADM)** and **fsck(ADM)**, commands to describe how the blocks are to be arranged on a disk. Table 13.5 lists the gap and block numbers for the floppy and hard disks used with a typical computer.

Table 13.5
Gap and Block Numbers

Disks	Gap	Block
floppy disk, 48ds9	1	9
floppy disk, 96ds15	1	15
floppy disk, 135ds9	1	9
floppy disk, 135ds18	1	18
hard disk	1	34

The number of blocks can also be determined by multiplying the number of sectors per track (for example, 17) by the number of heads on the hard disk, dividing by 2 (because there are 2 sectors per block), and rounding off to the nearest integer.

Terminal and Network Requirements

The **enable(C)** and **disable(C)** commands enable and disable logins on terminals. **enable** and **disable** require the names of the serial lines through which a terminal or network is to be connected. Table 13.6 lists the device special filenames of the two serial lines (actually two serial ports either with or without modem control). The character I/O files corresponding to these serial lines can be found in the */dev* directory. Note that the files */dev/console* and */dev/tty01* through */dev/tty12* represent “hardwired” Multiscreen devices and are not available for connection to terminals or hardware. Also, refer to **serial(HW)** for more information on serial lines.

13

Table 13.6
Serial Devices

Filename	Line
<i>/dev/tty1a</i>	main serial line (without modem control)
<i>/dev/tty2a</i>	alternate serial line (without modem control)
<i>/dev/tty1A</i>	main serial line (with modem control)
<i>/dev/tty2A</i>	alternate serial line (with modem control)

Chapter 14

Adding Terminals and Modems

Introduction 14-1

Adding and Using Terminals 14-2

 Changing Default Terminal Line Characteristics 14-6

 The gettydefs File 14-7

 Changing the gettydefs File 14-9

 Checking the Terminal Settings 14-11

 Changing Serial Line Operation 14-11

 Setting the Terminal Type 14-13

 Setting the Terminal Type Automatically 14-15

 Removing a Terminal 14-15

 Setting Up a Serial Console 14-16

 Using Serial Multiscreens with mscreen 14-17

 Troubleshooting 14-20

 Advanced Uses 14-21

Installing and Using Modems 14-24

 Choosing a Serial Port 14-24

 Configuring Your Modem 14-25

 Smartmodem 1200 14-25

 Smartmodem 2400 or V-series Smartmodem 9600 14-25

 Telebit Trailblazer 14-26

 Installing a Modem 14-27

 Dialing In to Your Computer 14-33

 Adding Passwords for Dial-in Lines 14-33

 Dialing Out from Your Computer 14-34

 Using Dialer Binaries 14-34

Introduction

An important aspect of a multiuser system is the addition and maintenance of terminal and modem devices. Adding terminals and modems allows greater access to the system and adds to overall system capabilities.

This chapter explains the following tasks:

- enabling terminal devices for operation, including serial multiscreens
- maintaining terminals
- configuring and using **msscreen(M)**, a serial terminal equivalent of **multiscreen(M)**
- connecting modems to your computer
- maintaining modem connections
- defining passwords for dial-in lines

A prerequisite to adding terminals or modems to your computer can include the addition of physical ports to your system. Consult the “Adding Multiport Cards, Memory, and Other Bus Cards” chapter of this guide to install the card, using the **mkdev serial** command to prepare the ports for use.

Adding and Using Terminals

Before you add a terminal to your system, look in the hardware manual for your terminal for instructions on connecting the terminal to a serial line. Also, refer to the list of standard serial lines in the “XENIX Directories and Special Device Files” chapter of this guide to find the name of your serial line. (If you add a serial card, the possible names of the additional device files are listed in **serial(HW)** or in the documentation for cards that include driver software.)

Many types of terminals are supported. Look in the **terminals(M)** manual page for a comprehensive list of terminals supported. Support for terminals is provided through the *etc/termcap* file, which contains the definitions and classifications of keystrokes and control sequences that vary from terminal to terminal. For a description of the *etc/termcap* file, see the **termcap(F)** manual page.

The following steps show how to install a terminal with the standard COM serial lines or with serial expansion cards:

1. If you are adding a terminal directly to a COM port, you need not run **mkdev serial**. Otherwise, you should consult “Adding Multiport Cards, Memory, and Other Bus Cards” to configure the ports for terminal connections.
2. Make sure you are logged in as *root* in multiuser mode.

Plug in your terminal and turn it on. Set it for 9600 baud, 8 data bits, 1 stop bit, no parity, full duplex, and XON/XOFF handshaking. If your terminal does not work in this mode, look for advice on configuring your terminal in the section “Changing the *gettydefs* File” later in this chapter and in the **stty(C)** manual page.

Some terminals connect with a straight cable directly to the computer. Other terminals connect to a modem. Terminals connected to a modem use a “null modem” or “modem connector,” which is a cable with pins 2 and 3 crossed. Connect the terminal so that Transmit Data on the serial port is connected to Receive Data on

the terminal, and Transmit Data on the terminal is connected to Receive Data on the serial port. Signal Ground should be connected to Signal Ground. Other pins probably do not need to be connected. The operating system requires only that pins 2, 3, and 7 are connected.

For more information on your terminal, refer to your terminal manual or a reference on serial communication.

3. If the port is enabled, press the <Return> key a few times to see if a “login:” prompt appears. If so, you are ready to log in. If not, use the console or a working terminal to log in as the superuser (*root*), and disable the port with this command:

```
disable ttyname
```

where *ttyname* is the device special name of the port in question. You should disable both the modem and non-modem devices, as in this example:

```
disable tty1A  
disable tty1a
```

Terminal connections actually use the non-modem control device, for example, */dev/tty1a*, and not */dev/tty1A*. For more information on serial port names, see **serial(HW)**.

4. Check that the entry for this serial port in the */etc/ttys* file looks like the following (*ttyname* is the name of the device file, for example */dev/tty1a*):

```
0mtty1a
```

If the entry does not look like this example, edit the file to correct it. Information on the format of the */etc/ttys* file can be found in the **ttys(F)** manual page.

5. From the console, as *root*, see if you can redirect output to the terminal by entering:

```
date > /dev/ttyname
```

Adding and Using Terminals

If you do not see the date printed on the terminal and you are not sure of the correct *ttyname*, try other *ttynames* on that serial port. If you still do not see the date printed on the terminal, then try the following:

- Make certain that the terminal is plugged in.
- Check that the cable is configured correctly. If the serial port you are using has a 25-pin connector (DB-25), read through step 2 in the preceding set of instructions. Are pins 2, 3, and 7 connected correctly? (Note that pins other than 2, 3, and 7 are probably not used.)

If your system or expansion card has a 9-pin connector (DB-9), you must use a 9-pin to 25-pin connector. Look in your hardware manual for information on 9-pin to 25-pin connections.

- Check your terminal setup configuration. See step 2 in the preceding set of instructions. Try changing the baud rate.
- Check the switches on your serial port. If you are using a multiport card, try other lines on that card.
- Attach the terminal to a standard serial port (COM1 or COM2) to see if the terminal and cable are working correctly. If you are already using a COM port, try switching to another one.

If you have successfully installed another terminal, switch hardware between the working and the nonworking terminal one piece at a time. This may help you isolate a hardware problem. Note that some faulty hardware may work under DOS but not on a XENIX system.

6. When the date prints on your terminal, enable the port with the following command:

```
enable ttyname
```

For example, if you are using COM1, use the command:

```
enable tty1a
```

The **enable** command starts a **getty** process that displays the following login prompt:

```
login:
```

If you do not see the “login:” prompt, enter the following command to verify that **getty** is running on the port and that the software is configured properly:

```
ps -t ttyname
```

Your screen should display a message similar to the one in the following example, with either “login” or “getty” listed in the “COMMAND” column:

```
PID    TTY    TIME  COMMAND
2557   1a    0:06  getty
```

14

7. If you have typed the **enable** and **disable** commands many times, it is possible that a new **getty** cannot be spawned on that port. If so, shut the system down, reboot, log in as *root* in multiuser mode, and try again.

Adding and Using Terminals

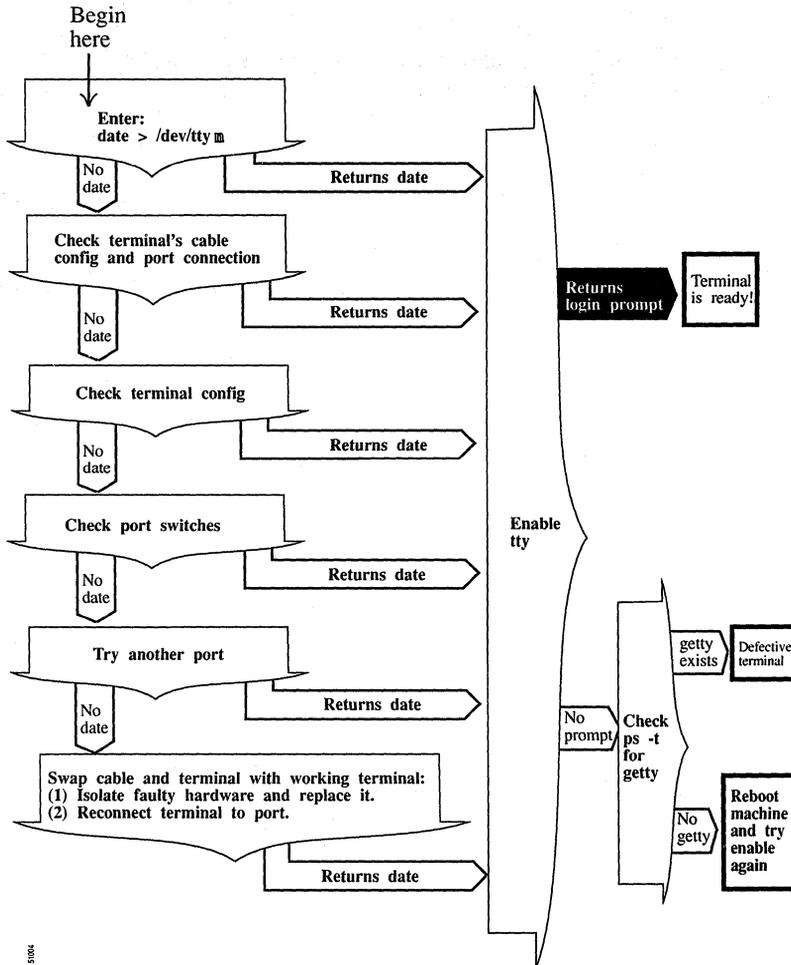


Figure 14-1 Checking the Terminal Connection

Changing Default Terminal Line Characteristics

Your system can automatically adapt to several different terminal baud rates and settings. The same program that displays the login message, **getty(M)** (for “get tty”), reads these terminal line values from a table, trying each setting until one is successful, and the user can log in to the system. This table provides several default settings for different kinds of terminal lines.

getty automatically executes as part of the login process. The table of terminal settings is found in a file called */etc/gettydefs*. You can edit *gettydefs* to add different sets of terminal characteristics or to change the existing ones.

The `gettydefs` File

The file `/etc/gettydefs` contains the information that `getty` uses to set up terminal line characteristics such as baud rate. The file is in the form of a table. Each table entry is divided into five fields. These fields include:

```
label # initial-flags # final-flags # login-prompt # next-label [# login program]
```

The fields are:

<i>label</i>	Identifies the <code>gettydefs</code> entry to <code>getty</code> . This could be a number or a letter. <i>label</i> corresponds to the line mode field in <code>/etc/ttyS</code> . <code>init</code> passes the line mode to <code>getty</code> as an argument.
<i>initial-flags</i>	Sets terminal line characteristics when <code>getty</code> first establishes the connection. <code>getty</code> recognizes the flags listed in <code>termio(M)</code> , in the <i>XENIX Reference</i> . Often the only flag in this field is the one setting the baud rate. For example, B300 would set the speed to 300.
<i>final-flags</i>	Sets the terminal line characteristics just before <code>getty</code> executes <code>login</code> . These flags describe the operating characteristics for the line. The baud rate (B) is set again. Other common flags include SANE (a composite flag that sets a number of terminal characteristics to reasonable values), TAB3 (expands tabs with spaces), IXANY (enables any character to restart output), and HUPCL (hangs up line on final close). Flags can be entered in any order.
<i>login-prompt</i>	Contains the login message that greets users. This field is printed exactly as it is entered, including spaces and tabs. An “@” symbol in the login-prompt field is expanded to the first line (or the second line if it exists) in the file <code>/etc/systemid</code> (unless the ‘@’ is preceded by a ‘\’).

Adding and Using Terminals

Several character sequences are recognized, including:

<code>\n</code>	Line feed
<code>\r</code>	Carriage return
<code>\v</code>	Vertical tab
<code>\nnn</code>	(3 octal digits) The specified ASCII characters
<code>\t</code>	Tab
<code>\f</code>	Form feed
<code>\b</code>	Backspace

next-label Identifies the next label in *gettydefs* for **getty** to try if the current one is not successful. **getty** tries the next label if a user presses the **BREAK** key while attempting to log in to the system. Groups of entries, such as dial-up or TTY lines, should form a closed set so that **getty** cycles back to the original entry if none of the entries is successful.

login-program The name of the program which actually logs users onto the system. The default program is */etc/login*. This field is optional.

If preceded by the keyword “AUTO,” **getty** does not prompt for a user name, but instead uses the device name (i.e. *ty03*) as the user name and immediately executes the *login-program*.

Each field is separated by a pound sign (#), and each entry in *gettydefs* is separated by a blank line.

An entry in *gettydefs* might look like the following:

```
4# B1200 # B1200 SANE TAB3 HUPCL #login: #2 #AUTO /etc/login.new
```

Here is a description of each part of this line:

- The number 4 identifies this entry to **getty**.
- The next field sets the baud rate to 1200.
- The third field indicates the baud rate (B1200), SANE (a composite flag for a number of characteristics), and HUPCL (hangs up line on final close).

- *login*: appears as the login prompt. If this setting is not successful, **getty** proceeds to label 2 in *gettydefs*.
- **AUTO** attempts to log in the user by executing */etc/login.new*.

Although the XENIX serial driver does not support baud rates exceeding 9600 baud, */etc/gettydefs* entries are provided for use with third-party serial drivers that do. These entries are lettered **n** and **o** and appear at the bottom of */etc/gettydefs*. These entries specify a baud rate of EXTA (19200) and EXTB (38400), respectively.

Changing the *gettydefs* File

The file */etc/gettydefs* has separate sets of entries for the dial-up lines and terminal lines. They are selected by the label characters in */etc/ttyd*. The **init** program passes the label character as an argument to **getty**.

You can edit *gettydefs* to add new terminal settings or to change existing ones. For example, the settings for terminal lines might look like the following:

```
4 # B2400 HUPCL # B2400 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login: # 5
5 # B4800 HUPCL # B4800 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login: # 6
6 # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login: # 4
```

To change the sample *gettydefs* file so that the first baud rate **getty** attempts is 1200, do the following:

1. Enter a text editor to edit the first line of the file *gettydefs*.
2. Change the first and third fields from B2400 to B1200.
3. Save *gettydefs* and exit the editor.

Adding and Using Terminals

The sample file should look like the next example:

```
4 # B1200 HUPCL # B1200 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login: # 5
5 # B4800 HUPCL # B4800 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login: # 6
6 # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login: # 4
```

You can also add additional terminal line settings to *gettydefs*. Flags and permissible values for terminal settings are listed in **stty(C)**, in the *XENIX Reference*.

When you add a new entry, be sure that the groups of entries in *gettydefs* form a closed set, so the *next-label* field of the last entry directs **getty** back to the first entry in the group.

To add an entry for a baud rate of 300 to the preceding sample *gettydefs* file, follow these steps:

1. Enter a text editor to edit the file */etc/gettydefs*.
2. Locate the point where you want to insert the new settings for *gettydefs*. The order of the entries does not matter; **getty** only looks for the label. In this example, the new entry is the last entry in the file.
3. Insert a carriage return after the last line in the file and enter the following on a new line:

```
7# B300 HUPCL # B300 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login: # 4
```

4. To incorporate label 7 into the set of labels, change the *next label* field for entry 6 to 7:

```
6 # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #r\n@!login:#7
```

getty is now directed from label 6 to 7, and then back to 4. Make certain that a blank line separates each pair of entries.

5. Exit the text editor, saving the revised *gettydefs* file.

The new *gettydefs* looks like the following:

```
4 # B1200 HUPCL # B1200 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n@!login: # 5
5 # B4800 HUPCL # B4800 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n@!login: # 6
6 # B9600 HUPCL # B9600 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n@!login: # 7
7 # B300 HUPCL # B300 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n@!login: # 4
```

Checking the Terminal Settings

Each time you change the terminal line settings or add new entries to *gettydefs*, you should check to make sure that the new values make sense to **getty**. To do this, you use the command **getty** with the check option, **-c**, and the filename.

For example, to check *gettydefs*, enter:

```
getty -c /etc/gettydefs
```

The file is scanned and the results are displayed. If any of the values and settings in *gettydefs* are not permitted, **getty -c** reports them. For more information on **getty** and *gettydefs*, see the **getty(M)** and **gettydefs(F)** manual pages.

Changing Serial Line Operation

Whenever you enable a terminal with the **enable** command, the system automatically sets the operating characteristics of the serial line to a set of default values. Sometimes these values do not match the values used by the terminal and, therefore, must be changed to allow communication between the system and the terminal. You can display the operating characteristics of a serial line with the **stty** (for “set tty”) command. If you need to change the characteristics of a port that is enabled, you should use the entries in the *gettydefs* file rather than the **stty** commands given below.

Note

Any settings on a port using the **stty** command only last as long as the port in question is still open. As an example, if you want to change the baud rate of `tty2a`, and `tty2a` is not enabled, the **stty** command first opens the port, then changes the port settings, and finally closes the port. When it closes the port for the last time, the settings revert to the original. In the **stty** commands later in this section, the use of the **while** loop is to avoid this behavior of **stty**. If you run **stty** without redirecting input, it works on your current serial line, which you have open. In this case, because the serial line stays open after the **stty** command, the settings also stay in place.

You can display the current operating characteristics of a serial line by entering this command at the terminal connected to that line:

```
stty
```

If it is impossible to log in at that terminal, you can use another terminal to display the characteristics. Log in as the super user at another terminal, and enter:

```
stty < ttyname
```

where *ttyname* is the name of the device special file corresponding to the serial line (see the “XENIX Directories and Special Device Files” chapter in this guide). For example, this command displays the current characteristics of the serial line named `/dev/tty1a`:

```
stty < /dev/tty1a
```

The command displays the baud rate, the parity scheme, and other information about the serial line. This information is explained in the **stty(C)** manual page.

One common change to a serial line is changing the baud rate. This is usually done from a terminal connected to another serial line because changing the rate disrupts communication between the terminal and the system. Before you can change the rate, you need to know the current baud rate of the terminal (review the terminal hardware manual to see how to determine the

current baud rate). Once you have the baud rate, log in as the super user at the other terminal, and enter:

```
(stty baud-rate ; while : ; do sleep 3600; done)< ttyname &
```

where *baud-rate* is the current baud rate of the terminal, and *ttyname* is the name of the device special file corresponding to the serial line you wish to change. The baud rate must be in the set 50, 75, 110, 134, 150, 200, 300, 600, 1200, 2400, 4800, and 9600. For example:

```
(stty 9600 ; while : ; do sleep 3600; done)< /dev/tty1a &
```

This command changes the baud rate of the serial line */dev/tty1a* to 9600. Note that the “less than” symbol (<) is used for both displaying and setting the serial line from another terminal.

Another common change is the way the system processes input and output through the serial line. Such changes are usually made from the terminal connected to the serial line. For example:

```
stty tabs
```

This command causes the system to expand tabs with spaces (used with terminals that do not expand tabs on their own). Another example is:

```
stty echoe
```

This command causes the system to remove a deleted character from the terminal screen when you back over it with the \langle Bksp) key.

Note that the **stty** command may also be used to adapt a serial line to an unusual terminal, to another type of serial device that requires parity generation and detection, or to special input and output processing.

For a full description of this command, see the **stty**(C) manual page.

Setting the Terminal Type

XENIX systems require that the terminal type be clearly defined before any work is done at the terminal. The preferable method for setting your terminal type is to assign the type to the **TERM** variable, a special environment variable that associates the terminal you are using with a list of characteristics given in the */etc/termcap* file. The characteristics tell the system how to interpret your terminal's keys and how to display data on your terminal screen.

Adding and Using Terminals

If you are using the Bourne or Korn shell (**sh**(C) or **ksh**(C)), the **TERM** assignment has the form:

```
TERM=termtype; export TERM
```

If you are using the C shell (**csh**), the **TERM** assignment has the form:

```
setenv TERM termtype
```

The *termtype* must be one of the names associated with one of the terminals defined in the */etc/termcap* file. The assignment must be entered at the terminal whose type you are setting.

For example, to set the terminal type to “ansi” from Bourne shell, go to the terminal you wish to set, enter at the shell prompt (“\$”):

```
TERM=ansi; export TERM
```

Press the <Return> key. From C shell, enter at the shell prompt (“%”):

```
setenv TERM ansi
```

Press <Return>.

If you are not sure which name you may use for *termtype*, you can view the names either by displaying the */etc/termcap* file, or by reading the **terminals**(M) manual page which lists all terminals supported in the */etc/termcap* file. To display the file itself, enter:

```
more /etc/termcap
```

Press the <Return> key.

You can let the system define the terminal type automatically whenever you log in by including the **TERM** assignment in your *.profile* file (see “Changing the *.profile* and *.login* Files” in the “Customizing System Startup” appendix of this guide).

For an alternate method of setting your terminal type, see the manual page for **tset**(C).

If you let the system set the terminal type, be careful when logging in on terminals that are not the same as your normal terminal. The system has no way of checking whether or not the terminal assignment is correct for the given terminal and assumes that it is the same as your normal terminal. If it is not, you must set the terminal type manually.

Setting the Terminal Type Automatically

If you want to have the terminal type set automatically at login time, follow this procedure:

1. Log in on the terminal in question and determine which *ttyname* you are using by entering the **tty(C)** command:

```
tty
```

2. Log in as *root* and edit the file */etc/ttytype* with a text editor. Change the terminal type field for the line associated with the terminal in question to the terminal type you desire to use. Follow the model for the console. If you want your terminal type to be set to 'wy50' for */dev/tty1a*, edit */etc/ttytype* as follows:

```
wy50 tty1a
```

3. Then the user's start up file must be edited with the appropriate **tset(C)** command line to set the terminal type automatically. In each C-shell user's *.login* file, add the following line:

```
set term=(`tset -m wy50:wy50 -m :`ansi -r -S -Q`)
```

In each Bourne or Korn shell user's *.profile*, add the following line:

```
eval `tset -s -m wy50:wy50 -m :`ansi`
```

Be sure to remove the existing **tset** command line from the *.profile* file.

4. Have all users log out, then log in again to test the new terminal type change. After they log in, have them verify the new term type by entering the **env(C)** command:

```
env
```

Removing a Terminal

From time to time it may be necessary to remove a terminal from the system, for example, if you wish to replace it with some other device. Before you can remove a terminal, you must disable it with the **disable(C)** command.

Adding and Using Terminals

To remove a terminal, follow these steps:

1. Turn off the power to the terminal.
2. Log in as the super user at another terminal.
3. Use the **disable** command to disable the terminal. The command has the form:

```
disable ttyname
```

where *ttyname* is the name of the serial line to which the terminal is attached. For example:

```
disable tty1a
```

This command disables the terminal connected to serial line */dev/tty1a*.

4. Disconnect the terminal from the system.

The serial line previously connected to the terminal is now free to accept another device.

Setting Up a Serial Console

You can configure a serial device, rather than a display adapter, as your system console. The **boot**(HW) program sets the default console at boot time according to the following procedure:

1. The **boot** program looks for the entry **SYSTTY=*x*** (*x* is the name of the system console device) in the */etc/default/boot* file.
2. If the **SYSTTY** entry is not found or the */etc/default/boot* file is not readable, **boot** checks your system for a display adapter and designates it as your system console.
3. If no display adapter is found, **boot** looks for **tty1a**, sets the serial port to 9600 baud, 8 data bits, 1 stop bit, and no parity, and uses it as the system console.

To set up a serial console, create the following entry in your */etc/default/boot* file (where *x* is “0” for a display adapter or “1” for a COM1 serial port):

```
SYSTTY=x
```

To change the system console device from the command line, enter **systty**=*x* at the boot prompt (where *x* is “0” for a display adapter or “1” for a COM1 serial port). This does not create or change a **SYSTTY** entry in the */etc/default/boot* file.

Using Serial Multiscreens with **mscreen**

If you are familiar with **multiscreen**(M), the feature that provides many separate login screens on the console, it is possible to use a similar feature on a terminal. Terminals that have multiple pages of screen memory can be used as separate screens, each with a different login session, as if you had several terminals at your service instead of one.

14

On a Wyse 60 terminal, the contents of two entire screens of activity can easily be saved. The use of a third screen on the Wyse 60 is discussed below. This means that using two screens is very much like having more than one terminal. The complete functionality of a login session is provided on each screen, and previously executed commands (or their results) are displayed on each screen when it is in use. This section focuses on the Wyse 60, using its two pages of screen memory as the basis for all examples. (See the **mscreen**(M) manual page for a technical explanation.)

You can also limit the number of **mscreens** available. The **mscreen** utility provides access to multiple terminal sessions, much like logging in on more than one terminal. These sessions are provided on “pseudo-ttys” rather than the tty devices usually used by terminals or modems. A tty is a special file associated directly with a particular hardware device used for communication with equipment such as terminals or printers. ttys can be seen in the */dev* directory as files with the name *tty* followed by a number and a letter.

A pseudo-tty is a device that is not associated with any real hardware, and it is used to simulate the function of a real tty. Users of networking products should already be familiar with pseudo-ttys, as they are the devices used to log in on remote machines. A pseudo-tty is represented by two software devices that appear in a listing of */dev* as “ptyp” and “ttyp”, each followed by a number. The former is called the “master” tty and the latter the “slave.” Between the two, they simulate a functional tty.

Adding and Using Terminals

As installed, the system allows a maximum of 16 screens system-wide, and only eight pseudo-ttys are created during the installation. You can create more pseudo-ttys and, as a result, allow more screens on your system, with the program in Figure 14-2. A limit on the number of available pseudo-ttys is controlled via a configurable parameter, NSPTTYS, which determines the number of pseudo-ttys that the system recognizes (the default at installation time is 16). The maximum is 32 pseudo-ttys. If you are planning to use more than 16 pseudo-ttys, you must increase this parameter in addition to creating the extra pseudo-ttys. Note that the “for x in 8 9 10...” line in Figure 14-2 extends to 23, meaning that it creates 16 extra pseudo-ttys (0-7 are already present). Be sure and shorten this line if you do not need that many pseudo-ttys. You can also extend it to 31 if you want the maximum of 32.

```
:`
# Type 'l /dev/ptyp*' You will get several lines similar to:
# crw-rw-rw-  1 bin    bin  55,  0 May 26 08:21 /dev/ptyp0
#
# Set the 'major_master' variable (below) equal to the major number
# given.
major_master=55

# Type 'l /dev/ttyp*' You will get several lines similar to:
# crw-rw-rw-  1 bin    bin  54,  0 May 26 08:21 /dev/ttyp0
#
# Set the 'major_slave' variable (below) equal to the major number
# given.
major_slave=54

for x in 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
do
    # make master
    /etc/mknod /dev/ptyp$x c $major_master $x
    chown bin /dev/ptyp$x
    chgrp bin /dev/ptyp$x
    chmod 666 /dev/ptyp$x

    # make slave
    /etc/mknod /dev/ttyp$x c $major_slave $x
    chown bin /dev/ttyp$x
    chgrp bin /dev/ttyp$x
    chmod 666 /dev/ttyp$x
done
```

Figure 14-2 makeptys Shell Script

Follow these steps to configure the additional screens:

1. Configure your kernel to support the additional pseudo-ttys. Read “Reallocating Kernel Resources with configure” in the “Tuning System Performance” chapter of this guide for instructions on how to run **configure(ADM)**. Select category 3, “Files, Inodes, and

Filesystems,” and set the NSPTTYS parameter to the number desired. Follow the instructions on relinking the kernel before continuing.

2. Copy the script in Figure 14-2 into a file called *makeptys*. (The lines beginning with a “#” are comments and need not be copied.) To make this shell script executable, enter the following command:

```
chmod 744 makeptys
```

3. To run this shell script, make sure you are logged in as *root*, and execute the shell script as follows:

```
./makeptys
```

4. After adding the devices, you must configure the system to start a **getty** process at each pseudo-tty. This is accomplished by adding entries to the */etc/ttys* file. The entries already present for the first eight pseudo-ttys look like this:

```
1mttp0
1mttp1
1mttp2
1mttp3
1mttp4
1mttp5
1mttp6
1mttp7
```

14

Note

The first eight pseudo-ttys may have a “0” instead of “1.” (Mice require the pseudo-ttys to be disabled.) You can change them to “1” by enabling each of the devices as in the following example:

```
enable ttyp0
```

This enables pseudo-tty */dev/ttyp0*. (Do not enable these pseudo-ttys if you are already using them for mice.)

If you have XENIX-NET installed on your system, the pseudo devices ttyp8-15 are already available; do not create these devices.

Adding and Using Terminals

You can copy and then modify these entries for the additional devices. Make certain you modify both files. For example, if you wanted to add an additional eight pseudo-ttys, you would add the following lines to */etc/ttys*:

```
lmttyp8
lmttyp9
lmttyp10
lmttyp11
lmttyp12
lmttyp13
lmttyp14
lmttyp15
```

Adding more **msscreen** capability to your system should increase the productivity of the users. However, too much of a good thing can slow your system down. A system with 10 users, all of whom use two screens, could make your system perform as though it is servicing 20 users. Keep system performance in mind when deciding how many msscreens should be allowed system-wide, and who should be able to use them.

While no terminal known contains enough screen memory to save the material displayed during the use of all 20 logins that **msscreen** is capable of, any terminal should allow the user to switch between as many as 20 screens, providing the keyboard has enough extra keys to indicate the switch between screens. However, the user will probably not find multiple screens very useful without multiple pages of screen memory. It is inconvenient, for example, to have to redraw the terminal's screen each time one switches screens when using a spreadsheet on one screen, and vi on the other. Most people who use terminals with minimal screen memory prefer shell layers **shl(C)** to **msscreen** for multiple login sessions. For more information, see the **shl** manual page.

Troubleshooting

The following suggestions should enable you to avoid some of the more common mistakes made by new **msscreen** users.

In preparing to use **msscreen**, make sure your terminal works with the program. Find out how much screen memory is provided by consulting your terminal manual. The **msscreen** utility uses the file */etc/msscreencap* to determine how to change screen images for your particular terminal. As shipped, */etc/msscreencap* is supplied with only a few terminals. This is not to say that other terminals do not work with **msscreen**; they do. You need only configure the */etc/msscreencap* file before using your terminal. If you run **msscreen** on a terminal that does not have an entry in */etc/msscreencap*, **msscreen** fails.

If you are sure your terminal works with **mstty**, and you have a working *mstty*, but **mstty** still fails, check the following common problems:

- Create pseudo-ttys.

You may need to create more pseudo-ttys if the pseudo-ttys currently on your system are in use. Use the shell script in Figure 14-2 to create more pseudo-ttys.

- Remember mice also use pseudo-ttys.

As discussed in the procedure on configuring additional pseudo-ttys, mice require pseudo-ttys to be disabled. If there is a conflict, screen switching will result in seeing a shell prompt instead of a login prompt.

- Verify switching.

Make sure the */etc/mstty* for your terminal is correct. Use one of the examples in */etc/mstty* to check the way your function key output sequence is mapped to a particular **mstty** command. You must log in separately to each screen you intend to use.

- Kill **mstty** processes.

If you are testing an *mstty* entry and you have trouble with the screens, you should do the following:

1. Check the processes that are running:

```
ps -uusername
```

2. Kill all the **mstty** processes:

```
kill -9 process_numbers
```

Advanced Uses

Many users find **mstty** satisfactory as provided. For advanced **mstty** users, or anyone interested in learning more about both **mstty** and the operating system, here are some “tuning” tips for using and extending **mstty**.

Adding and Using Terminals

In addition to invoking **m**screen automatically, the script in Figure 14-3 allows three full-featured **m**screens on a Wyse 60 and adds a number of convenience features for the **m**screen user. Figure 14-4 presents the same material for the Bourne and Korn shell *.profile* file. Note that these examples are designed to be added to the end of your *.login* or *.profile* file, and replace any existing *tset* material.

```
#
# Example material for the end of a C-Shell .login file.
#
# If logging in via pseudo-tty, suppress terminal initialization.
set ttyname='tty'

# Set init to null, initially.
set init = ""

set noglob

# Reset init to the value "-I" when logging in on a pseudo-tty to
# suppress the tset terminal initializations string.

if ( `expr $ttyname : "/dev/tty*" ` > 0 ) set init = "-I"
set term = (`tset -m ansi:ansi -m wy60:wy60 -m:\?wy60 -r -S -Q $init`)
setenv TERM $term[1]
setenv TERMCAP $term[2]          # terminal data base
unset noglob term
# Put WYSE 60 in ECON-80 mode during initial log in process.
if ( "$init" != "-I" && "$TERM" == "wy60" ) /bin/echo "\033eG\c"

# Set the prompt to indicate the tty number of the current
# mscreen and command.
set prompt = "`expr $ttyname : '/dev/\(.*)' ` \!% "

# Release the local variables used.
unset ttyname init

# Run mscreen and logout if the 'stop' key (defined as S-F9 in
# the default /etc/msscreencap for wy60) is pressed. This string
# is described in the mscreen(M) manual pages.
mscreen -n 3
if($status == 0) logout
```

Figure 14-3 .login Script

```

#
# Example material for the end of a Bourne shell .profile file.
#
ttyname=`tty`

init=""

if [ `expr $ttyname : "/dev/tty*" -gt "0" ]
then
    init="-I"
fi

eval `tset -m ansi:ansi -m wy60:wy60 -m :?\wy60 -r -s -Q $init`
export PATH

if [ "$init" = "-I" -a "$TERM" = "wy60" ]
then
    /bin/echo "\033eG\c"
fi

PS1=`expr $ttyname : '/dev/\(.*)' ` $ "

unset ttyname init

mscreen -n 3
if [ "$?" = "0" ]
then
    exit
fi

```

Figure 14-4 .profile Script

Many termcap entries (including wy60) clear the screen buffers (that **mscreen** uses to store the contents of multiple screens) as part of the initialization string. In Figures 14-3 and 14-4, **tset(C)** sends the initialization string only during the first login procedure. When logging in on pseudotys, **tset** is invoked with the **-I** flag. This is done by adding the **init** variable to the **tset** line. The first time **tset** is run, **init** has a value equal to "", adding nothing to the **tset** command. When it is run subsequently, **init** has a value of "-I", adding the option to **tset**.

Following the **tset** command, during first login procedure, the string "\033eG\c" is echoed. This escape sequence changes the COLUMNS setting in the Wyse 60 to ECON-80 mode. The combination of these settings frees up just enough screen memory to use three screens. As an extra convenience, the user's prompt is set to display the current slave pseudotty number, allowing the user to keep track of which screen is in use easily.

If you do not use a Wyse 60 terminal, you can still set your prompt to indicate the current screen, and invoke **mscreen** automatically while checking for the shell return code, as illustrated in Figures 14-3 and 14-4.

Installing and Using Modems

Modems (from **mod**ulate **demod**ulate) are a significant addition to your system, allowing you to communicate over phone lines from remote sites. You can connect a modem to your system via a serial port. Before adding a modem, you should make certain you have a port available, either directly on COM1 or COM2, or from a multiport card. Configuring serial ports is discussed in the “Adding Multiport Cards, Memory, and Other Bus Cards” chapter of this guide.

Choosing a Serial Port

The system supports modem control on serial ports. Table 14.1 contains sample device names of serial ports with and without modem control.

Table 14.1
Serial Ports

Device	Function
<code>/dev/tty1a</code>	main serial adapter without modem control
<code>/dev/tty1A</code>	main serial adapter with modem control
<code>/dev/tty2a</code>	alternate serial adapter without modem control
<code>/dev/tty2A</code>	alternate serial adapter with modem control

`/dev/tty1a` and `/dev/tty1A` refer to the same serial port (likewise for `/dev/tty2a` and `/dev/tty2A`). The operating system uses different device-driver subroutines for each. Never attempt to use both modem and non-modem control ports at the same time or you will see the warning:

```
cannot open: device busy
```

Multiport serial cards may use a different convention from Table 14.1. Some use the devices `/dev/tty[1,2][a-m]` for use without modem control, and the devices `/dev/tty[1,2][A-M]` for use with modem control.

Configuring Your Modem

Proper modem configuration is necessary when using **cu(C)** and **uucp(C)**. Modem settings differ for each modem. Consult your modem manual for the proper switch settings.

Smartmodem 1200

If you have a Hayes Smartmodem 1200 or compatible, switches 3 and 8 should be down:

	1	2	3	4	5	6	7	8
up	•	•		•	•	•	•	
down			•					•

When switch 3 is down, the resulting codes are sent to (echoed by) the modem to the terminal or computer. When switch 8 is down, the modem can interpret the command being issued. This allows both MS-DOS and XENIX communications systems to work.

14

Smartmodem 2400 or V-series Smartmodem 9600

The Hayes 2400 and V-series 9600 Smartmodem or compatible modems require online configuration if they are to be used as a dial-in line. Note that the Hayes 2400 does not answer the phone with a 2400 baud carrier if it is not set up with 2400 baud commands. Follow the instructions for “Installing a Modem” so that you can establish a connection with the modem. When you have completed the procedure, you must then issue set up commands via **cu(C)**. The form of the **cu** command is:

```
cu -s2400 -ltty $nn$  dir
```

where *nn* is the tty number of the serial line. To configure a modem on tty1a, enter this command and press (Return):

```
cu -s2400 -ltty1a dir
```

Installing and Using Modems

Next, enter the following commands to configure the modem. They are saved in the modem's nonvolatile memory. If you do not want to save the settings, do not enter the last command (AT&w). Follow each command with a `<Return>`.

AT&f	fetches factory configuration.
ATT	is tone dialing.
ATI0	is low speaker volume.
AT&d2	sets dtr "2": goes on hook when dtr drops.
AT&c1	sets dcd "1": dcd tracks remote carrier.
ATs0=1	answers phone after 1 ring (AA light should come on).
ATs2=128	disables modem escape sequence.
ATe0	does not echo (modem no longer echoes what is sent to it).
ATq1	is quiet mode (modem does not respond with "OK" after this command or any that follow).
AT&w	saves settings in nonvolatile memory.

Exit from `cu(C)` by entering a "tilde" and a "period," followed by a `<Return>`:

~.

The modem is now configured and ready for use.

Telebit Trailblazer

If you have a Telebit Trailblazer modem or compatible, log in as `root` and enter the following command:

```
/usr/lib/uucp/dialTBIT -z /dev/ttynn 9600
```

where `nn` is the tty number of the serial line.

Installing a Modem

Note

We strongly advise that you do not use an internal modem. We have successfully tested the Hayes 1200 and 2400 baud internal modems, and these modems are known to work. However, most Hayes-compatible internal modems have compatibility problems which prevent them from working properly under XENIX. These incompatibilities include:

- No support for “quiet mode,” “no-echo mode,” or auto-answer mode.
- Modem floods the bus with spurious interrupts.

14

When you are hooking up your modem, or any other device, make sure that serial wires connected to your computer are not left hanging. An unterminated line connected to your computer can considerably reduce system performance; always unplug a modem wire at the computer end instead of at the modem end.

Three-wire cables often used to connect terminals to the computer are not sufficient for connecting modems. For a modem cable on a 25-pin serial port, pins 2, 3, 7, 8, and 20 must be connected straight through. If you are unsure as to what to use, a cable that connects all pins works correctly. Either a ribbon cable, or what is called a “straight-through” cable, meaning that it connects the pins straight across, works.

To install your modem for dial-in or dial-out, follow these steps:

1. Make sure the UUCP package is installed. Use **custom(ADM)** to install it if necessary.
2. Make sure the serial port you have chosen for your modem is recognized at bootup (check */usr/adm/messages* or use **hwconfig(C)**) and, if the modem is internal, make sure that the COM port the internal modem is configured for does not conflict with any other device. Only serial devices attached to COM1 and COM2 are supported.

Installing and Using Modems

3. Make sure the modem and non-modem control ports are disabled by using the **disable(C)** command. For example, the following command applies to a modem connected to COM1:

```
disable tty1a  
disable tty1A
```

4. If you are going to use the line for dial-out, the serial port must be owned by *uucp*. To make sure the line is owned by *uucp* enter this command:

```
chown uucp /dev/ttynn  
chgrp uucp /dev/ttynn
```

where *nn* is the number of the corresponding line.

5. Connect the modem to the machine using a straight-through cable (pins 2 and 3 are not crossed). The cable must have at least pins 2, 3, 7, 8, and 20 connected.

Most standard COM ports use straight-through cables, but some hardware requires a null-modem cable (pins 2 and 3 are crossed). A standard COM port is known as DTE (Data Terminal Equipment); a port that needs a null-modem cable is known as DCE (Data Communications Equipment). Check your hardware documentation if you are unsure. If the COM board is a DCE, you need a null-modem cable.

6. If you intend to use the modem for dial-in, check that the entry for this serial port in the */etc/ttys* file looks like the following:

```
03tty1A
```

The “3” in the second field selects the **gettydefs** entry for the 2400-1200-300 cycle. Other *gettydefs* values include “2” for 1200 baud, and the letter “m” for 9600 baud. This is necessary if the line is to be used for dial-in and is ignored when dialing out. Information on the format of the */etc/ttys* file can be found in the **ttys(F)** manual page.

Note

Never alter the first digit of the */etc/tty*s file; the system maintains this field to indicate whether the port is enabled or disabled.

7. Add the correct entries to the */usr/lib/uucp/Devices* file. This file should have two entries for each serial port being used for a modem. One of the entries is used when you start a call using the modem (the ACU line), and the other line is used to configure the modem using the standard Hayes command set (the Direct line). (ACU stands for Automatic Calling Unit.) This is the default *Devices* file supplied with your system (XENIX-386 only), which is set up for a Hayes-compatible modem operating at 2400 baud, using COM1:

```
ACU  tty1A - 1200-2400 /usr/lib/uucp/dialHA24
Direct  tty1a - 1200-2400 direct
```

Make sure that the entries do not have a pound sign (“#”) in front of them. This is the syntax to show that the line is only a comment, and is to be ignored. There are many examples in the *Devices* file that are commented out with this character. Entries that begin with a blank space are also ignored; make sure entries you wish to use have no leading spaces.

8. Test the modem’s ability to dial correctly by entering the following command. For example, the following command uses *tty1a*:

```
cu -ltty1a dir
```

You should see a message indicating that you are connected. If you see the message “cu: dir permission denied,” the user executing the **cu** command does not have write permission on the */usr/lib/uucp/Devices* file.

If you do not see such a message, and there was no message to indicate that you connected correctly, then the **cu** command is incorrect, the *Devices* file is incorrect, or the serial port is not operating correctly.

Note

The instructions that follow assume a Hayes-compatible command set and response codes. Other modems may use other conventions. Consult your modem documentation for further details.

9. If you see a message confirming your connection, enter the following on your keyboard:

AT

The screen should echo "OK." If the modem is set to return result codes as numeric codes rather than text, you see "0."

10. If this does not occur, check that the "receive" light on the modem flashes when you press a key. This indicates the modem is receiving signals from the keyboard. If this light is not flashing, check your cable and modem switch settings. If the "receive" light flashes, but you still do not get an "OK" response from the modem, enable the modem's echo capability and responses to commands by entering:

ATE1Q0

11. If your terminal still does not display "OK" or "0," connect the modem to another port (COM1 or COM2). If the modem works with the new port, check that the device is using the correct interrupt vector. (Serial port COM1 uses interrupt 4; COM2 uses interrupt 3.)
12. If the terminal does not echo the "OK" message when it is connected to the new port, the modem is defective; see your hardware documentation.

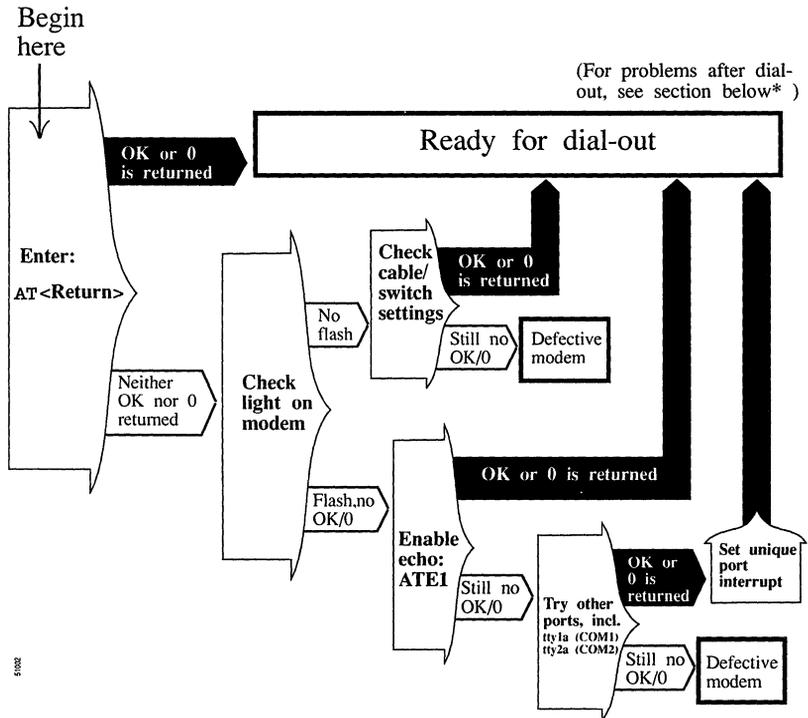


Figure 14-5 Testing Your Modem Connection

13. If you plan to use the modem primarily for dial-in, skip to step 15. To test the modem for dial-out, enter:

ATDT phonenumber

When you confirm that the modem can dial out, exit **cu** by entering:

Then press **<Return>**.

You are now ready to dial in to another system. Use the following command to dial out:

cu -l tty1A 555-1212

Change “555-1212” to the phone number of the system into which you are dialing.

Note

When including a phone number in the **cu** command to connect with a modem, you should not use the comma (,) from the Hayes command set to indicate a pause. Use hyphens instead. This also applies to *Systems* file entries.

If you have any problems, refer to the section on troubleshooting your dial-out modem in the “Troubleshooting Your System” chapter in this guide.

14. If the line is also to be used for dial-in, you must follow some additional steps to configure the modem. Some modems have switches or software commands for setting the modem configuration. If your modem has such settings, configure it as instructed in your modem manual. The modem should be configured as follows:
 - The modem must be set to Auto-answer. Your modem must support auto-answer; most internal modems do not have auto-answer and some external modems do not have this setting.
 - It should not answer when the DTR (Data Terminal Ready) line is not active, and disconnect from the current connection when DTR goes from active to inactive.
 - The CD (Carrier Detect) line should follow the incoming carrier; it should go low when a carrier is present, high when a carrier is not present.
 - Echo should be turned off; it should not echo characters sent to it.
 - It should be set for quiet mode, sending response codes instead of response strings.

See “Configuring Your Modem” for instructions on setting these modes for Hayes 2400 or V-series 9600 modems.

15. Enable the port you are using for your modem with the **enable(C)** command:

```
enable ttyname
```

where *ttyname* is the modem control port.

16. Dial this modem from another modem.

If you have any problems, refer to the section on troubleshooting your dial-in modem in the “Troubleshooting Your System” chapter in this guide.

Dialing In to Your Computer

To allow dialing in to your computer, you must enable a serial line that recognizes modem control signals with the **enable(C)** command.

14

To use the main serial adapter (COM1), enter:

```
disable tty1a  
enable tty1A
```

Or, for the alternate serial adapter (COM2), enter:

```
disable tty2a  
enable tty2A
```

Note that **tty1A** and **tty1a** refer to the same (main) serial line, and **tty2A** and **tty2a** refer to the same (alternate) serial line. Do not enable the same line in both its modem and non-modem modes at the same time, as this causes an error.

Adding Passwords for Dial-in Lines

If desired, you can define special dial-in passwords on selected tty lines, requiring selected classes of users to input dial-in passwords. Logging information, including the last time of connection, can be stored for later use. See “Adding Dial-in Password Protection” in the “Maintaining System Security” chapter of this guide for information on using dial-in passwords.

Dialing Out from Your Computer

The **cu**(C) and **uucp**(C) utilities call remote systems and transfer data on XENIX systems. The file */usr/lib/uucp/Devices* (referred to as *Devices*) contains information used by these programs to determine the characteristics of a particular serial line.

The *Devices* file contains lines that specify the device for the line, the call-unit associated with the line, and the baud rate, which are to be used by UUCP. (Modem control devices should be used with lines connected to modems.)

Using Dialer Binaries

For dialing, both **cu** and UUCP use a common set of dialers, which can be standalone binaries (programs) like */usr/lib/uucp/dialHA12*, or entries from the file */usr/lib/uucp/Dialers*. (For more information on *Dialers* file entries, see the “Building a Remote Network with UUCP” chapter in this guide.) Table 14.2 lists the binary types available in */usr/lib/uucp*.

Table 14.2
Dialer Binaries

Binary File	Modem
dialHA12	Hayes Smartmodem 1200 or compatible
dialHA24	Hayes Smartmodem 2400 or compatible
dialHA96V	Hayes Smartmodem 9600 or compatible
dialMUL	Multitech Multimodem 224 EH
dialVA3450	Racal Vadic 3451 modem
dialVA96	Racal Vadic 9600 modem
dialTBIT	Telebit Trailblazer Modem

Note

Some dialers shown are not on XENIX-286 distributions.

The source for these dialer programs and a make file for recompiling the source program are included in the directory `/usr/lib/uucp`. If you have any other kind of modem, you can modify any of the source files and create your own dialer program. Note that you must have the *Development System* installed to compile a program.

To make a new dial program, follow these steps:

1. Change directory to `/usr/lib/uucp` with the following command:

```
cd /usr/lib/uucp
```

2. Edit the file `makefile` in the directory `/usr/lib/uucp` and find the line that reads:

```
EXES= dialHA12 dialHA24 dialMUL dialTBIT dialVA3450
```

Add the name of the dialer program that you wish to use. When this is done, exit the file, saving the changes you made.

3. Next, enter the following command at your shell prompt:

```
make
```

Then press (Return).

When the **make** command is finished, you have a new dialer program. This can be used in the fifth field of an entry in the *Devices* field.

Chapter 15

Using Printers

- Introduction 15-1
 - The Printer Spooling System 15-1
- Installing a Printer 15-3
- Adding a Local Printer 15-12
- Print Service Command Summary 15-16
- Starting and Stopping the Print Service 15-18
 - Manually Stopping the Print Service 15-18
 - Manually Starting the Print Service 15-19
- Managing the Print Service 15-20
 - Canceling a Print Request 15-20
 - Enabling and Disabling Printers 15-20
 - Adding a Printer to a Class 15-21
 - Setting the System Default Destination 15-22
 - Displaying Printer Status 15-23
 - Removing a Printer or Class 15-23
- Managing the Printing Load 15-25
 - Rejecting Requests for a Printer or Class 15-25
 - Accepting Requests for a Printer or Class 15-26
 - Moving Requests to Another Printer 15-26
 - Examples 15-27
 - Example 1: Moving Requests for Maintenance 15-27
 - Example 2: Temporary Load Shifting 15-27
 - Example 3: Preventing Requests for Maintenance 15-28
- Using Interface Programs 15-29
 - How to Write an Interface Program 15-30
 - What Does an Interface Program Do? 15-30
 - How an Interface Program is Used 15-30
 - Customizing the Interface Program 15-31
 - Adjusting the Printer Port Characteristics 15-33
 - How to Add an Interface Program 15-35

Setting Up RTS/CTS Protocol Serial Printers 15-36

Using a Printer Without the Spooler 15-39

Creating an Init Device File 15-40

Introduction

Printers are a useful addition to any computer system. Most systems require the ability to print out data on paper. A wide variety of printing hardware or line printers are supported. Some line printers are *parallel* devices, but most are connected as *serial* devices.

To add a printer, the system administrator must:

- connect the physical hardware to the computer, then
- use the correct system commands to enable the printer for operation.

This chapter explains how to do this and how to maintain printers once they are added. Note that physical connections between a printer and the system vary depending on hardware configuration. This chapter provides some information about making the necessary physical connections, but for more information about these connections, see the hardware manuals provided with the printer and your computer.

The operating system supports printers that use the standard RS-232 interface and Centronics parallel interface.

15

The Printer Spooling System

The XENIX line printer spooling system is a collection of commands that help you, as system administrator, to install, monitor, and control efficiently the line printers serving your system. A request to print a file is *spooled* or lined up with other printing jobs to be sent to the printer. Each print job is processed and waits its turn in line to be printed, thus the term *queue*.

When a user requests a file to be printed using the **lp(C)** command, the line printer system responds with a “request ID.” This consists of the name of the printer on which the file is printed and a unique number identifying the file. With this request ID, the user can find out the status of the print request or cancel it. The **lp** options help the user to control printer output easily.

Introduction

The print service performs the following functions:

- handles the task of receiving files users want printed
- schedules the work of one or more printers
- starts programs that interface with the printer(s)
- keeps track of the status of jobs
- issues error messages when problems arise

There are several terms used in this chapter to describe the operation of the print service:

<i>device</i>	The target for lp output. A device is represented by a special device file (see the “XENIX Directories and Special Device Files” chapter of this guide).
<i>printer</i>	The name assigned by the system administrator to represent a device. This name can have up to 14 characters. At different times, a printer can be associated with different devices.
<i>class</i>	An ordered list of printers. Print requests sent to a class of printers are printed by the first available member of that class.
<i>destination</i>	A place where print requests are sent. A destination can be a class or a printer.

Consult your computer and line printer hardware manuals for information on making the connection between your system and printing devices.

Installing a Printer

This section instructs you on how to install new printing devices on your XENIX system. You must connect the printer to a proper port (serial port for serial printers, parallel port for parallel printers), ensure that it works, and set up the XENIX printer spooling software using the **mkdev lp** command.

Note

If you have any problems connecting your printer, see the “Troubleshooting Your System” chapter in this guide for helpful information.

Follow the steps below to install a printer:

1. Find a place for your printer and make sure that it is properly assembled and plugged into a power outlet.
2. **If you are connecting a serial printer:** connect the RS-232 cable from a computer serial port to the port on your printer. Serial printers must be capable of supporting XON/XOFF or DTR (Data Terminal Ready) protocols and must be configured for those protocols. Consult your printer owner’s manual for more information. Next, enter the following command substituting the correct port number for *nn*:

```
disable /dev/ttynn
```

Press <Return>. This disables logins on the port you have connected to your printer and allows the port to be used for serial communication.

3. **If you are connecting a parallel printer:** The printer must use a standard Centronics interface cable. The parallel port on a monochrome card should be configured for interrupt vector 7. (Consult “Adding Multiport Cards, Memory, and Other Bus Cards” for more information on adding cards and configuring interrupt vectors.) The parallel port on a monochrome card is automatically

Installing a Printer

recognized as **lp1** when booting up. The main parallel port (built-in as opposed to being on a monochrome card) should be configured for interrupt vector 7 and is recognized as **lp0**. You must use either the main or the monochrome port - not both - to avoid a hardware conflict. The alternate or second parallel port should be configured for interrupt vector 5, and is automatically recognized as **lp2**. Make sure no other hardware is using these interrupts. (See your hardware manual for information on configuring your parallel ports.)

4. Verify that you have correctly hooked up the printer by sending data directly to the device. (This procedure is shown in flowchart form in Figure 15-1 at the end of this section.) Enter one of the following commands:

For serial printers:

```
date > /dev/ttynn
```

where *nn* is the number of the serial port you are using (for example, */dev/tty1a*).

For parallel printers:

```
date > /dev/lpn
```

where *n* is the number of the parallel port you are using (for example */dev/lp0*).

Note

For certain laser printers (for example: HP LaserJet) you need to send a form feed to the printer. Use the following command, where *xxx* is the tty (serial) or lp (parallel) port:

```
(cat foo ; echo "\f") > /dev/xxx
```

5. If you do not see the date printed on your printer, there is most likely some type of hardware malfunction, so verify the following:

For serial printers:

- Make certain you are using the non-modem control device, for example: `/dev/tty1a`, not `/dev/tty1A`. (For more information on the naming convention for serial ports, see `serial(HW)`.)
- Try using a cable with only pins 2, 3, and 7 connected.
- Recheck your printer configuration by verifying its switches in your printer hardware manual.
- Make certain that the system recognizes your serial port. You can verify this by running the `hwconfig(C)` command, or checking the file `/usr/adm/messages`. If your port is one of the non-intelligent boards supported by the built-in serial driver, you see a line similar to this:

```
%serial 0x03F8-0x03FF 04 - unit=0 type=Standard nports=1
```

If “unit=0” is displayed, the serial port is considered COM1. If the unit is 1, it is considered COM2. The “nports=” denotes how many ports the driver recognized on the board. If you connect your serial printer to the first port on COM1, the associated device name will be `tty1a`. The second device on COM1 is `tty1b`, and so on. Devices on COM2 are named `tty2a`, `tty2b`, etc.

Intelligent serial boards using 3rd-party drivers may display different bootup messages specific to their drivers; they may also use a different scheme for device names. Here is an example of such a line:

```
%ONBOARD 0x0230-0x023F 34 0 unit=0 mem=0x000D0000 nport=16
```

Be sure to read the documentation for the board and its drivers before attempting to install serial devices such as printers.

Installing a Printer

- Recheck your switches on your serial port. If you are using a multiport card, try other lines on that card and be sure it does not conflict with the standard COM ports.
- Try attaching the printer to a standard serial port, COM1 or COM2, to see if the printer and cabling are correct.

For parallel printers:

- Make certain your cable is securely connected and all wires are good. Using the cable on a known good system and printing under DOS are good ways to test this.
- Recheck your printer configuration by verifying its switches in your printer hardware manual.
- Recheck the switches on your parallel card. It must also be recognized at bootup. You can verify this by running the **hwconfig(C)** command, or checking the file */usr/adm/messages* for a line similar to the following:

```
parallel 0x378-0x37A 07 - unit=0
```

To confirm that your card is recognized, enter the following command:

```
hwconfig name=parallel
```

If the card is recognized, an entry will be printed that has similar information to the entry above. The interrupt vector is listed in column 3 (or with “vec=” in the case of the **hwconfig** display); make certain it does not conflict with other hardware.

- Two default printers are already configured on your system; if your configuration matches these defaults, you need not run the **mkdev lp** program to configure your printer. The default printers are shown in Table 15.1.

Table 15.1
Default Printers (XENIX-386 Only)

Printer Name	Type	Interface Script	Port
laser	serial	hpjet	COM1 (tty2a)
default	parallel	crnlmap	lp1

For example, if you have connected an HP LaserJet as a serial printer on */dev/tty1a*, you can simply enable the printer and it is ready for use:

enable laser

The same is true for the default parallel printer, “default.” You can also choose to customize these default printers by using the “Reconfigure an existing printer” option of **mkdev lp**.

15

- If the default printers do not match your configuration, you should run **mkdev lp**. Enter the following command:

mkdev lp

Δ **sysadmsh** users select: System→Add→Printer

- If the scheduler is running you are reminded that any jobs which are printing may be interrupted and you are prompted if you want to continue.
- The following menu is displayed:

```

Do you wish to:
  1. Add a new printer.
  2. Remove a printer.
  3. Reconfigure an existing printer.
  4. Assign a default printer.
  5. Print lp status information.

Select an option or enter q to quit:
    
```

Enter **1** and press <Return>.

Installing a Printer

10. Next you see the following menu:

```
1. Parallel Printer
2. Serial Printer
3. Remote Printer
```

```
Select the type of printer you are adding
or enter q to return to the previous menu:
```

Enter your choice and press `<Return>`.

11. The following message is displayed:

```
Enter a name for the printer, press <RETURN>
to use the default name (printer) or enter q to
return to the previous menu.
```

Printer names can be up to 14 characters long and can be any combination of numbers, letters, or underscore characters. Enter the printer name and press the `<Return>` key.

12. If you specified that you are adding a remote printer, you now see the following messages and you are prompted to answer questions about the remote printer. If you specified Serial or Parallel you will not see these messages. For remote printers, you see:

```
Enter the node name of the remote computer to which your printer
is connected or enter q to return to the previous menu:
```

The “node” name is the name of the machine that owns the printer in question. Next you see:

```
Enter the name of the printer you wish to use on node_name
or enter q to return to the previous menu:
```

Next you see the message:

```
Is node_name connected via (M)icnet or (U)ucp?
```

Enter “M” or “U” depending on the network your system uses. Once you have answered these questions for your remote printer, you see the “Printer enabled and accepting requests” message described below. You do not see the other messages described here

for serial and parallel printers. Note that this is to set up printing from a remote machine. You must also run **mkdev lp** for the printer locally on the machine where it is physically connected.

13. The following message is displayed:

```
If you have an unusual printer you must create an
interface program in /usr/spool/lp/model. For a sample
interface program look at /usr/spool/lp/model/dumb.
```

14. Now you are prompted for a printer type. The screen displays a numbered list of the available printer types on your system, followed by these instructions:

```
Enter one of the interface programs above or the
full pathname of an interface program or enter q
to return to the previous menu:
```

The section “Using Interface Programs” later in this chapter describes the scripts available and how to customize them. Enter one of the options, then press **<Return>**.

15

15. If you are adding a parallel printer, you see the following:

```
Some printers require conversions for line feed, tab and form feed
(see lpinit (ADM) for more information) . Does this printer need this
special handling? (y/n)
```

Some printers require conversions for line-feeds, tabs and form feeds. In interpretive mode, the system sends line-feeds as carriage-returns; tabs as the appropriate number of spaces; and form-feeds as the appropriate number of carriage-returns. In non-interpretive mode (the default value), the system sends every character to the printer unmodified. Unless you have been told to do otherwise for your printer, enter **n** and press **<Return>**.

Installing a Printer

16. After you have finished responding to these questions, the following message is displayed:

```
destination printername now accepting requests  
printer printername now enabled
```

printername is then automatically enabled and is ready accept print requests.

17. After enabling the printer, you are prompted to respond to one more question:

```
Is this the default printer ? (y/n)
```

You can enter **y** (for yes) or **n** (for no) depending upon whether you want user print requests to be automatically routed to the printer.

18. One last message is displayed:

```
If you need to modify your interface program  
after installing it, the installed version is  
in /usr/spool/lp/interface/printername.
```

After you have responded to these questions, **lpsched**(ADM) is automatically started, and users can begin printing files on the new printer.

You can also add printers to your system using the **lpadmin**(ADM) command. However, you will need to give separate commands to stop **lpsched**, to enable the printer, and to allow it to accept print requests. For more information on these programs and commands, see sections “Starting and Stopping the Print Service,” and “Managing the Print Service.”

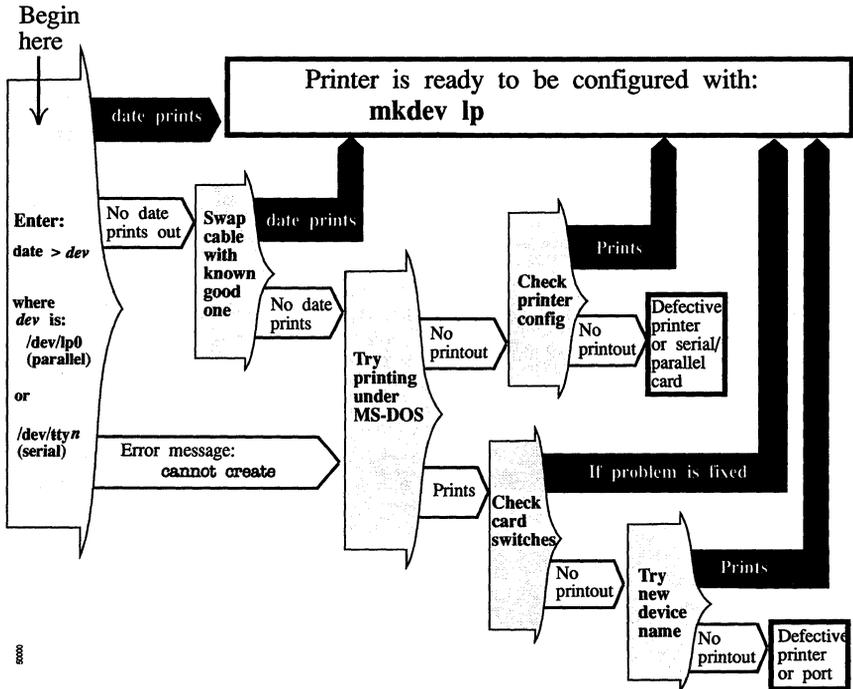


Figure 15-1 Testing Your Printer Connection

Adding a Local Printer

The operating system also supports the use of local printers attached to the AUX or PRINT port on the back of a normal serial terminal. These printers are connected via standard RS-232 connections and can significantly reduce the load on shared system resources. The **lprint(C)** command is used to print files on a local printer, but the terminal must be properly configured for the command to work. To add a printer connected to the AUX or PRINT port on the back of a terminal and use it for local printing, follow this procedure:

1. Connect your local serial printer to the AUX port on your terminal with a standard RS-232 cable with pins 2,3, and 7 (minimum) connected. Make sure the printer is powered on and is online.
2. Log in on the terminal and verify that it is working correctly.
3. Make sure that the AUX port on your terminal is configured with the same settings as your printer (baud rate, parity, data bits, xon/xoff, and so forth).
4. For the **lprint** command to work, **lprint** needs to know how to start and stop local printing for each specific terminal. **lprint** looks in the file */etc/termcap* to find two terminal attributes: PN (start printing) and PS (stop printing). These are escape sequences that must be sent to the terminal to control local printing. Very few terminals have these attributes defined in their **termcap** entries. Use a text editor (such as **vi(C)**) to examine the */etc/termcap* file. (*/etc/termcap* can also be an alternate file, as defined by the TERMCAP variable.) Search for the entry for your terminal. For example, if your terminal is a Wyse 60, you would search for "wyse60".

The **termcap** entry for the Wyse 60 appears in Figure 15-2.

```
w7|wy60|wyse60|Wyse WY-60 with 80 column/24 line screen in wy60 mode:\
:is=\E'\072\Ee(\EO\Ee6\Ec41\E^4\Ec21\Ed/:\
:if=/usr/lib/tabset/std:pt:\
:G1=\EH3:G2=\EH2:G3=\EH1:G4=\EH5:GD=\EH0:GG#0:GH=\EH\072:\
:GU=\EH=:GV=\EH6:GR=\EH4:GL=\EH9:GC=\EH8:GF=\EH7:\
:PU=\EJ:PD=\EK:\
:al=\EE:am:bs:bt=\EI:cd=\EY:ce=\ET:c1=\E+:\
:cm=\Ea%i%dR%dC:co#80:dc=\EW:d1=\ER:ei=\Er:im=\Eq:k0=^AI\r:\
:k1=^A@\r:k2=^AA\r:k3=^AB\r:k4=^AC\r:k5=^AD\r:k6=^AE\r:k7=^AF\r:\
:k8=^AG\r:k9=^AH\r:kd=^J:kh=^_:kl=^H:kr=^L:ku=^K:\
:li#24:mi:nd=^L:se=\EG0:so=\EG4:sg#0:ug#0:ue=\EG0:ul:up=^K:us=\EG8:\
:PN=\Ed#:PS=\T:hs:ts=\Ez(:fs=\M:
```

Figure 15-2 Wyse 60 termcap Entry

The Wyse 60 has PN and PS defined (shown in bold). With other terminals, you must add a line containing these two attributes to the */etc/termcap* entry for your terminal. The line you add has the form:

```
:PN=start sequence:PS=stop sequence:\
```

5. Refer to your terminal manual to find the sequence of control characters used to switch the auxiliary port on and off. This is sometimes referred to as “passthrough” or “transparent” mode. For an example of the sequence to enable auxiliary printing, the code to switch the port on for a Wyse 60 terminal is:

```
<Esc> d #
```

And the code to turn it off again is:

```
<Ctrl>t
```

Adding a Local Printer

6. These keystrokes must be translated into **termcap** format before inserting them into the **termcap** file. **termcap** uses the codes in Table 15.2 to represent keystrokes.

Table 15.2
termcap Keystroke Translations

Keystroke	termcap Sequence
ESCAPE	\E
CTRL- <i>x</i>	^ <i>x</i> (<i>x</i> is any character)
NEWLINE	\n
RETURN	\r
TAB	\t
BACKSPACE	\b
FORMFEED	\f

To use a control sequence, use the caret (^) symbol, not the <Ctrl> key. For example, <Ctrl>x would be represented by ^x. In addition, characters can be represented by their octal codes (see **ascii(M)**, and the caret and backslash characters represented by \^ and \\, respectively). Entries for **termcap** attributes must be separated by a colon (:). (See **termcap(M)** for more details.)

Recall that the **termcap** attributes for starting and ending printing are PN and PS. Using the table above the **termcap** entry for the Wyse 60 keystrokes <Esc> # d (start printing, PN) and <Ctrl>t (stop printing, PS) looks like this:

```
:PN=\Ed#:PS=^T:\
```

7. For a terminal missing these entries, you simply insert a modified version of the above line into the **termcap** entry for the terminal. (You must be certain to insert the line within the entry for your terminal; do not add it as the first line or the last line.) For other terminals, check your owner's manual and locate the proper sequences for turning the auxiliary print mode on and off and substitute the **termcap** sequences as in the example.

Note

You must be logged in as *root* to edit */etc/termcap*. We recommend that you copy the original file to another name in case you make an error. You can also extract the file again from your distribution using **custom(ADM)**.

8. Once you have added the PN and PS entries, log out and back in again to activate the new termcap entry.
9. Use the following command to print the file *filename* on your local printer:

lprint *filename*

Do not touch your keyboard while local printing is taking place; you cannot perform other tasks on your terminal while printing.

10. If your file is printed on the screen instead of the printer, the PS and PN entries you created are incorrect. Revise the entries with the correct codes. If the file still does not print on the printer or the terminal, try crossing the Transmit and Receive Data pins in the cable connecting the terminal AUX port and the printer. (This is also known as a “null modem” connection.)

Print Service Command Summary

The print service has five regular user commands, which are shown in Table 15.3.

Table 15.3
User Commands for the Print Service

Command	Description
cancel(C)	Cancels a request for a file to be printed.
lp(C)	Sends a file or files to a printer.
lpstat(C)	Reports the status of the print service.
disable(C)	Deactivates the named printer(s).
enable(C)	Activates the named printer(s).

A separate set of commands available for the administrator is shown in Table 15.4. These commands are found in the */usr/lib* directory. If you expect to use them frequently, you might find it convenient to include that directory in your PATH variable. To use the administrative commands, you must be logged in as *root*.

Table 15.4
Administrative Commands for the Print Service

Command	Description
/usr/lib/accept	Permits job requests to be queued for a specified destination.
/usr/lib/reject	Prevents jobs from being queued for a specified destination. Described on the same manual page as accept(C) .
/usr/lib/lpadmin	Sets up or changes printer configurations.
/usr/lib/lpmove	Moves output requests from one destination to another. Described on the same manual page as lpsched(ADM) .

(Continued on next page.)

Table 15.4
Administrative Commands for the Print Service (*Continued*)

Command	Description
/usr/lib/lpsched	Starts the print service.
/usr/lib/lpshut	Stops the print service. Described on the same manual page as lpsched(ADM) .

Starting and Stopping the Print Service

Under normal operation, you should never have to start or stop the print service manually. It automatically starts each time the system goes into multiuser mode. Sometimes it is necessary to stop **lpsched**, especially if you want to reconfigure printers or if you want to add new printers using the **lpadmin** command (**mkdev lp** automatically stops and restarts **lpsched**).

Each time **lpsched** sends a print request to an interface program, it records an entry in a log file, */usr/spool/lp/log*. The entry includes the user name, the request ID, the name of the printer the request will be printed on, and the date and time requested. **lpsched** also records any error messages in this file. After you have stopped **lpsched**, the log file is renamed */usr/spool/lp/oldlog* and **lpsched** starts a new log file. Requests waiting to be printed before **lpsched** was stopped may have an entry in both log files.

Note

To start and stop the print service manually, you must be logged in as the super-user (*root*).

For more information on **lpsched**, see **lpsched(ADM)** in the *XENIX Reference*.

Manually Stopping the Print Service

To stop the print service manually, enter the following command:

```
/usr/lib/lpshut
```

The message is displayed:

```
scheduler stopped
```

Starting and Stopping the Print Service

All printing ceases within a few seconds. If you try to stop the print service when it is not running, you see:

```
Print services already stopped
```

Note

Jobs can appear to pass through a printer that is not online. If a printer is not online or operating properly, you should disable the printer.

Manually Starting the Print Service

To restart the print service manually, enter the following command:

```
/usr/lib/lpsched
```

It may take a minute or two for the printer configurations to be re-established before any saved print requests start printing.

15

Managing the Print Service

This section explains how to do the routine tasks associated with maintaining the print service.

Canceling a Print Request

To cancel a printout you have requested, use the **cancel(C)** command. When you request a printout, the system displays a request ID for your job. For example, if you send a job to a printer named "laser," the XENIX system displays the request ID as:

```
request id is laser-number
```

where *number* is the number assigned to your job. To cancel the job before it begins printing, use the following command:

```
cancel laser-number
```

The printout is canceled. Most systems print quickly, so a **cancel** command must be used promptly to have any effect.

Enabling and Disabling Printers

The **enable** command allows **lpsched** to print files on printers. A printer can accept requests for printing after the **accept** command is given for it, but to print the files, the **enable** command must be given as well.

For example, to enable a printer named "laser," enter:

```
enable laser
```

You can disable printers with the **disable** command. The scheduler, **lpsched**, does not send printing requests to disabled printers regardless of their status with respect to the **accept** command. The **-r** option of the **disable** command allows you to send a message to users explaining why a printer was disabled.

For example, to disable a printer named “laser” because of a paper jam, enter:

```
disable -r"paper jam" laser
```

Users requesting the status of “laser” with the command **lpstat -plaser** receive the following message:

```
printer laser disabled since Dec 5 10:15
paper jam
```

For more information on these two commands, see the **enable(C)** and **disable(C)** manual pages.

Adding a Printer to a Class

It is occasionally convenient to treat a collection of printers as a single class. This way a user can send a job and the print service picks the first printer in the class that it finds free. This allows faster turnaround, as printers are kept as busy as possible.

Classes are not needed if the only purpose is to allow a user to submit a print request by type of printer.

One use of classes is to put into a class a series of printers that should be used in a particular order. If you have a high-speed printer and a low-speed printer, for instance, you probably want the high-speed printer to handle as many print requests as possible, with the low-speed printer reserved for use when the other is busy. Because the print service always checks for an available printer in the order that the printers were added to a class, you could add the high-speed printer to the class before the low-speed printer and let the print service route print requests in the order you wanted.

Add a printer to a class by stopping the scheduler with **lpshut** and entering the following command:

```
/usr/lib/lpadmin -pprintername -cclassname
```

If the class *classname* does not exist yet, it is created. (Use **lpstat -c** to get a list of all classes and their printers.)

Note

Class names and printer names must be unique. This allows a user to specify the destination for a print request without having to know whether it is a class of printers or a single printer. Thus, you can not have a class and printer with the same name.

In addition to using the **lpadmin**(ADM) option described in this section, **mkdev lp** offers the same function via option 3: "Reconfigure an existing printer."

Until you add a printer to a class, it does not belong to any.

Setting the System Default Destination

You can define the printer or class used to print a file when the user has not explicitly asked for a particular destination and has not set the LPDEST shell variable. The printer or class must already exist first. Make a printer or class the default destination by entering the following command:

```
/usr/lib/lpadmin -dprintername or classname
```

If you later decide that there should be no default destination, enter a null *printername* or *classname* as in the following command:

```
/usr/lib/lpadmin -d
```

If you do not set a default destination, there is none. Users must explicitly name a printer or class in each print request, or they have to set the LPDEST shell variable with the name of a destination.

For C-shell:

```
setenv LPDEST printer
```

For Bourne or Korn shells:

```
LPDEST=printer; export LPDEST
```

Users can also place these commands in their *.login* and *.profile* files, respectively.

Displaying Printer Status

Use the **lpstat** command to examine the current status of a printer. A short form of this command gives just the status; you can use it to see if the printer exists and if it is busy, idle, or disabled. A long form of the command adds the complete configuration. Enter one of the following commands to examine a printer:

```
lpstat -p printername
lpstat -t
```

The second command is the complete status of all printers. With either command you should see something like the following:

```
device for printer-name: /dev/tty $n$ 
printer-name accepting requests since date.

printer printer-name is idle. enabled since date.

printer-name now printing request-id.
```

lpstat -r prints the current status of the scheduler, **lpsched**. See **lpstat(C)** for a complete list of options.

15

Removing a Printer or Class

You can remove a printer or class if it has no pending print requests. If there are pending requests, you have to first move them to another printer or class using the **lpmove** command, or remove them using the **cancel** command.

Removing the last remaining printer of a class automatically removes the class as well. However, removing a class does not remove the printers associated with that class. If the printer or class removed is also the system default destination, the system no longer has a default destination.

To remove a printer or class, enter the following command:

```
/usr/lib/lpadmin -xprintername or classname
```

Managing the Print Service

If all you want to do is remove a printer from a class but not delete the printer, enter the following command:

```
/usr/lib/lpadmin -pprintername -r classname
```

Note

In addition to using the **lpadmin**(ADM) option described in this section, **mkdev lp** offers the same function via option 2: "Remove a printer."

Managing the Printing Load

Occasionally, you may need to stop accepting print requests for a printer or move print requests from one printer to another. There are various reasons for doing this, such as the following:

- The printer needs periodic maintenance.
- The printer is broken.
- The printer was removed.
- The configuration was changed so that the printer can be used differently.
- Too many large print requests are queued for one printer and should be evenly distributed.

If you are going to make a big change in the way a printer is used, such as reconfiguration, print requests that are currently queued for printing on it must be moved or canceled. The print service attempts to find alternate printers, but only if the user does not care which printer is used. Such requests are not automatically moved; if you do not move them first, the print service cancels them.

If you decide that a printer is to be taken out of service, its configuration is to be changed, or it is too heavily loaded, you can move print requests from it and reject additional requests for it. Use the **lpmove** and **reject** commands for this. If you do reject requests for a printer, you can later accept requests using the **accept** command.

Rejecting Requests for a Printer or Class

To stop accepting any new requests for a printer or a class of printers, enter the following command:

```
/usr/lib/reject -r"reason" printername or classname
```

You can reject requests for several printers or classes in one command by listing their names on the same line, separating the names with spaces. The *reason* is displayed whenever anyone tries to print a file on the printer. You can omit the **-r** if you do not want to give a reason.

Managing the Printing Load

Although the **reject** command stops any new print requests from being accepted, it does not move or cancel any requests currently queued for the printer. These continue to print as long as the printer is enabled.

Accepting Requests for a Printer or Class

The **accept** command allows printers or classes of printers to accept print requests made with the **lp** command. You can allow a printer to accept requests after it has been properly configured.

After the condition that led to denying requests is corrected or changed, enter the following command to start accepting new requests:

```
/usr/lib/accept printername or classname
```

Again, you can accept requests for several printers or classes in one command by listing their names on the same line. You will always have to use the **accept** command for a new printer or class after you have added it because the print service does not initially accept requests for new printers or classes.

Moving Requests to Another Printer

If you have to move requests from one printer or class to another, enter one of the following commands:

```
/usr/lib/lpmove request-id printername  
/usr/lib/lpmove printername1 printername2
```

You can give more than one request ID before the printer name in the first command. The first command moves the listed requests to the named printer. The latter command moves all requests currently queued for the first printer to the second printer. When the latter command is used, the print service also no longer accepts requests for the first printer (this has the same effect as the **reject** command).

Note

The **lpmove** command requires the scheduler to be shut down.

Examples

This section includes examples of how to use **lpmove**, **accept**, and **reject**.

Example 1: Moving Requests for Maintenance

You decide to change the ribbon on printer **printer1** and perform some preventive maintenance. You want to move all the requests for printer **printer1** to printer **printer2**. After the requests are moved, the print service no longer accepts requests for **printer1** (this has the same effect as a **reject printer1** command issued after the **lpmove** command).

```
/usr/lib/lpmove printer1 printer2
```

Now you can disable the printer and start working on it. When you are finished, you can bring it back into service:

```
/usr/lib/accept lp1
```

At this point, if you had disabled the printer you should re-enable it. See the “Enabling and Disabling Printers” section in this chapter.

Example 2: Temporary Load Shifting

You notice that someone has queued several large files for printing on the printer **laser1**. Meanwhile **laser2** is currently idle because no one had queued requests for it. You'll move the two biggest requests, **laser1-23** and **laser1-46** to **laser2**, and you reject any new requests for **laser1** for the time being.

```
/usr/lib/lpmove laser1-23 laser1-46 laser2  
/usr/lib/reject -r "too busy--will reopen later" laser1
```

Managing the Printing Load

Example 3: Preventing Requests for Maintenance

You want to prevent printing requests from being routed to printer4 because of repairs:

```
/usr/lib/reject -r"printer4 needs repair" printer4
```

A user who requests a file to be printed on printer4 receives the following message:

```
Requests for destination "printer 4" aren't being accepted
```

To find out the acceptance status of printing destinations, enter:

```
lpstat -a
```

The output looks like this:

```
printername accepting requests since date time  
printer4 not accepting requests since date time  
printer4 needs repair
```

Using Interface Programs

Interface programs (also known as interface scripts) are used to initialize printers and take advantage of their individual capabilities. These programs are human-readable files located in */usr/spool/lp/model*. The *dumb* interface program defines minimal capabilities common to most printers. The interface programs already present on your system are listed in Table 15.5. You can select an interface program when you install or change a printer configuration using the **mkdev lp** command or the **sysadmsh** System→Add→Printer selection.

Table 15.5
Printer Interface Programs

Name	Description
<i>TandyDMP</i>	Tandy DMP Printers
<i>crnlmap</i>	serial or parallel printer requiring newline mapped to carriage return plus linefeed
<i>1640</i>	DASI 1640 terminal
<i>dumb</i>	dumb line printer
<i>emulator</i>	Tandy Printers in IBM Emulation Mode
<i>epson</i>	Epson serial or parallel printer
<i>f450</i>	DASI 450 terminal
<i>hp</i>	hp2631a line printer
<i>hpjet</i>	HP Laserjet, Thinkjet or QuietJet
<i>network</i>	remote printing over UUCP, Micnet or ethernet
<i>ph.daps</i>	Autologic APS-5 phototypesetter
<i>postscript</i>	postscript printer
<i>pprx</i>	Printronix line printer (parallel)
<i>prx</i>	Printronix line printer (serial)
<i>proprinter</i>	IBM Proprinter XL

15

Many of these interface programs have special options available by using the **-o** option to the **lp** command. (Read the actual interface file for this information.) For example, the *postscript* interface program includes the options listed in Table 15.6.

Table 15.6
postscript Options

Option	Description
port	prints text in portrait mode
land	prints text in landscape mode
land2	prints text in 2 page landscape mode
raw	prints a postscript file

You can also create your own interface scripts or customize existing ones to suit your needs. See “How to Write an Interface Program” later in this chapter for details.

How to Write an Interface Program

If you have printing needs that are not supported by the standard interface program, you can furnish your own interface program. It is a good idea to start with the *dumb* interface program (or one of the many others found in */usr/spool/lp/model*) and change it to fit, rather than starting from scratch. The *dumb* script is found under the name */usr/spool/lp/model/dumb*.

What Does an Interface Program Do?

Any interface program performs the following tasks:

- Initializes the printer port, if needed. The generic interface program uses the **stty** command to do this.
- Initializes the physical printer.
- Prints a banner page, if needed.
- Prints the correct number of copies of the request content.

How an Interface Program is Used

When the print service routes an output request to a printer, the interface program for the printer is invoked as follows:

```
/usr/spool/lp/interface/printer id user title copies options files
```

Arguments for the interface program are:

<i>printer</i>	Printer name. This is the name given to the interface program itself.
<i>id</i>	Request id returned by lp .
<i>user</i>	Login name of user who made the request.
<i>title</i>	Optional title specified by the user.
<i>copies</i>	Number of copies requested by user.
<i>options</i>	List of options separated by blanks, specified by user or set by the print service.
<i>files</i>	Full pathname of the file(s) to be printed.

When the interface program is invoked, its standard input comes from */dev/null*, its standard output is directed to the printer port, and its standard error output is directed to a file that will be given to the user who submitted the print request.

The file */etc/default/lpd* contains a line “BANNERS=*d*” where *d* is the number of banner pages to be printed at the front of every printing request. If *d* is set to 0, no banner pages are printed. Interface programs should examine */etc/default/lpd* and behave accordingly.

15

Customizing the Interface Program

Make sure that the custom interface program sets the proper **stty** modes (terminal characteristics such as baud rate or output options). The *dumb* interface program does this, and you can follow suit. Look for the section that begins with the shell comment:

```
# If it is necessary to change the baud rate or other stty settings for
# your serial printer change the following line
```

Follow the code used in the *dumb* interface program. It sets both the default modes and the adjusted modes given by the print service or the user with a line like the following:

```
stty modes 0<&1
```

This command line takes the standard input for the **stty** command from the printer port. An example of an **stty** command line that sets the baud rate at 1200 and sets some of the option modes is shown here:

```
stty -parenb -parodd 1200 cs8 cread clocal ixon 0<&1
```

Using Interface Programs

Devices are opened for reading as well as writing when file modes permit. If a device is a regular file, all output is appended to the end of that file.

Interface programs can format their output in any way. They must, however, ensure proper **stty** modes for terminal characteristics such as baud rate and output options. In a shell script interface, this means the printer device must be open for reading - take the standard input for the **stty** command from the device.

When printing is complete, your interface program should exit with a code that tells the status of the print job. Exit codes are interpreted by the print service as shown in Table 15.7.

Table 15.7
Exit Codes

Code	Meaning to the Print Service
0	The print request completed successfully. If a printer fault occurred, it was cleared.
1 to 127	A problem was encountered in printing this particular request (for example, too many non-printable characters or the request exceeds the printer capabilities). This problem does not affect future print requests. The print service notifies the person who submitted the request via mail(C) that there was an error in printing it. If a printer fault occurred, it was cleared.
> 127	These codes are reserved for internal use by the print service. Interface programs must not exit with codes in this range.

Finally, when problems occur in printing that are likely to affect future printing requests, the printer interface program should disable printers so that print requests are not lost. When a busy printer is disabled, the interface program will be terminated with a signal 15 so that print requests are not lost. (See **signal(S)** in the *XENIX Programmer's Reference*, and the **kill(C)** manual page).

For more information on printer interface programs, see **lpadmin(ADM)**.

Adjusting the Printer Port Characteristics

You should make sure that the printer port characteristics set by the print service match the printer communication settings. The standard printer port settings were designed to work with typical XENIX files and many printers, but they do not work with all files and printers. This is not really a customizing step, because a standard feature of the print service is to allow you to specify the port settings for each printer. However, it is an important step in getting your printer to work with the print service, so it is described in more detail here.

When you add a new printer, read the documentation that comes with it so that you understand what it expects from the host (the print service). Then read the manual page for the `stty(C)` command. It summarizes the various characteristics that can be set on a terminal or printer port.

Only some of the characteristics listed in the `stty(C)` manual page are important for printers. The ones likely to be of interest to you are listed in the following table (but you should still consult the `stty(C)` manual page for others). The most commonly-used options are listed in Table 15.8.

Table 15.8
Commonly-used stty Options

15

stty Option	Meaning
evenp	Sends even parity in the 8th bit.
oddp	Sends odd parity in the 8th bit.
-parity	Does not generate parity, sends all 8 bits unchanged.
110 - 9600	Sets the communications speed to this baud rate.
ixon	Enables XON/XOFF (also known as START/STOP or DC1/DC3) flow control.
-ixon	Turns off XON/XOFF flow control.

(Continued on next page.)

Table 15.8
Commonly-used stty Options (*Continued*)

stty Option	Meaning
-opost	Does not do any output post-processing.
opost	Does output post-processing (such as onlcr , ocrnl , and tabs).
onlcr	Sends a carriage return before every linefeed.
-onlcr	Does not send a carriage return before every linefeed.
ocrnl	Changes carriage returns into linefeeds.
-ocrnl	Does not change carriage returns into linefeeds.
-tabs	Changes tabs into an equivalent number of spaces.
tabs	Does not change tabs into spaces.
ixany	When XON/XOFF flow control is enabled, allows any character (even XOFF) to restart communications.
-ixany	When XON/XOFF flow control is enabled, allows only XON to restart communications (recommended for printers).

Printers connected directly to computers and those connected over some networks require that the printer port characteristics be set by the interface program. These characteristics define the low-level communications with the printer. Included are the baud rate; use of XON/XOFF flow control; 7, 8, or other bits per byte; style of parity; and output post-processing. The standard interface program uses the **stty** command to initialize the printer port, minimally setting the baud rate and a few other default characteristics.

If you have a printer that requires printer port characteristics other than those handled by the **stty** program, you must customize the interface program.

How to Add an Interface Program

If you do not choose an interface program, the standard one provided with the print service is used. This should be sufficient for most of your printing needs. If you prefer, however, you can change it to suit your needs or completely rewrite your own interface program, and then specify it when you add a new printer.

To identify a customized interface program by name, give the printer name and the pathname of the interface program as follows:

```
/usr/lib/lpadmin -pprintername -i pathname
```

To identify a customized interface program by reference to another printer, give the printer names as follows:

```
/usr/lib/lpadmin -pprintername1 -e printername2
```

printername₁ should be replaced with the name of the printer you are adding; *printername₂* should be replaced with the name of the printer already added that is using the customized interface program.

To identify an interface program by reference to a model interface program, give the printer name and model name as follows:

```
/usr/lib/lpadmin -pprintername -mmodelname
```

This takes the interface program from the file *usr/spool/lp/model/modelname*.

Note

If *printername* is a new printer to be added, the **lpadmin** command line must include the **-v** (device) option. The printer must then be enabled and accepted.

In addition to using the **lpadmin**(ADM) options described in this section, **mkdev lp** offers the same functions via option 3: "Reconfigure an existing printer."

Setting Up RTS/CTS Protocol Serial Printers

The RTS and CTS lines for the RS-232 serial interface were originally intended as handshaking signals between a Data Terminal Equipment (DTE) device (computer, printer, and so forth) and a Data Communications Equipment (DCE) device (almost always a modem). The RTS (Ready To Send) line is asserted by the DTE when it is ready to send data to the DCE. The DCE asserts the CTS (Clear To Send) line when it was ready to receive data. If the CTS line goes low, then the DTE should stop sending data until CTS goes high again.

The XENIX system also uses the RTS line for handshaking in the other direction. If the printer sees that its input buffer is nearly full, it will lower the CTS line. The serial driver will then stop sending, and wait for the printer to catch up. The operating system will raise the CTS line when it is ready for more data.

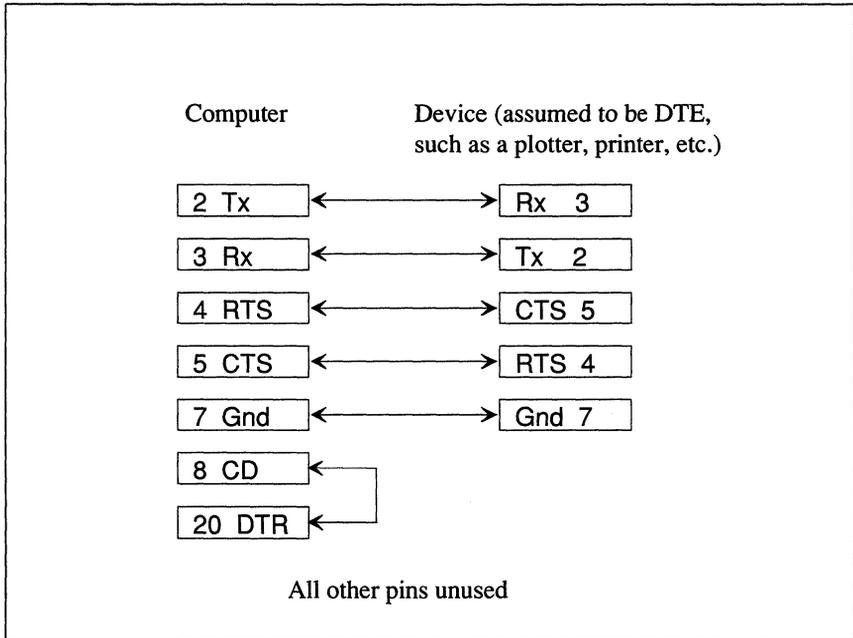
Many printers use the DTR (Data Terminal Ready) line for handshaking rather than RTS or CTS. For these devices, the cable must be wired to connect the printer's DTR pin to the computer's CTS pin (see Figure 15-4).

To set up for RTS/CTS flow control, do the following:

1. Use the modem-control port (for example: `/dev/tty1A`). If you plan to use the spooler to access this printer, make sure you specify the modem control port rather than one of the standard serial devices displayed when you use the `sysadmsh` System→Add→Printer selection asks you to enter a device name.
2. Make sure the `stty` settings in the interface script include **-ixon -ixoff -clocal rtsflow ctsflow**.

Setting Up RTS/CTS Protocol Serial Printers

- For a device that uses the RTS and/or CTS lines for handshaking, the cable should be wired as shown in Figure 15-3.



15

Figure 15-3 RTS/CTS Handshaking

Setting Up RTS/CTS Protocol Serial Printers

4. If the device uses the DTR line for handshaking, the cabling should be as shown in Figure 15-4.

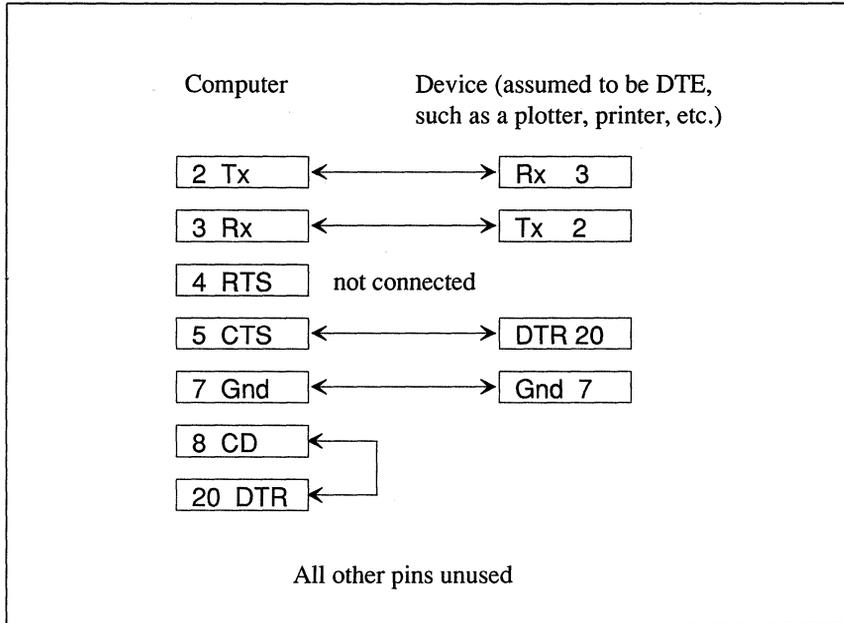


Figure 15-4 DTR Handshaking

5. If the information contained here does not solve the problem, try removing **rtsflow** from the **stty** command string.

Using a Printer Without the Spooler

If you use a printer without the spooler, any **stty** settings you have specified for use with that printer do not stay in effect. The spooler opens the file and then runs the **stty** commands as specified in the printer interface script. To use a printer without the spooler, follow the instructions in this section.

While logged in as root, give the following commands or place them in the initialization file in */etc/rc.d/7/uudaemons* before the line that calls */usr/lib/lpsched*. The first command is for serial printers, and the second for parallel printers:

```
(stty baud ixon ixoff -ixany ; cat > /dev/null) < /dev/ttyn &
```

```
(stty onlcr opost; while : ; do sleep 3600; done) < /dev/lpn &
```

where *baud* is the baud rate of the printer, and *tty*n or *lp*n is the serial or parallel device name. This command sets the **stty** options and holds the port open for use without the spooler.

15

Note

If you ever need to enable the port, make sure you kill this process first.

This command does not work from a C-shell (*csh*). It returns the message:

```
stty: no such device
while: expression syntax
```

In addition, with certain multiport cards, it is necessary to add a **sleep** command after the initialization program supplied with the card, *initprogram*, followed by the **stty** holdopen command:

```
initprogram &
sleep 3
```

Creating an Init Device File

The standard parallel printer devices (*/dev/lp*, */dev/lp0*, */dev/lp1*, and */dev/lp2*) send a printer initialization string (*init*) the first time the device is opened after the system starts up. This is done on the first open only, so printers with large text buffers are not flushed by the sending of another file.

Some parallel printers require initialization every time a file is received for printing. Others require an *init* if the printer is turned off and back on again (for example, after changing paper or ribbons). The symptom of this situation is that the printer works fine until it is turned off and then back on.

If you need to initialize the printer more often than the standard devices provide, you can create an additional device file for the parallel port in use. This init device file can be used when necessary to initialize the printer.

1. Log in as super user.
2. Determine which device is the parallel port you are using. This example assumes the device is the main parallel port (*/dev/lp0*).
3. Invoke **mkdev lp** (or the **sysadmsh** System→Add→Printer selection) to associate one of the parallel *init* devices (*lp0i*, *lp1i*, *lp2i*) with the printer. Select option 3: "Reconfiguring an existing printer."

If your printer needs an *init* when it is turned off and on, use the following command line after the printer is turned on. Initialize the printer before the first file is sent to the printer (this example assumes the main parallel port):

```
>/dev/lp0i
```

If your printer needs an *init* every time a file is sent (and it does not have a large internal text buffer), you can use the */dev/lp0i* device all the time.

The **lp(C)** command will then send an *init* every time a file is sent to the printer.

Chapter 16

Using Floppy Disks and Tape Drives

Introduction 16-1

Using Cartridge Tape Drives 16-2

Installation and Configuration 16-2

Cartridge Tape 16-3

Mini-Cartridge 16-6

Irwin 16-6

QIC-40/80 16-7

SCSI 16-8

Editing /etc/default/tar 16-9

Archiving Files on Tape 16-11

The tar Command 16-11

Tape Drive Maintenance 16-12

Tape Formatting 16-13

Tape Driver Error Correction Code (ECC) Support 16-14

Using Floppy Disks 16-15

Formatting Floppy Disks 16-15

Copying Floppy Disks 16-16

Using diskcp 16-17

Using dd(C) 16-18

Using Floppies for File Storage 16-18

Making Filesystems on Floppy Disks 16-19

Mounting a Floppy Filesystem 16-21

Introduction

An important part of any computer system is the ability to offload files and restore them when needed. There are several types of media used to store and recall files. Among these are floppy disks and magnetic tape devices. This chapter explains how to install and use storage media with your system. Your system should come with at least a floppy disk drive already installed and ready to run. This chapter provides instructions on how to add tape drives and how to use floppy disks.

Note

Certain tape drives are not supported on XENIX-286 distributions. Check your *Release Notes* for details.

Using Cartridge Tape Drives

A tape cartridge drive is a mass storage device that uses 1/4-inch tape cartridges to store data. A tape cartridge can hold many times the data that can be stored on floppies, making it much more useful for large backup operations.

The drives that are supported are listed in the *Release Notes*. For hardware-specific information, refer to the manual for your drive and the `tape(HW)` manual page.

Installation and Configuration

Read your tape drive hardware manual for physical installation instructions and general information.

To add a tape drive, log in as *root* and enter the following:

```
mkdev tape
```

The following menu is displayed:

```
Xenix System V Tape Driver Installation
```

1. Install a Tape Driver
2. Remove a Tape Driver

```
Select an option or enter q to quit:
```

Enter **1** to add the drive. You are then asked to select the type of tape drive you have installed:

1. Install Cartridge Tape Driver
2. Install Mini-Cartridge Tape Driver
3. Install Qic-40/Qic-80 Tape Driver
4. Install SCSI Tape Drive

```
Select an option or enter q to return to the main menu:
```

The subsections that follow describe the configuration requirements for each drive type. Be sure and consult the sections on “Kernel Relinking” and “Boot Messages” after following the instructions for your drive type.

Note

Where it is indicated to set jumpers or switches on your tape controller card, it is primarily referring to an AT type machine. If you are installing on a Micro Channel machine, there will be nothing to physically set. You must change settings using a Reference (setup) diskette provided with your machine.

Cartridge Tape

The Cartridge Tape Driver selection refers to the QIC-02-type full-size cartridge tape drives. When you select the Cartridge Tape, another menu is displayed that controls the tape parameters:

- 1) Display current tape parameters
- 2) Modify current tape parameters
- 3) Select previous tape parameters
- 4) Select default tape parameters

16

The system has default values for each supported drive; these values are documented in Appendix A of the *Release Notes*. You need not modify these defaults unless you have configured your tape drive differently. Enter **q** and press <Return> if you are using the default values.

Using Cartridge Tape Drives

If you have a non-default configuration, you must enter **2** to modify the system parameters. You see a display similar to the following:

<u>Tape Parameters</u>	<u>Values</u>	<u>Comments</u>
1. Controller Type	0	1 = type A, 3 = type W, 4 = type E 5 = type M, 6 = type T or 7 = type X
2. DMA Channel	0	1 or 3
3. Interrupt Vector	0	logical vector number
4. Base Address	0H	i/o addresses start here

Enter a parameter to modify or 'q' to return to the main menu.

Enter a value for each of the four categories listed above. Included in this section are some basic explanations of each category.

Controller Type

This is the kind of tape drive/controller you have installed. The values, type, and manufacturer are shown in Table 16.1.

Table 16.1
Tape Drive Types

Value	Type	Manufacturer
1	A	Archive
2	W	Wangtek
3	E	Emerald
4	M	Mountain
5	T	Tecmar
6	X	Everex

DMA Channel

This is the Direct Memory Access (DMA) Channel that is being used. The choices are DMA Channel 1 or DMA Channel 3. On a typical tape controller card, you will have a set of jumper pins for DACK and another set for DRQ. Each of these should have three pairs of pins labeled 1, 2, and 3. The DACK and DRQ pins should each be jumpered for the appropriate value (1 or 3) as they combine to give you your DMA value. In other words, if you want a DMA value of 3, you should jumper DACK for 3 and DRQ for 3. Either DMA 1 or 3 are acceptable and neither provides any particular advantage over the other.

Interrupt Vector

This is the interrupt value being used. On a tape controller card, there will be an IRQ set of jumper pins, typically labeled 2-7. You can use any interrupt you choose, as long as it is not being used by another device on your system. Standard interrupts are shown in Table 16.2.

Table 16.2
Typical Device Interrupts

Interrupt	Device
0*	Clock
1*	Console
2	Networks, Tapes and others
3	Serial COM2
4	Serial COM1
5	Alternate Parallel Port (lp2)
6*	Floppy Disk
7	Main Parallel Port (lp0 or lp1)

Never use interrupts 0,1, or 6, as they are always used for standard devices on your system. The rest are potentially available, unless devices that typically use them are installed on your system. For example, if you have a COM1 serial port on your system and it is recognized on the XENIX boot screen, then interrupt 4 is not an available interrupt, unless you disable your COM1 port. The rule of thumb is that if a device is recognized by XENIX on the boot screen, the interrupt it uses is taken even if you are not actively using the device. In this case, you must either use another free interrupt, or physically disable the device which is using the interrupt you want to use for the tape controller.

The most commonly available interrupts are 2 and 5 as most systems do not come standard with an LP2 port or a network card. Be aware that if you do choose to use interrupt 2, you need to indicate a value of 25 when asked for the "Interrupt Vector" in the **mkdev tape** program. This is the only case where a value other than the actual value is given. This is because interrupt 2 is actually connected to a slave interrupt chip which allow AT machines to support more devices. The software interrupt 25 maps through the slave chip to interrupt 2 on the master.

When you determine what interrupt you will be using, jumper the appropriate IRQ pins on your tape controller card and then indicate the same value (except in the case of interrupt 2 where you should indicate 25), in the **mkdev tape** program.

Using Cartridge Tape Drives

Base Address

This is the I/O address that your tape controller card is configured for. A typical tape controller card will have either jumper pins or switches for configuring the address. Check your tape hardware documentation to determine what the available address settings are. Again, XENIX does not really care what address you are using, as long as it is not being used by another device on your system. This is not usually an issue, as common tape controller addresses are not often used by other devices.

After setting an address (or finding out what has already been set) on your tape controller card indicate the same address in the **mkdev tape** program. You should always enter the address with a preceding "0" and a succeeding "H." For example, if your address is 300 (hexadecimal), you should indicate "0300H" in the **mkdev tape** program.

Mini-Cartridge

Mini-tape drives use the floppy disk drive controller and are significantly different from standard QIC tape drives. These units are the so-called "floppy" or "Irwin" tape units. They are not configurable and do not require parameters to be entered. In addition, mini-tapes must be formatted before they can be used (preformatted tapes are available and recommended). There are also some differences in the installation of mini-tapes.

First make sure that your drive is correctly jumpered. The correct setting may be different for different brands of machines. See your hardware documentation and your *Release Notes* for more information.

Irwin

The Irwin driver includes a configuration file */etc/default/mcconfig* that can contain options for debugging, hardware-specific options, and other useful features. The *mcconfig* file does not exist by default, it must be created and entries added. Table 16.3 contains some commonly used *mcconfig* entries.

Table 16.3
***/etc/default/mcconfig* Entries**

String	Meaning
<code>iropt=F</code>	Floating drive search (Adaptec controller)
<code>iropt=M</code>	Microchannel bus
<code>iropt=I</code>	Wait-for-index before transfer of each block
<code>4100=P:370,I:6,D:2,T:2,T:0</code>	Specifies 4100 PC Bus Controller parameters

Refer to the **mcconfig(F)** manual page for more information on these and other options.

In addition, the irwin driver includes a daemon process */etc/mcdaemon* that is automatically started at boot time. The irwin drive will not function in single-user mode unless the daemon is started. To start the daemon while in single-user mode, enter the following command:

```
/etc/mcdaemon
```

Note

XENIX-286 distributions include a different Irwin driver and the functionality of the *mconfig* file and the */etc/mcdaemon* is not included.

QIC-40/80

These units are specialized mini-cartridge units that use a different format from the mini-cartridge units described earlier. You are asked to select the drive type, QIC-40 or QIC-80. In addition, you must supply the position of the drive:

1. Available for floppy disk only
2. Drive is installed as Floppy Position 2
3. Drive is installed as Floppy Position 3
4. Drive is installed as Floppy Position 4

Refer to your tape drive documentation for instructions on how to jumper your drive properly. Choose the option that corresponds to the physical drive select jumper on your tape drive. For example, if you are configuring your tape drive as the second device on your floppy controller, you would need to put the physical drive select jumper on your tape drive on position 2 and then choose option 2 of the menu.

SCSI

SCSI tape drives are attached to a SCSI adapter. You are asked to provide the device ID number (0-7) and the number of the Host adapter the drive is attached to. The logical unit number of the device (LUN) is always 0 and you are not prompted for it. See Figure 16-1.

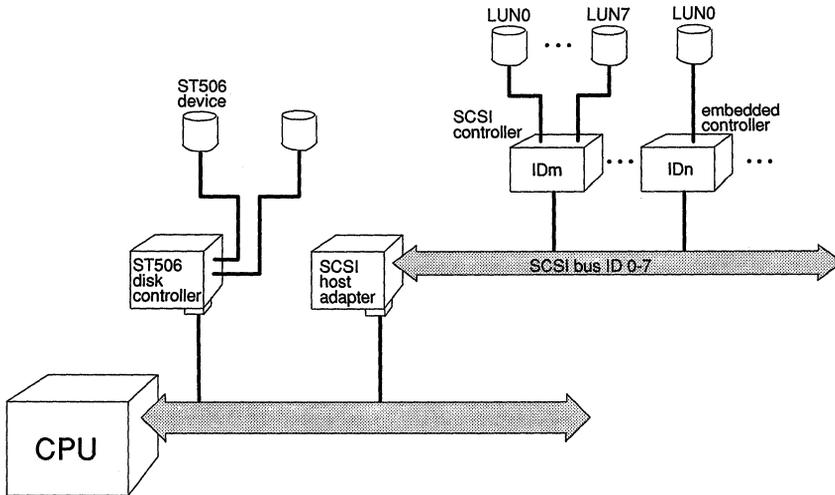


Figure 16-1 SCSI Configuration Schematic

As shown in Figure 16-1, a SCSI host adapter (HA) translates signals from the CPU bus to the SCSI bus. A SCSI controller is known by a SCSI ID. To configure a SCSI tape device, you must know the ID number of the controller (0-7) on the host adapter; the host adapter itself is usually ID 7, giving it the highest priority on the SCSI bus

Because the tape drive and its controller are one unit (referred to as “embedded”) the LUN or logical unit number is simply 0 and it is not necessary. The **mkdev tape** prompts for these values appear as follows:

```
What ID is this tape? Choose a number between 0 and 7:
```

```
Which SCSI host adapter will interface with this tape?
```

```
Enter either 1 (first adapter) or 2 (second adapter):
```

Kernel Relinking

After you select the driver to be installed and provide any additional information, you are prompted to permit relinking of the kernel. The tape drive unit is available for use after rebooting.

Boot Messages

When the kernel recognizes a tape drive (and when the driver is linked into the kernel) a message is always displayed at boot time indicating the device is present. This information can also be displayed using **hwconfig(C)**. Table 16.4 contains the messages displayed, indicating the drive type.

Table 16.4
Tape Drive Boot Messages

Type	Boot Display Message					
QIC-02*	%tape	0x0338-0x033C	05	1	type=W	
mini	%ctmini	-	-	-	type=ir	
QIC-40	%ctmini	-	-	-	type=qic40	
QIC-80	%ctmini	-	-	-	type=qic80	
SCSI	%tape	-	-	-	type=S ha=0 id=2 lun=0	

* The address, interrupt, and DMA shown for QIC-02 are examples.

Editing /etc/default/tar

After you install your tape drive, you must enter the correct size setting in the */etc/default/tar* file. When you edit the file, you see several entries for various default devices. Figure 16-2 shows the */etc/default/tar* file provided with the 96tpi distribution.

Using Cartridge Tape Drives

# device	block	size	tape
archive0=/dev/rfd048ds9	18	360	n
archive1=/dev/rfd148ds9	18	360	n
archive2=/dev/rfd096ds15	10	1200	n
archive3=/dev/rfd196ds15	10	1200	n
archive4=/dev/rfd096ds9	18	720	n
archive5=/dev/rfd196ds9	18	720	n
archive6=/dev/rfd0135ds18	18	1440	n
archive7=/dev/rfd1135ds18	18	1440	n
archive8=/dev/rct0	20	0	y
archive9=/dev/rctmini	20	0	y
# The default device...			
archive=/dev/rfd096ds15	10	1200	n

Figure 16-2 /etc/default/tar File

The utilities **backup**(ADM) and **restore**(ADM) have similar files and entries. For more information on default files, see the **default**(F) manual page and the manual entry for the particular backup or restore command.

QIC Cartridge Drives

The */dev/rct0* entry accesses the QIC cartridge tape drive. The cartridge sizes are indicated in Table 16.5. You can edit */etc/default/tar* and set the appropriate size, or create multiple entries to accommodate different tape sizes.

Table 16.5
QIC Cartridge Sizes

Length in Feet	Entry in "Size" Field
300	30000
450	45000
600	60000

Mini-Cartridge Drives

Find the entry in your */etc/default/tar* file for */dev/rctmini*. In the above sample file, this is **archive9**. Note that the size value for *rctmini* is 0. If you plan to use the default file, you must change this entry when you install your *rctmini* device. The correct number for your *rctmini* device varies with the size of the tape you use. See Table 16.6

Table 16.6
Mini-Cartridge Sizes

Tape Size	Actual Capacity	Entry in “Size” field
10 Mbyte	8 Mbyte	8000
20 Mbyte	17 Mbyte	17000
40 Mbyte	35 Mbyte	35000
80 Mbyte	72 Mbyte	72000

QIC-40/80 Mini-Cartridge Drives

If you wish to use the default file with a QIC-40 drive, you must make a new entry and add it yourself. Depending on the tape size, the QIC-40/80 entry should look like this:

```
archive9=/dev/rft0      20    0    y
```

Archiving Files on Tape

You use a tape drive much like a floppy, but the volume of data stored is much greater. Tapes are much better for storing (backing up) entire filesystems. The **tar(C)** command is the recommended archive program for users and is best used for general archiving or transporting of files. Other programs such as **backup(ADM)** and **restore(ADM)** are meant for system administrators making copies of entire filesystems. Consult “Backing Up Filesystems” in this guide for making regular backups of filesystems.

The **cpio(C)** command is a general-purpose archive program that uses a different format than **tar**. The **dd(C)** program transfers or converts archives of unusual format; the input and output format can be specified on the command line.

The tar Command

The **tar** command is useful for making a backup copy of entire directories. The command has the syntax:

```
tar cvf devicefile files
```

Using Cartridge Tape Drives

The *devicefile* is the filename that corresponds to the cartridge drive. *files* are the names of the files or directories to be copied. The */etc/default/tar* provides device abbreviations as shown in Figure 16-2 so you do not have to specify the device name. For example, to copy all the files in the directory */u/bogart* to the cartridge drive */dev/rct0*, enter:

```
tar cv8 /u/bogart
```

Δ **sysadmsh** users select: Media→Archive

To restore files stored on the same tape, insert the cartridge containing the files or directories you wish to restore and enter the following command:

```
tar xv8
```

Δ **sysadmsh** users select: Media→Extract

tar restores all the files on the tape to the original directory.

Tape Drive Maintenance

The **tape(C)** utility performs various tape maintenance operations on all tape drives. **tape** sends commands and receives status from the tape drive. The basic form of the command is:

```
tape command [ devicefile ]
```

For example, to rewind a cartridge tape device, enter:

```
tape rewind
```

Other commands include the following:

- erase** erases tape cartridge and re-tensions.
- reset** resets tape controller and tape drive, clears error conditions, and returns tape subsystem to power-up state.
- reten** re-tensions tape cartridge. Should be used periodically to remedy slack tape problems that generate an unusually large number of tape errors, or in extreme cases, actually tangle the tape in the drive.

There are also several commands that are specific to Irwin drives:

info	prints information about the inserted cartridge, including whether or not it is formatted, what type of format, and write protect status.
capacity or kapacity	displays the capacity of the inserted cartridge in 512- and 1024-byte blocks, respectively.

After certain tape operations are executed, the system returns a prompt before the tape controller has finished its operation. If you enter another tape command too quickly, the message “device busy” is displayed until the tape device is finished with its previous operation.

You should clean the tape drive heads and re-tension cartridges to keep it operating error-free.

Tape Formatting

Tape cartridges used with the mini-tape drive (ctmini) must be formatted before use. The following command formats a ctmini-tape cartridge:

tape format

Preformatted mini-cartridge tapes are available and recommended. See also **tape(HW)** and **tape(C)** for more information.

16

Note

Do not specify the raw device (example, */dev/rct0*) when using the **tape** command.

Tape Driver Error Correction Code (ECC) Support

Tape ECC is supported for QIC-02 only. The ECC tape device node, */dev/erct0*, is automatically created when you run **mkdev tape**. To use ECC, you must read and write from this device, not the normal */dev/rct0*. Users with tape drives that support cartridges larger than 60 Mbytes should consider editing the */etc/default/tar* file and substituting */dev/erct0* for their normal tape device.

The error recovery scheme is 2/64, which means that two 512-byte blocks out of every 64 blocks can go bad and the driver corrects them. The probability of error with ECC is $1:10^{14}$. Standard drives have an error probability of $1:10^9$.

Be sure and label tapes that are created with the ECC device; these tapes cannot be read by standard devices. In addition, if transporting data from one machine to another, it is advisable to use the ECC device only if the target machine supports the ECC scheme.

Using Floppy Disks

Floppy disks are the most convenient form of storage media. Depending on your floppy disk drive, you may be able to store from 360 Kbytes to 1.4 Mbytes on a single disk. Floppy disks can be used for simple data storage in **tar**, **cpio**, or **dd** formats or you can make a mountable filesystem on a floppy disk. The following sections explain how to use floppies for data storage and as extra filesystem space.

Formatting Floppy Disks

Floppy disks must be formatted before they can be used. The XENIX command to format a floppy disk is:

```
format /dev/floppy-device
```

△ **sysadmsh** users select: Media→Format

The floppy device you specify in the command relates to the type of disk drive and floppy you are using. For example, if you have a high-density 5.25-inch floppy disk drive, you can use it in high-density mode (96 tracks per inch) or in low-density mode (48 tpi). If you have high-density floppies to use with your drive, the floppy device to specify is:

```
/dev/rfd096
```

In this example, *rfd* indicates the raw floppy device, *0* indicates that this is the primary floppy drive, and *96* indicates high-density mode. Similarly, if you wish to use low-density floppies and the low-density mode of the floppy drive, the device name is:

```
/dev/rfd048
```

In this example, *48* indicates the low-density mode of floppy drive *0*.

Using Floppy Disks

Note

Only the raw device (for example: */dev/rfd096*) can be formatted. Attempting to format the block device (for example: */dev/fd096*) will result in an error.

/etc/default/format file

You can also define a default format device by adding an entry to the file */etc/default/format*. For example:

```
DEVICE=/dev/rfd096ds15
```

After adding the above line, you no longer have to specify the device name. In addition, it is possible to define that all floppies be verified, which confirms that the data on the floppy is readable. (This can also be specified on the command line with the *-y* option.) Automatic verification can be specified by the following entry:

```
VERIFY=Y
```

If this entry is placed in */etc/default/format*, all floppies formatted with the **format** command are verified. (To override verification, use the *-n* option on the command line.)

Refer to the **format(C)** manual page for more details.

Copying Floppy Disks

To ensure against the loss of data stored on floppy disks, any user can use the **diskcp(C)** command, or the **dd(C)** command to make copies of floppy disks on new, formatted disks.

diskcp makes use of **dd** and provides a simple interface to that program. **dd** is very powerful, and you can use it to perform many different kinds of copying.

You must copy information onto formatted disks. If you format floppies, you can use them over again without reformatting.

If you have disks that were formatted under another operating system, you must reformat them before you can use them to make copies of XENIX disks. Be aware that floppies formatted under some operating systems cannot be used under other operating systems, even with reformatting.

You can use the **format** command to format floppies. This command is described in the section “Formatting Floppy Disks” in this chapter. The **diskcp** command can also format floppies for you.

Using diskcp

To copy a floppy disk using **diskcp**, do the following:

△ **sysadmsh** users select: Media→Duplicate

1. Insert the disk you want to copy, known as the *source* floppy, in drive 0, your primary floppy drive.
2. Insert another floppy in the other drive. This floppy is known as the *target* disk. Note that any information already on the target disk is destroyed.

If you have only one disk drive, leave the source floppy in the drive. **diskcp** prompts you to remove the source disk at the correct time.

3. To format the floppy disk before the image is copied, enter the command:

```
diskcp -f
```

Press ⟨Return⟩.

If your computer has dual floppy drives, enter the following command to copy the image directly on the target floppy:

```
diskcp -d
```

Press ⟨Return⟩.

If you do not need to format the target floppy, simply enter:

```
diskcp
```

Press ⟨Return⟩.

Using Floppy Disks

4. Follow the instructions as they appear on your screen. Note that, with a single drive system, you are prompted to remove the source disk and insert the target disk.

Using `dd(C)`

To copy a disk using `dd`, follow these steps:

1. Insert the disk to be copied into floppy drive 0.
2. Insert a formatted disk into drive 1. If necessary, you can format a disk with the `format` command described under "Formatting Floppy Disks" earlier in this chapter.
3. Enter:

```
dd if=/dev/fd0 of=/dev/fd1 count=blkcount
```

Press `<Return>`. The `blkcount` is the number of blocks on the disk to be copied. If you do not know this number, leave the `count=blkcount` section out of the command. `dd` will stop automatically at the end of the source disk.

This command copies the first disk to the second, then displays a record of the number of blocks copied.

Using Floppies for File Storage

To use a floppy for simple file storage, first make sure that the floppy is formatted. Then, place the floppy in the floppy drive. You can use any of the standard XENIX file archiving utilities with floppy disks. These include `tar`, `cpio`, or `dd` formats.

`tar` is recommended for most file-archiving tasks. For example, to place a copy of a file on a high-density floppy disk in `tar` format, use the following command:

```
tar cv filename
```

△ **sysadmsh** users select: Media→Archive

For more information on **tar**, see the **tar(C)** manual page. For more information on **cpio**, **dd**, and **backup** formats, see the associated manual pages.

Making Filesystems on Floppy Disks

You can make a filesystem on a floppy disk similar to how you make one on a hard disk. Filesystems on floppy disks are portable and can be mounted on any XENIX system. A special directory called */mnt* is used for mounting filesystems that do not have a specified mounting point. Note that for system security, you must be logged in as root to use floppy filesystems.

To make a portable filesystem on a floppy disk, use the following procedure:

1. Log in as *root* and enter the command:

```
mkdev fd
```

2. You see the following menu:

```
Choices for type of floppy filesystem.
```

1. 48tpi, double sided, 9 sectors per track
2. 96tpi, double sided, 15 sectors per track
3. 135tpi, double sided, 9 sectors per track
4. 135tpi, double sided, 18 sectors per track

```
Enter an option or q to quit:
```

Enter the number of the disk type desired and press **<Return>**.

3. You see the following prompt:

```
Insert a type floppy into drive 0.
```

```
Press Return to continue or enter q to quit:
```

Press **<Return>**.

Using Floppy Disks

4. Next you see:

```
Choices for contents of floppy filesystem.
```

1. Filesystem
2. Bootable only
3. Root filesystem only
4. Root and Boot (for 96ds15 and 135ds18 floppies only)

```
Enter an option or enter q to quit:
```

Enter **1** and press (Return).

5. The following prompt is displayed:

```
Would you like to format the floppy first? (y/n)
```

If you have already formatted the floppy, enter **n** and the filesystem is immediately created. If the floppy has not yet been formatted, enter **y** and you see:

```
formatting /dev/type  
track 00 head 0
```

The track and head numbers count up as the floppy is formatted. (If */etc/default/format* contains `VERIFY=Y`, the format is also verified after formatting.)

6. When the process is complete, you see:

```
Filesystem creation complete.
```

7. You are then returned to the first menu. Enter **q** and press (Return) to quit. Your floppy now contains a filesystem.

Mounting a Floppy Filesystem

To use a floppy filesystem, you must mount it on your system. For example, a 96tpi floppy would use the following command:

```
mount /dev/fd096 /mnt
```

△ **sysadmsh** users select: Filesystems→Mount

Note that you use the floppy device *fd096* and not *rfd096*. When you mount a floppy filesystem, you must use the name without the preceding “r”. As another example, if you choose to mount a filesystem on a 96tpi disk, use the following command:

```
mount /dev/fd096 /mnt
```

When you give the **mount** command, the shell should return a prompt. This indicates that the filesystem was successfully mounted. You can now use the **cd** command to move into the filesystem and create files there normally. When you are done and you wish to remove the floppy, give the following command:

```
umount /dev/fd096
```

△ **sysadmsh** users select: Filesystems→Unmount

Your filesystem is immediately unmounted. Your files are contained on the floppy and can be stored or transported easily. Note that the current directory must not be on the mounted filesystem, or the unmount command will fail. Always **cd** out of the mounted filesystem before issuing **umount**.

Chapter 17

Adding Multiport Cards, Memory, and Other Bus Cards

Introduction 17-1

Installing Bus Cards 17-2

 Dip Switches and Jumpers 17-2

 Installing the Hardware 17-3

 Using the Manufacturer's Setup Disk 17-3

Adding More Memory 17-5

Adding and Configuring Serial Ports 17-7

Introduction

The bus (or “motherboard”) of your computer is the center of your system. Every system administrator must deal with the bus and the hardware associated with it. To find the bus on your system, you must generally remove the shell from the main body of your computer. Generally, you find a large circuit board with expansion slots for extra boards. These boards are commonly known as *bus cards*.

Bus cards can be extra memory for your system, internal modems, multiport serial boards for extra terminals, controller boards for peripheral devices such as hard disk tape drives, control cards for monitors with color and graphics capabilities, mouse controllers, or other devices. In this chapter we explain a little about bus cards and how to install them in your XENIX system. Installation of most devices with bus cards is explained in detail in other chapters of this guide.

Installing Bus Cards

To install a bus card, you must first shut down the operating system and power down the system. Make sure that the computer is unplugged or you may injure yourself. Before you begin working on the computer, ground yourself by touching a metal object close at hand that is not the computer. Static electricity that builds up and jumps from your hand when you touch the hardware inside the computer can ruin your equipment.

Note

Micro Channel (MCA) and Extended Industry Standard Architecture (EISA) bus cards do not have dip switches and jumpers; use the manufacturers setup disk to adjust settings.

Dip Switches and Jumpers

Before you plug your board into the bus, make sure that there are no settings on the board that must be changed. Again, your hardware documentation that comes with the board should list the default settings and how to change them. Generally, to change the settings of a board, there are dip switches and “jumpers.” Dip switches operate in “down” and “up” positions. Your hardware documentation should list the correct settings if your board has these switches. Jumpers are clips that slide over metal posts that stick out of the board to make a connection. You can change the settings on a board by moving the jumper to connect a different pair of posts. Again, your hardware documentation should provide you with specific instructions for jumper settings on your hardware.

Note

Your XENIX system is designed to work with most hardware using default settings. You will rarely have to change the settings on a board.

Installing the Hardware

Carefully perform any steps necessary to expose the expansion slots on your computer. Your hardware documentation should explain this in detail. Once you can examine this area, note the number of available spaces for bus cards. A new system has up to 8 or 10 available slots. Note that some slots are longer than others. There are both short and long cards. Short cards are about half as long as long cards. Usually there are two to three short slots and the rest are long slots. There may also be different kinds of slots, such as 8-bit slots (with a single edge connector) and 16-bit slots (with two edge connectors). Find a slot that fits your board and gently, but firmly, plug the board into the slot in the bus. The board should have a tab on one side that fits into the slot on the bus. Bus cards only fit one way.

Some bus cards have a port that should face the outside of the computer. As stated before, bus cards only fit into the system one way. There may be a small plate covering an opening in the computer held on with a small screw. You can remove this cover plate if you need to. Boards such as modems, serial and parallel cards, and external device control cards require this.

When you are done, replace the shell for your computer, and turn it on and boot. You may first need to use the manufacturer's setup program as described below to change the system's configuration before you can use the new hardware.

Using the Manufacturer's Setup Disk

Your computer should come with a manufacturer's setup program on a bootable floppy disk. Copy this disk for use and keep the original in a safe place. This disk configures the permanent memory on your computer to describe the system hardware setup. Whenever you add a major device, like an extra hard disk or an extra serial card, you may need to run your

Installing Bus Cards

setup program to tell your computer about the new hardware. Some computers automatically recognize the presence of new hardware. Your manufacturer's documentation should let you know if you need to run this software.

Many newer computers have setup software stored in ROMs and is accessed by entering a key sequence after turning on the computer. Consult the manufacturer's documentation.

Adding More Memory

You can improve system performance and run larger programs by increasing the amount of internal memory.

To increase internal memory follow these steps:

1. Turn off your computer. Steps for this task are provided in the “Starting and Stopping the System” chapter of this guide.
2. Install extended memory according to the manufacturer’s instructions. Make sure you have set all switches as noted in the instructions.
3. Boot the operating system. The boot screen details how the additional memory has affected your system.
4. Some features may have been expanded. For example, you may have:
 - more multiscreens
 - more buffers
 - a larger maximum user process size

The number of multiscreens may be unchanged. Because the number of multiscreens can be set by the user, you may have already set a specific limit to the number of multiscreens available. If you have not set a limit to the number of multiscreens, then you are already using the maximum number of multiscreens that the system allows.

The number of buffers may also be unchanged. Because the number of buffers can also be set by the user, you may have already set a specific limit to the number of buffers available. If you have not set a limit to the number of buffers, then you are already using the maximum number of buffers that the system allows.

Adding More Memory

If the maximum user process size is unchanged, then it is now limited by the size of the *swap* filesystem instead of the amount of internal memory. You can do the following:

- Reinstall the operating system and increase the size of the swap space.
- Change the process so that it runs without being swapped. Refer to **proct1(S)** for details.

You can follow the same procedure if you wish to remove internal memory from the system.

If the memory hardware reports an error, the following message is displayed:

```
PANIC: memory parity error
```

You then see the software reboot message:

```
** Safe to Power Off **  
    - or -  
** Press Any Key to Reboot **
```

If the system repeatedly panics from parity errors, consider replacing the memory chips. Parity errors can also be caused by video cards with an autoswitch feature.

Note

Some machines have a hardware limitation on the maximum amount of memory that can be installed. Refer to your computer hardware manual to determine the maximum amount of memory you can install.

The XENIX system uses only “extended,” not “expanded,” memory.

Adding and Configuring Serial Ports

To add a multiport expansion card, you must first determine whether the card is a “smart” serial card or a standard serial card. If the card is a “smart” card, the manufacturer will have supplied installation software and a driver. These should be all you need to add the card to your XENIX system.

Before installing your card, check your *Release Notes* for information about hardware compatibility. Follow the instructions for insertion furnished with your card, referring to your computer hardware manual if necessary. If you are using a supported multiport expansion board, check to see if your board is recognized at bootup by checking the XENIX bootup message. If the boot process does not accurately report your board, then the switches on your card are not set properly. Check your board’s hardware documentation for the proper switch settings and the *Release Notes* for the correct addresses. This applies to boards that are listed as supported in the *Release Notes*.

Vendor-supplied drivers may not print a recognition message at boot time. If your serial expansion card is a smart card with a vendor-supplied driver, you should not need to run **mkdev serial** to install it. For your system to recognize the new card, run the vendor-supplied installation software.

Configure the interrupts for the two standard COM ports: COM1 as interrupt 4 and COM2 as interrupt 3. Most serial cards use one interrupt per board, so two multiport boards can use COM1 and COM2. Be aware of the requirements of other products and hardware to avoid interrupt conflicts. See **serial(HW)** for more information on COM1 and COM2.

17

Note

You cannot use the COM3 and COM4 serial ports because there are only two interrupt vectors in the IBM interrupt scheme allocated to COM devices.

Adding and Configuring Serial Ports

If your card is a standard serial card, the following instructions explain how to create new device files for additional ports:

1. Boot the system and enter system maintenance mode.
2. When you are in system maintenance mode, enter:

```
/etc/mkdev serial
```

△ **sysadmsh** users select: System→Add→Card_Serial

3. The following is displayed:

```
Serial Board Initialization

You would like to install a:
  1. 1 port card
  2. 2 port card
  3. 4 port card
  4. 5 port card
  5. 8 port card

Select an option or enter 'q' to quit:
```

Enter the appropriate number and press <Return>.

4. The program responds with the following menu (only COM1 and COM2 appear and are usable on most systems):

```
The card is configured as:
  1. COM1
  2. COM2
  3. COM3
  4. COM4

Select an option or enter 'h' for help or 'q' to quit:
```

If you select “h”, you see a table listing ports, card types, I/O addresses, and status addresses.

Enter a number and press <Return>. After **mkdev** accepts the COM slot, you see a list giving the newly configured ports and their modem control counterparts. For example, **tty2a** and **tty2A** refer to the same serial port, but **tty2A** has modem control, whereas **tty2a** refers to the same port without modem control. You can access the port by only one name at a time, either with or without modem control.

5. When the process is complete, you see messages similar to the following:

```
Modifying system files...
System files have been successfully updated.

The following standard serial device(s) have been installed:
    tty2a tty2b tty2c tty2d

The following device(s) access the same physical port as
their lower case counterparts, but have modem control properties:
    tty2A tty2B tty2C tty2D
```

Now that your serial ports are configured, make sure that they are also defined in the system hardware configuration.

Check your computer hardware manual to determine how your system is configured. If your system is configured using a CMOS database, the ports are defined in the database (see **cmos(HW)**).

If your system is configured with switch settings on the main system board (motherboard), define the new ports by setting the proper switches (refer to your hardware manuals for the settings).

Note

An error message is displayed if you attempt to access a serial port that is not installed and defined.

Chapter 18

Using a Mouse

Introduction 18-1

Configuring the Hardware 18-2

Installing a Mouse 18-3

Testing a Mouse 18-7

Removing a Mouse 18-8

Using the Mouse 18-9

Using the Mouse with Multiscreens 18-9

Using the Mouse on Serial Terminals 18-9

Sharing a Mouse with Several Terminals 18-9

Using a Mouse with Keyboard-Based Programs 18-10

Introduction

This chapter deals with the basics of installing a supported mouse on your system. Using a mouse can be a great convenience for users and developers alike. For this reason, support is provided for both serial and bus mouse hardware.

Configuring the Hardware

Consult your hardware manufacturer's documentation for specific instructions on hardware configuration. Note the brand and type of your mouse and whether it is attached to a serial port or directly to the system bus. For more information about the system bus, see the "Adding Multiport Cards, Memory, and Other Bus Cards" chapter in this guide. You need to know this information when you configure your software to accept the mouse.

Note

Please note the following restrictions regarding mouse usage:

- The Microsoft Bus mouse cannot be configured using interrupt vector 2; use 3, 4, or 5 instead.
 - Do not use the **usemouse** utility while in single-user (maintenance) mode.
 - You cannot invoke System→Terminate (**shutdown**) using the **usemouse** utility.
-

Installing a Mouse

To install a mouse on your system, you must perform the following steps:

1. Install the mouse according to the manufacturer's instructions.
2. Make sure your link kit is installed and functioning correctly. The mouse drivers cannot be installed without the link kit. (The link kit is installed using `custom(ADM)`.)
3. Log in as `root` and input the following command:

```
mkdev mouse
```

You see the Mouse Initialization Menu:

```
Mouse Initialization Program
```

- ```
1. Display current configuration
2. Add a mouse to the system
3. Remove a mouse from the system
4. Associate a terminal with an existing mouse
5. Disassociate a terminal from an existing mouse
6. Remove the mouse drivers from the kernel
```

```
Select an option or enter q to quit:
```

To install a mouse, select option 2 and press `<Return>`. The other options allow you to change your mouse configuration at any time. For example, you can add or remove additional mice on your system or change the terminals that are allowed to receive input from an existing mouse.

## Installing a Mouse

4. Next you must specify the type of mouse you will use. You see the menu:

The following mouse devices are supported:

1. Logitech Serial Mouse
2. Microsoft Serial Mouse
3. Mouse Systems PC II Serial Mouse
4. Mouse Systems PC Mouse
5. Microsoft Bus mouse
6. Olivetti Bus Mouse
7. Logitech Bus Mouse
8. Keyboard Mouse

Select an option or enter q to return to the previous menu:

---

### Note

Certain mice are not supported on XENIX-286 distributions. Check your *Release Notes* for details.

Under XENIX-286, steps 8-10 occur before step 7 and some prompts differ in appearance.

---

Enter the number corresponding to the mouse you wish to install and press (Return).

5. You see:

*mouse\_type* is currently configured  
to attach to the system on /dev/tty

Do you want to install this mouse on a different port? (y/n)

Enter **n** and press (Return) if you do not need to change the default port. Enter **y** followed by (Return) if you wish to change the default and enter a port when prompted.

6. If you are installing a busmouse, you are asked to select the configuration for the busmouse card. If you are installing a serial mouse, skip this step and go directly to step 7. If you chose a busmouse, you see the message:

```

Busmouse Configuration

1. Display current busmouse parameters
2. Modify current busmouse parameters
3. Select previous busmouse parameters
4. Select default busmouse parameters

Enter an option or q to quit:

```

If you wish to use the default busmouse parameters, select **4**. The current parameters are displayed, and you can press **q** to quit this menu. The default busmouse selection auto-configures your busmouse. If you change the interrupt vector, then using interrupt vector 5 conflicts with a cartridge tape device (using the same vector) if both devices are in use at the same time. (This is also true of the */dev/lp2* parallel device.)

7. Next, you are asked to specify the terminals and multiscreens that are allowed to accept input from the mouse. Do not attempt to allow mouse input on any tty where any mice are not physically connected or you receive an error message. You may choose to allow any or all other terminals and console multiscreens to use the mouse. Entering the word **multiscreen** associates all of the console multiscreens.

Note that only one mouse can be allowed for input on a given tty.

For more information on sharing the mouse between several terminals or multiscreens, see “Using the Mouse” later in this chapter. You see:

```

This mouse may be configured for use on any of the system's
terminals and multiscreens. The multiscreens and terminals
that will be associated with this mouse need to be specified.

Specify them by entering, at the following prompt, all the
ttys to be associated with this mouse. Entering the word
"multiscreen" will associate all of the console multiscreens.

Enter a list of terminals (e.g. tty1a tty2a multiscreen)
or enter q to quit. Press return when finished:

```

## Installing a Mouse

Press **<Return>** when you have entered all the devices desired. Now you see:

```
Do you want to use the <mouse_type> on any other terminals?
(y/n)
```

Note that in this example *mouse\_type* is replaced with the brand or type of mouse you specified earlier in the procedure. Respond **n** if no other terminals are allowed to receive mouse input. If you answer **y**, you are returned to the screen prompting for a list of terminals.

8. You see the following messages, which may take a few minutes to appear on your screen:

```
Updating system configuration...
You must create a new kernel to effect the driver change you specified.
Do you wish to create a new kernel now? (y/n)
```

Answer **y** to add the mouse device driver to your kernel.

9. Next, you see:

```
Do you want this kernel to boot by default? (y/n)
```

Answer **y** if you want this kernel to be used every time you boot the system.

10. The following is displayed:

```
Backing up /xenix to /xenix.old.
Installing xenix on hard disk.

You will need to shut down and reboot the system
in order to use the newly installed /xenix into memory.
```

11. You are returned to the main mouse menu again. If you have no changes to make to your mouse configuration at this time, enter **q** to quit and press **<Return>**.
12. Use the **shutdown(ADM)** command (or **haltsys(ADM)** if you are in single-user mode) to shut down the system and reboot.

You can invoke **mkdev mouse** at any time to allow or prevent input on different terminals, remove mice, or check your current configuration.

## Testing a Mouse

Use the following procedure to test your mouse:

1. Log in as *root* in multiuser mode.
2. Enter the following at the system prompt:

```
usemouse -t vi -c "view /etc/termcap"
```

3. Observe the cursor as you move the mouse. The cursor should move as the mouse moves.

Pressing each mouse button should result in the following actions:

- Left button moves the cursor to the beginning of the file.
  - Middle button deletes the current character.
  - Right button moves the cursor to the last line of the file.
4. To stop the **usemouse** utility, enter the following:

```
:q!
```

If invoking **usemouse** does not produce the cursor behavior described here, or results in error messages:

```
Open event driver failed:: No such file or directory
Open event driver failed:: Not enough space...giving up
```

## Installing a Mouse

Your mouse is not installed incorrectly. Check the following:

1. First, verify that your mouse is one of the supported mice listed in your *Release Notes*.
2. If you are using a busmouse, verify that the busmouse card is recognized during boot up of your system and that there is no conflict with the interrupt vector or base address.

To do this, check the hardware configuration information by logging in as *root* and entering **hwconfig** at the system prompt.

Make sure that the mouse is plugged into the busmouse card.

3. If you are using a serial mouse, verify that the serial card to which your mouse is attached is recognized during boot up of your system (by running **hwconfig**).

## Removing a Mouse

Removal of any mouse or the mouse drivers on your system is an exact reversal of the process of installing a mouse. Choose the menu options to remove rather than to add a mouse.

---

## Using the Mouse

Use of a mouse is automatic. If a program or utility accepts mouse input and the terminal is allowed to use the mouse, you simply invoke the program and the mouse works. If the terminal or multiscreen is not allowed to use the mouse, or the program is not configured to accept mouse input, using the mouse has no effect.

### Using the Mouse with Multiscreens

Multiscreens (on monitors attached to video cards in the bus) provide the most convenient method for using the mouse. If a mouse is associated with the multiscreens on your main system console (typically, a monitor attached to a video card in the system bus), the mouse input is associated with the current active multiscreen. For example, if your system has four multiscreens enabled on the main system console and all those screens are allowed to use the mouse, the input from the mouse goes to the program running on the active multiscreen.

Remember that programs that do not accept mouse input are unaffected by moving the mouse, even on a mouse-allowed multiscreen.

Serial (terminal) multiscreens and serial consoles can also be configured to use the mouse.

### Using the Mouse on Serial Terminals

When you install the mouse, you are prompted to list the ttys that are allowed to use mouse input. You can allow terminals on serial lines to use the mouse just as you allow multiscreens. Again, you cannot allow mouse input on a tty where a mouse is physically connected.

### Sharing a Mouse with Several Terminals

When the mouse is shared among several terminals, the mouse is associated with a tty on a “first-come, first-served” basis. The first user to invoke a mouse-enabled program has the mouse for the duration of that program. For another user to use the mouse, the first user must quit the

## Using the Mouse

program. (This closes the input queue from the mouse.) Then, the next user for the mouse can invoke the program and open the line for input from the mouse.

Note that other users on mouse-allowed ttys can use programs that accept mouse input while the mouse is busy. If the mouse is busy, the programs are unable to receive input from the mouse but should otherwise function normally.

## Using a Mouse with Keyboard-Based Programs

The `usemouse(C)` utility maps mouse movements and operations to keystrokes used by keyboard-based programs. Refer to the `usemouse(C)` manual page for complete information.

## Chapter 19

# Troubleshooting Your System

---

Introduction 19-1

Solving Startup (Boot) Problems 19-2

- System Does Not Boot During Installation 19-2
- System Does Not Boot After Successful Boots 19-3
- Restoring Missing or Corrupted System Files 19-3
- /boot Not Found 19-4
- xenix Not Found 19-5
- System Hangs at the Letter Z 19-6
- Cannot Load Floating Point Emulator 19-7
- Can't exec /etc/init or /etc/inir 19-8
- NO OS 19-9
- System Hangs at Login Prompt 19-9
- Restoring the Root Filesystem Super Block 19-9

Resolving Console Problems 19-11

- Console Keyboard Locks Up 19-11
  - Preventing Console Keyboard Lockup 19-12
- Wrong Console Keyboard Type 19-14

Fixing Filesystem Problems 19-15

- Out of Inodes on Filesystem 19-15
- Out of Space on Filesystem 19-16
- Checking Free Space on Filesystems 19-18
- Reducing Disk Fragmentation 19-18
- Restoring a Corrupted root Filesystem 19-18
- Repairing a Filesystem When fsck Stops at Size Check 19-21
- Filesystem Mount Failed 19-24
- srmount: panic cannot cvtv7superb( ) yet 19-25

Troubleshooting Installation Problems 19-26

- divvy: Mount on /mnt Failed 19-26

Troubleshooting Your Modem 19-28

- Errors When Dialing Out 19-28
  - No OK Message 19-28
  - Modem Dials, but Does Not Connect 19-29
  - Connect Failed: NO DEVICES AVAILABLE 19-29

- Modem Answers, but Terminal Displays Garbage 19-30
- Modem Does Not Hang Up 19-31
- Double Echo 19-31
- Problems Dialing In 19-32
  - Modem Does Not Answer the Phone 19-32
  - Modem Answers, but Hangs Up 19-32
  - Garbage or Loose Cable 19-33
  - Modem Answers, But No Login Prompt 19-33
  - Screen Displays a Series of Login Prompts 19-34
  - System Displays Meaningless Characters 19-34
- Troubleshooting the Print System 19-35
  - Printer Does Not Print 19-35
  - Cannot Redirect Output to Printer 19-37
  - Port Not Recognized at Bootup 19-38
  - Printer Output is Illegible 19-39
  - Printer Output Spacing is Wrong 19-39
  - Parallel Printer is Slow 19-40
  - Printer Reports UUCP Errors 19-41
- Resolving Process Errors 19-42
  - No More Processes 19-42
  - Stopping a Runaway Process 19-43
  - Removing an Unkillable Process 19-44
- Fixing Problems with Schedulers 19-45
  - cron Daemon is Not Running 19-45
  - lpsched Print Scheduler is Not Running 19-45
- Recovering from Other System Failures 19-47
  - File table overflow 19-47
  - Bad HZ Value 19-47
  - Inode Table Overflow 19-48
  - Mapping a Bad Track 19-49
  - Recovering from a System Panic 19-50
  - kernel PANIC: Trap E in system: non-recoverable kernel page fault 19-51
  - Recovering After a Power Failure 19-52
- Troubleshooting Your Tape Drive 19-54
  - Tape Not Recognized at Bootup 19-54
    - Cartridge Tape 19-54
    - Irwin and QIC-40/80 19-56
    - SCSI Tape 19-56
  - Tape Commands Hang 19-56
  - Cannot Open /dev/rct0 19-57

|                                            |       |
|--------------------------------------------|-------|
| Solving Terminal Problems                  | 19-58 |
| Restoring a Non-Functional Terminal        | 19-58 |
| Fixing a Hung Terminal                     | 19-59 |
| Fixing a Scrambled Terminal Display        | 19-61 |
| Restoring a Non-Echoing Terminal           | 19-61 |
| Troubleshooting Network Connections        | 19-62 |
| Check for Faulty ACU/Modem                 | 19-63 |
| Errors When Testing the Connection with cu | 19-64 |
| Connect Failed: CANNOT ACCESS DEVICE       | 19-64 |
| Connect Failed: SYSTEM NOT IN Systems FILE | 19-64 |
| Connect Failed, NO DEVICES AVAILABLE       | 19-64 |
| Connected, but No Login Prompt             | 19-65 |
| Connected, but Screen Displays Garbage     | 19-65 |
| UUCP Failed Messages                       | 19-65 |
| Checking the Status of a uucp Request      | 19-66 |
| Debugging UUCP Communications              | 19-68 |
| UUCP Log and Status File Messages          | 19-69 |
| DEVICE LOCKED                              | 19-69 |
| LOGIN FAILED                               | 19-70 |
| NO DEVICES AVAILABLE                       | 19-70 |
| REMOTE DOES NOT KNOW ME                    | 19-71 |
| REMOTE HAS A LCK FILE FOR ME               | 19-71 |
| SYSTEM NOT IN Systems FILE                 | 19-71 |
| RETRY TIME NOT REACHED                     | 19-71 |
| CANNOT ACCESS FILE                         | 19-72 |
| WRONG TIME TO CALL                         | 19-72 |
| Checking UUCP Files Permissions Settings   | 19-72 |
| Verifying That Sitename Is Unique          | 19-73 |
| UUCP is Abnormally Slow                    | 19-73 |
| uucp Works, But uux Does Not               | 19-74 |
| UUCP Troubleshooting Utilities             | 19-75 |



---

# Introduction

This chapter covers common problems that you may have while administering your system. The topics that are covered in this chapter include:

- Situations in which the system does not boot
- Reasons why the console and console keyboard do not work
- Common problems, including repair and recovery, with filesystems
- Problems that come up during the installation procedure
- Troubleshooting information for dial-out and dial-in modems
- Common problems with printers and the print system
- Information on stopping runaway and unkillable processes
- Starting stopped schedulers
- Recovering from other system failures, such as a power failure, system panic, or a bad track on the hard disk
- Fixing common problems with tape drives
- Solving problems with terminals, such as fixing a scrambled display or unlocking a locked terminal
- Troubleshooting the UUCP system

---

# Solving Startup (Boot) Problems

Situations in which the system does not boot can be divided into two categories:

- System does not boot during installation
- System does not boot even though it has successfully booted in the past

This section briefly describes common reasons for why the system does not boot during installation. The discussion of what to do when your system does not boot after successful boots is more extensive. It contains instructions for recovering critical system files and solutions to specific problems, such as what to do when the system hangs at the login prompt and why the system cannot enter multi-user mode.

## System Does Not Boot During Installation

If the system fails to boot from the floppy the first time during installation, one of several things may be wrong.

1. If the floppy in the drive is not the N01 (boot) floppy, the system does not display the boot prompt. Replace the floppy with the N01 floppy and power-cycle the machine.
2. If the floppy in the drive is the N01 floppy, make sure that the floppy is inserted correctly and that the floppy drive door is closed. Then, power-cycle the machine.
3. If you get a persistent read error on your N01 floppy, request a new floppy from your provider.

If the system still does not boot, you most likely have a problem with your hardware.

1. Check your computer's documentation to see that your system has enough RAM (Random Access Memory). Your computer must have the minimum recognized memory listed in your *Release Notes* to install the operating system. If there is not enough RAM, the system cannot boot. See the section "divvy: Mount on /mnt Failed" under "Troubleshooting Installation Problems" later in this chapter for information on memory requirements.

2. If your system has enough RAM, check to see that the boards (bus cards) are seated properly in the motherboard.
3. If the system still does not boot, you may have a hardware failure. Check the documentation that came with your computer for hardware tests.

### System Does Not Boot After Successful Boots

If your system successfully booted in the past but does not boot now, the problem may be due to one of the following situations:

- The floppy drive contains a floppy that is not a boot floppy.
- Critical system files, such as */boot* or */xenix*, are corrupted or missing. See the section “Restoring Missing or Corrupted System Files” later in this chapter for information on booting from floppy disks and restoring these files.
- Your hard disk may have developed a bad track, corrupting system files that are required for booting the system. For information on how to recover from this situation, see the section “Mapping a Bad Track” under “Recovering from Other System Failures” later in this chapter.
- If your system does not have a 287 or 387 math coprocessor chip, and the */etc/emulator* file is missing, the system does not boot. If this is the case, see the section “Cannot Load Floating Point Emulator” later in this chapter.

### Restoring Missing or Corrupted System Files

On rare occasions, one or more of the critical system files is accidentally modified or removed, preventing the system from booting or operating correctly. In cases where your system does not boot, you must boot from floppy disks in order to access the system so that you can restore the critical files from backups.

In order to boot and access a system that does not boot from the hard disk, you must have made an Emergency Boot Floppy as directed in the *XENIX Installation Guide*. The Emergency Boot Floppy contains three files necessary for booting and loading the XENIX kernel: */boot*, */etc/default/boot*, and */xenix*. The floppy also contains a subset of the XENIX utilities that you can use to restore your system.

### Note

You must have a separate Emergency Boot Floppy for each system or further corruption can result.

---

If you have not made this floppy, you must reinstall the operating system. In some cases, if you do not also have a backup of the root filesystem, you must reinstall the operating system. To do this, follow the instructions in “Reinstalling Your System” in the *XENIX Installation Guide*.

## **/boot Not Found**

If your system displays the following message when you turn on the power to your computer, the */boot* file is missing:

```
/boot not found
Stage 1 boot failure: error loading /boot
```

The */boot* file contains the **boot(HW)** program which loads and executes the kernel each time you turn on the computer.

If */boot* is missing, use the following procedure to boot the system from the Emergency Boot Floppy set so that you can then restore the file:

1. Insert the Emergency Boot Floppy in the drive and power-cycle the machine. This executes the initial boot from the floppy.
2. At the boot prompt, enter:

```
hd(40)xenix
```

After you boot the system from the floppy, this command loads the kernel from the hard disk.

3. Bring up the system in single-user mode by entering the root password at the prompt.
4. Mount the floppy filesystem from the command line using:

```
mount /dev/fd0 /mnt
```

5. While the floppy is in the drive, restore */boot* by entering the following command at the system prompt:

```
cp /mnt/boot /
```

This allows you to access the */boot* file on the hard disk.

6. Before you remove the floppy disk from the drive, unmount the floppy filesystem (*/dev/fd0*) by entering:

```
umount /mnt
```

7. Remove the floppy disk from the drive and bring down the system using **haltsys(ADM)**.
8. Reboot the system from the hard disk by pressing <Return> at the "Safe to Power Off" message.

### xenix Not Found

If the system displays the following message when you turn on the power to your computer, the */xenix* file is missing:

```
xenix not found
```

The */xenix* file contains the XENIX kernel. If */xenix* is missing, you can boot from another kernel file, such as */xenix.old*, by specifying the complete pathname of the file at the boot prompt.

If there are no other kernel files on the system, use the following procedure to boot the system from the Emergency Boot Floppy so that you can then restore */xenix*:

1. Insert the Emergency Boot Floppy in the drive and power-cycle the machine.
2. At the boot prompt, enter:

```
fd(64)xenix root=hd(40) swap=hd(41) pipe=hd(40)
```

This executes the initial boot and loads the kernel from the boot floppy and mounts the root filesystem from the hard disk.

3. Bring up the system in single-user mode.

## Solving Startup (Boot) Problems

4. Mount the floppy filesystem to */mnt*. (See step 4 of the previous section, “/boot Not Found.”)
5. While the floppy is in the drive, restore */xenix* with the following command:

```
cp /mnt/xenix /
```

This copies the */xenix* kernel file from the boot floppy to the hard disk.

6. Unmount the floppy filesystem. (See step 6 of the previous section, “/boot Not Found.”)
7. Remove the floppy from the drive and bring down the system with **haltsys(ADM)**.
8. Reboot the system from the hard disk by pressing **<Return>** at the boot prompt.
9. You should relink and reboot the kernel in case the kernel you copied from the floppy is not up to date with any changes made since the creation of the Emergency Boot Floppy.

## System Hangs at the Letter Z

If the system hangs at the letter “Z,” certain system files are probably corrupted. Follow the instructions under “Cannot Load Floating Point Emulator” to mount the hard disk and then enter the following commands:

```
/mnt/bin/rm /mnt/dev/console /mnt/dev/mem /mnt/dev/kmem
/mnt/etc/mknod /mnt/dev/console c 3 1
/mnt/etc/mknod /mnt/dev/mem c 4 0
/mnt/etc/mknod /mnt/dev/kmem c 4 1
/etc/haltsys
```

Remove your floppy diskette and reboot from the hard drive when the “Safe to Power Off” message is displayed. If the system does not boot, follow the instructions under “Can’t exec /etc/init or /etc/inir.”

## Cannot Load Floating Point Emulator

If your computer does not have a 287 or 387 math coprocessor chip and the */etc/emulator* file is missing or corrupted, the boot fails with the following message:

```
WARNING:cannot load floating point emulator
```

(If the 287 or 387 chip is present, the kernel recognizes it in the hardware recognition boot message.)

If the boot fails with this message, use the following procedure to boot the system and restore */etc/emulator*:

1. Insert the Emergency Boot Floppy in the drive and power-cycle the machine.
2. At the boot prompt, press <Return>. This boots the system and mounts the root filesystem from the floppy.
3. You should first check the filesystem with **fsck(ADM)**. At the system prompt, enter:

```
/bin/fsck /dev/hd0root
```

4. After **fsck** exits successfully, mount the hard disk root filesystem:

```
mount /dev/hd0root /mnt
```

If **mount** fails, see the section “Filesystem Mount Failed” under “Fixing Filesystem Problems” later in this chapter for information on checking the hard disk with **fsck(ADM)**.

5. Copy */etc/emulator* from the root filesystem on the floppy to the mounted hard disk:

```
cp /etc/emulator /mnt/etc/emulator
```

6. Unmount the hard disk:

```
umount /mnt
```

7. Make sure that the floppy is still in the drive and shut the system down with **haltsys(ADM)**.
8. Remove the floppy from the drive and press <Return> at the boot prompt to boot from the hard disk.

### Can't exec /etc/init or /etc/inir

If the boot process hangs, displaying the messages:

```
Can't exec /etc/init
Can't exec /etc/inir
```

Your files */etc/init* and */etc/inir* might be corrupted. The */etc/init* file contains the **init(M)** program. Once started, the **init** process spawns all other processes on the system. Without the */etc/init* file, no new processes are started.

Use the following procedure to restore */etc/init*:

1. Insert the Emergency Boot Floppy in the floppy drive and power-cycle the machine.
2. Press  $\langle$ Return $\rangle$  at the boot prompt.
3. You should first check the filesystem with **fsck(ADM)**. At the system prompt, enter:

```
/bin/fsck /dev/hd0root
```

4. After **fsck** exits successfully, mount the hard disk root filesystem:

```
mount /dev/hd0root /mnt
```

If **mount** fails, see the section "Filesystem Mount Failed" under "Fixing Filesystem Problems" later in this chapter.

5. Copy the */etc/init* file from the root filesystem on the floppy to the mounted hard disk:

```
cp /etc/init /mnt/etc/init
cp /etc/inir /mnt/etc/inir
```

6. Unmount the hard disk by entering **umount /mnt**.
7. With the floppy in the drive, reboot the system with **haltsys(ADM)**.
8. Remove the floppy from the drive and press  $\langle$ Return $\rangle$  at the boot prompt to boot from the hard disk.

## NO OS

If you see “NO OS,” the boot block on your hard disk has probably been corrupted. Follow the steps described for booting from the Emergency Boot Floppy and mounting the hard disk under “Can’t exec /etc/init or /etc/inir,” then enter the following commands:

```
/bin/dd if=/mnt/etc/hdboot0 of=/mnt/dev/hd0a
/bin/dd if=/mnt/etc/hdboot1 bs=1k seek=1 of=/mnt/dev/hd0a
/etc/haltsys
```

Now boot from the hard disk. If this boot fails, your last resort is to reinstall XENIX from scratch and restore your root filesystem from your backups.

## System Hangs at Login Prompt

If the system boots correctly but hangs at the login prompt when you enter multi-user mode, the */etc/login* file is missing. The */etc/login* file contains the **login(M)** program. This command is run at the beginning of each terminal session to allow users access to the system. Follow these steps to restore */etc/login*:

1. Power-cycle the machine and press <Return> at the boot prompt.
2. At the prompt, enter the root password to go into single-user mode.
3. Run **custom(ADM)** and select option 1 “Operating System,” followed by option 5 “Install a single file.” Enter */etc/login* when prompted and insert the requested media.
4. Enter **q** to quit out of **custom** and press <Ctrl>d to return to multi-user mode.

## Restoring the Root Filesystem Super Block

If the super block of the root filesystem has been corrupted, the system will not boot. When this occurs, do the following:

1. Insert the Emergency Boot Floppy in the drive and power-cycle the machine.
2. At the boot prompt, press <Return>.

### *Note*

Never use an Emergency Boot Floppy that was created on another system or further corruption will result.

---

3. When the root prompt appears, enter the following command:

```
dd if=/superblock of=/dev/rhd0root bs=1K count=2
```

4. You should then check the filesystem with **fsck(ADM)**. At the system prompt, enter:

```
/bin/fsck /dev/hd0root
```

5. When **fsck** completes, enter the following command to shut the system down:

```
/etc/haltsys
```

You can then reboot the system normally.

---

## Resolving Console Problems

This section covers some common problems that you may have with the console on your system:

- The keyboard on your console locks up
- How to prevent keyboard lockup by applying a special “patch”
- The console does not recognize keyboard input because you selected the wrong console keyboard type
- You cannot log into the console multiscreens when the system is in multi-user mode

### Console Keyboard Locks Up

When the system does not respond to input from the console keyboard, the situation is known as “keyboard lockup.” Console keyboard lockup only affects keyboards that are attached to the video display adapter, not standard terminals that are attached to serial lines.

You may be experiencing keyboard lockup if the following statements are true:

- The system console keyboard cannot be used to enter data or perform any tasks.
- You cannot use the <Alt>-<F1> through <F12> keys to switch multiscreens, and the <CapsLock> key does not turn the CapsLock light on or off.
- Other terminals on the system continue to work.
- Printers or other devices continue to work, and the system is still running.

## Resolving Console Problems

Before trying to fix a locked keyboard, make sure that:

- You did not accidentally press `<Ctrl>s` (which stops the screen from scrolling). To check this, press `<Ctrl>q` and then see if you can enter characters from the keyboard.
- If your computer has a `<Keyboard Lock>` key, it is not in the locked position.
- The keyboard is plugged into the correct socket.
- The system itself is still running.

Check a terminal to see if it is still working and that you can perform system tasks, such as logging in and checking the date. If you do not have a terminal, watch the hard disk access light, if your computer has one. If it flashes periodically, at least once every 30 seconds, the system is still running and is using the hard disk.

---

### *Note*

You cannot use other terminals and the hard disk access light may not flash if you are in single-user mode.

---

If the console keyboard is still locked after checking these suggestions, unplug the console keyboard then plug it in again. If this fixes the problem, your situation is definitely keyboard lockup. If this last step does not fix the problem, you may still have keyboard lockup.

### **Preventing Console Keyboard Lockup**

You can prevent keyboard lockup by applying a special “patch” that changes the operating system kernel. (The kernel is the main program of the operating system that is always running in memory.)

### Note

This patch disables the keyboard lights, so you should use it only if you have tried the other approaches.

---

The procedure for using the patch is as follows:

1. Get the system console working, if it is not. Reboot the system if you have to and bring it up in single-user mode.

If you did not reboot, log in as **root** on the system console and shut the system down to single-user mode with the **shutdown** command:

```
/etc/shutdown su
```

(For more information, see **shutdown(ADM)** in the manual pages.)

2. Once the system is in single-user mode, back up the kernel with the following commands:

```
cd /
cp xenix xenix.00
```

3. Now, patch the kernel with the following commands:

```
/etc/_fst -w /xenix
ledspresent/w 0
$q
```

4. Shut down the system using **/etc/shutdown**.
5. When you see the “Normal System Shutdown” message, press any key to reboot the system. You have now fixed the keyboard lockup problem.
6. Finally, call your support center and report your problem.

### Wrong Console Keyboard Type

If your console keyboard is an XT or other non-AT keyboard and the operating system is configured for use with an AT keyboard, the system does not recognize input from the keyboard. For information on testing and switching keyboard modes, see the section on console keyboard type selection in the “Using the System Console and Color Displays” appendix in this guide.

---

## Fixing Filesystem Problems

This section describes how to free up inodes and create space when a filesystem runs out; how to repair the filesystem when an abnormal shutdown corrupts the data; how to check and repair a filesystem when **mount**(ADM) fails; and explains why a user is unable to remove files from a directory with the *sticky bit* set.

### Out of Inodes on Filesystem

When a filesystem runs out of inodes, the system displays the following error message on the system console:

```
kernel:WARNING:Out of inodes on dev nn/mm
```

Where *nn/mm* is the filesystem that has run out of free inodes. To fix this problem:

1. Remove unnecessary (old, temporary, *core*, or log files) files from the filesystem.
2. Check to see if there is large number of small files on the filesystem using the **find**(C) command with the **-size** parameter. For example, to locate and print a list of all files smaller than 3 blocks in size (**-size -3**) in all the directories (*/* and below), enter:

```
find / -size -3 -print
```

The number of inodes available on a filesystem is determined when the filesystem is created (using **mkfs**(ADM)). If the filesystem consistently runs out of free inodes, you can reconfigure the filesystem and increase the number of inodes. To do this:

1. Back up the filesystem using the procedure in the “Backing Up Filesystems” chapter. First, follow the instructions under “Performing an Unscheduled Backup,” making certain to check the volumes when prompted to do so; this ensures that the backup has no errors.

## Fixing Filesystem Problems

2. Run **mkfs** from the command line and specify more inodes for the filesystem. For example, to reconfigure the number of inodes on the */dev/u* filesystem to 6400, use the following **mkfs** command:

```
mkfs /dev/u fssize:6400 9 400
```

*fssize* must be replaced with the size of the filesystem in question, in 1K blocks. This value can be obtained as described in steps 5 and 6 of the section “Restoring a Corrupted Root Filesystem” later in this chapter. For more information on the **mkfs** command, see the **mkfs(ADM)** manual page.

3. Restore the filesystem from the backup as described in in “Restoring an Entire Filesystem” in the “Backing Up Filesystems” chapter.

## Out of Space on Filesystem

When a filesystem has little or no space left to work, the system displays the following message on the system console:

```
No space left on device: nn/mm
```

When the filesystem runs out of space, the system stops any attempts to write to the filesystem. The only way to restore system operation is to delete or reduce files from the named filesystem.

Use the following suggestions to restore space to the filesystem:

1. Use the **wall(ADM)** command to send a system-wide message asking users to remove unnecessary files.
2. Check for the number of blocks used by each file and directory in the named filesystem with the **du(C)** command. For example, to display the number of blocks in the files in the */u* filesystem, use the following command from */u*:

```
du | sort -nr
```

This command sorts the directories by size and displays them largest first. You can then send mail, asking the users with the largest files to remove them.

3. Use the **find(C)** command to locate exceptionally large or old directories and files, and send mail to the owners asking them to remove unnecessary files. For example, to locate and print a list of all the directories (**-type d**) greater than 3 blocks in size (**-size +3**) in all the directories (**/** and below), enter:

```
find / -type d -size +3 -print
```

4. Use **find** to locate and remove temporary files and *core* files. For example, to find all core files in the */usr* directory that have not been accessed for one week (**-atime +7**), enter:

```
find /usr -name core -atime +7 -print
```

5. Clear the contents of system log files, such as */usr/spool/lplog*, */usr/adm*, in particular, */usr/adm/messages* and */usr/adm/sulog* (if enabled), */usr/preserve*, any unused *a.out* files, files in the *lost+found* directories, and any log files for add-on programs.

To do this, use one of the following constructions:

*Bourne or Korn Shell:*

```
>filename
```

*C-Shell:*

```
cat /dev/null >filename
```

6. Reduce disk fragmentation by following the steps in the section “Reducing Disk Fragmentation” later in this chapter.
7. If the system is chronically short of free space, create and mount an additional filesystem.

Each of these steps is described in more detail in the section on maintaining free space in filesystems in the “Managing Filesystems” chapter in this guide.

### Checking Free Space on Filesystems

To prevent the problem of running out of space on a filesystem, periodically to check the amount of free space in your filesystems with the **sysadmsh** selection:

System→Report→Disk

This command prints the amount of space left in the filesystem in *blocks* of 512 bytes.

### Reducing Disk Fragmentation

If your system has been in use for some time, the constant creation and removal of files creates a situation called *disk fragmentation*. This means that the files in the filesystem are written in small pieces scattered widely across the hard disk. This results in increasingly poor disk I/O performance. Fragmentation becomes a problem when the disk is (roughly) more than 75% full.

To reduce disk fragmentation, use the procedures found in the “Backing Up Filesystems” chapter. First, follow the instructions under “Performing an Unscheduled Backup,” making certain to verify the backup as instructed, then follow the instructions in “Restoring an Entire Filesystem.” We recommend that you create two complete backups.

### Restoring a Corrupted root Filesystem

If your root filesystem is so corrupted that, when you boot the system, **fsck(ADM)** cannot run, use the following procedure to restore your system:

1. Insert the Emergency Boot Floppy in the drive and power-cycle the machine.
2. At the boot prompt, press **<Return>**. This boots the system and mounts the root filesystem from the floppy.

3. At the system prompt, enter:

```
/bin/fsck /dev/hd0root
```

You should see messages indicating that **fsck** is proceeding through five or six phases of system cleaning.

---

### *Note*

If you cannot boot the system or **fsck** does not run, see “Restoring the Root Filesystem Super Block” earlier in this chapter.

---

If **fsck** asks for a scratch file, press the **<Del>** key. When the prompt returns, enter the following command:

```
/etc/mknod ramdisk b 31 26
```

If **fsck** exits within a few seconds or the system displays error messages that make no sense, you must restore the entire root filesystem from backups. Here are two examples of nonsense messages:

```
UNKNOWN FILE SYSTEM VERSION 65535
CLEANING NON SYSTEM 3 FILESYSTEM
```

If **fsck** appears to be successful, shut down the system with **haltsys(ADM)** and boot from the hard disk by pressing **<Return>** at the boot prompt. If **fsck** is not successful, continue with the following steps to restore the root filesystem.

4. Mount the hard disk root filesystem:

```
/etc/mount /dev/hd0root /mnt
```

You must reinstall your operating system if the system displays the following message:

```
mount: cannot access special file
```

See the *XENIX Installation Guide* for more information.

5. If you have a tape backup unit, and previously created a backup of your root filesystem using the **backup(ADM)** command, then booting from the Emergency Boot Floppy will enable you to restore your root filesystem from your tape backups. Any additional

## Fixing Filesystem Problems

filesystems can be restored after rebooting from the hard disk. Boot from the Emergency Boot Floppy and enter the following, replacing *x* with the partition number of your XENIX partition:

```
/etc/divvy -b 1 -c 1 -v x
```

You see a table similar to the following, plus the main **divvy** menu:

| Name | New File System? | # | First Block | Last Block |
|------|------------------|---|-------------|------------|
|      | no               | 0 | 0           | 39023      |
|      | no               | 1 | 39024       | 41250      |
|      | no               | 2 | 41251       | 41502      |
|      | no               | 3 | —           | —          |
|      | no               | 4 | —           | —          |
|      | no               | 5 | —           | —          |
|      | no               | 6 | 41503       | 41512      |
| hd01 | no               | 7 | 0           | 41759      |

41513 blocks for divisions, 247 blocks reserved for the system

6. Enter “q” to quit. Record the information in your table. The divisions will either not have names, or have false ones. The root filesystem is represented by division 0. To find the size of the root filesystem, perform the following calculation:

$$fssize = \text{last\_block} - \text{first\_block} + 1$$

The first block should be zero and so *fssize* should equal *last\_block*+1.

7. Issue the following commands, substituting *fssize* with the value you calculated:

```
/etc/mkfs /dev/hd0root fssize
/bin/fsck /dev/hd0root
/bin/restore fr /dev/rct0 /dev/hd0root
/bin/fsck -s /dev/hd0root
/etc/haltsys
```

Replace */dev/rct0* with the name of your tape drive if you normally access it with a different name.

---

*Note*

If the **restore** command returns with an error message such as: “cannot open /dev/hd0root,” make sure that your XENIX partition is active. If it is, then issue the commands in step 3 and try to restore again. If it still fails, you will need to reinstall the operating system from scratch and reissue the **restore** command.

---

Boot from the hard disk. If you have other filesystems, they may not have been damaged. If they are damaged, you can restore them by going into system maintenance mode and restoring them from tape. For example, the following command is used to restore a /u filesystem:

```
restore fr /dev/rct0 /dev/u
```

## Repairing a Filesystem When fsck Stops at Size Check

If the super block for a filesystem is so badly damaged after an abnormal shutdown or hardware failure that **fsck**(ADM) quits at the size check or reports an unusually large number of errors, you can use the **fsdb**(ADM) (“filesystem debugger”) utility to patch the super block by hand.

The **fsck** utility reads the size of the filesystem from the super block. If the values currently stored in the super block for *FSIZE* (the total number of blocks in the filesystem) and *ISIZE* (the number of inodes allocated in the filesystem) are not the normal values for the filesystem, **fsck** displays the results of the size check and quits, as in the following example:

```
fsck /dev/root
/dev/root
/dev/root File System: / Volume: root
Size check: FSIZE 0 ISIZE 0
#
```

When using **fsdb**, you have to change one or both of the values in the super block, *FSIZE* or *ISIZE*, back to their normal values for your filesystem.

### Note

You cannot run **fsdb** on the root filesystem if you are unable to boot up the system.

Make sure that the filesystem is unmounted before using **fsdb** to repair it.

---

Because **fsdb** allows you to directly change the super block (which contains important information about your filesystem), be very careful when using this tool. If you enter improper values with **fsdb**, you can permanently damage your filesystem and lose all filesystem data. The **fsdb(ADM)** manual page describes a number of other ways in which you can use **fsdb** to look at and manipulate the super block. The use of such advanced features is recommended only for the experienced system administrator.

To repair the damage described above, follow this procedure:

1. Enter the following command:

```
/etc/divvy -b 1 -c 1
```

Because the **divvy** filesystem table is not located in the super block, the information in the table is probably correct. From the **divvy** table, obtain the first and last block number of the filesystem that you are attempting to restore.

2. To obtain the correct value for *FSIZE*, apply the following formula:

$$FSIZE = \text{last\_block} - \text{first\_block} + 1$$

3. To obtain the correct value for *ISIZE*, run **bc(C)**, and enter the following command:

```
((FSIZE * .25) - ((FSIZE * .25) % 16))
```

Where *FSIZE* is the value computed in step 2. Round down any fractions and record this calculation of *ISIZE* for later use. Exit **bc** by entering **quit**.

4. To convert the number of available inodes into the actual address of the first block following the blocks allocated for inodes, you must perform an additional calculation. Refer to this number as *isize* to distinguish it from the *ISIZE* value displayed when **fsdb** starts.

To do this, perform the following calculation:

$$isize=(ISIZE) / 16 + 2$$

Where *ISIZE* is the number that you calculated in step 3.

5. With your *FSIZE*, *ISIZE*, and *isize* values, you can now correct the super block values with **fsdb**(ADM). Enter the following:

```
fsdb /dev/filesystem
```

Where *filesystem* is the name of the filesystem that you want to repair.

6. **fsdb** displays output similar to the following:

```
FSIZE = 52985, ISIZE = 13232
```

If the values that you calculated are close approximations of the actual values, you do not need to modify that parameter. If both values seem correct, something else is wrong and you must restore your data from backups.

---

### Note

You can end the display of addresses at any time by pressing INTERRUPT, <Del>, or <Ctrl>C.

---

7. Decide which parameter (or both), *ISIZE* or *FSIZE*, must be corrected.
8. Next, you must enter **fsdb** commands while substituting the values that you calculated earlier for *FSIZE* and *isize*. (Commands are in bold, **fsdb** responses are not). If *isize* is correct, simply skip it by pressing <Return>.

### Note

Each time you press `<Return>`, **fsdb** displays what the filesystem currently thinks the values of *FSIZE* and *isize* are.

---

Enter the following commands with the proper substitutions:

```
1024<Return>
002000: 000000 (0)
=isize<Return>
002000: 000xxx
<Return>
002002: 000000 (0)
=FSIZE<Return>
002002: 000yyy
q
```

9. If the corrupted *FSIZE* value is so small that **fsdb** thinks that you cannot move that far into the block, the following error message is displayed when you input the beginning address:

```
block out of range
```

To disable error checking, enter a capital **O**. You can then input the address without complaint.

At this point, you can run **fsck** on the filesystem and properly restore the system.

## Filesystem Mount Failed

If the **mount(ADM)** command fails, check the filesystem with the **fsck(ADM)** command before running **mount** again:

```
/bin/fsck -y /dev/filesystem
```

Where *filesystem* is the name of the filesystem that you want to check and repair.

### **srmount: panic cannot cvtv7superb() yet**

If you get this message when attempting to mount a filesystem, it means the super block has been corrupted. You should be able to use **fsck -s** to rebuild the super block. In addition, you must specify the raw device (for example: */dev/ru* instead of */dev/u*). Enter the following command:

```
fsck -s /dev/rname
```

You should be able to mount the filesystem normally. If the error message persists, you must restore the filesystem from backups.

---

# Troubleshooting Installation Problems

This section discusses some problems you may have during initial installation or subsequent reinstallations of the operating system, and minimum memory requirements.

## divvy: Mount on /mnt Failed

If installation fails and the system displays the following messages, your system does not have enough memory:

```
divvy:mount on /mnt failed
invalid argument
cannot set up /dev directory on new device
```

XENIX requires the minimum memory listed in the *Release Notes* to install. In addition, when you are using a minimal memory configuration, the memory must be fully recognized. During installation, much of this memory is used as a RAM swap device. If you do not have enough memory, the system runs out of swap space during the filesystem creation procedure and installation fails.

If your machine has a minimal memory configuration, verify that the operating system recognizes all the memory you have installed. To do this, check the boot screen message. The system displays a message, like the following, after the hardware recognition information:

```
mem: total = 1664k, kernel = 622k, user = 1002k
```

On some 386 machines, the XENIX kernel cannot recognize the memory between 640 and 1024K because the hardware manufacturer has mapped this 384K of memory to another location for their firmware to use. Because this location varies, the XENIX kernel does not know where to find it.

To install the system and avoid the **divvy** error, add enough memory so that the XENIX kernel recognizes the minimum required memory. We strongly recommend that you use 32-bit memory made by the manufacturer.

## Troubleshooting Installation Problems

In addition, if your machine's setup program allows "BIOS shadowing," in which the machine's BIOS ROMS are copied into RAM for faster access, disable it. XENIX does not use the BIOS, and this feature can decrease available memory and cause other problems.

Use one of the following two methods to add memory:

- If the motherboard only has room for 640K, buy an expansion board and start it at 1024K. This is the best solution because no memory gets "wasted."
- If the motherboard already has two megabytes and has additional empty sockets, add at least another 384K to the motherboard. If the motherboard does not have any empty sockets, you must install an expansion board.

See the "Adding Multiport Cards, Memory, and Other Bus Cards" chapter in this guide for more information on adding additional memory.

---

# Troubleshooting Your Modem

This section discusses common problems that you may have with your modem. Note that, while other serial ports are often used, the examples in the modem troubleshooting sections assume that the modem is attached directly to COM1. If you have problems, first verify that the phone jack is plugged in and that you have a dial tone on the phone line.

## Errors When Dialing Out

This section describes some situations and solutions to problems that may occur when dialing out on your modem.

The most useful tool for diagnosing dial-out problems is the **-x9** option to **cu(C)**. This option causes **cu** to display diagnostic output when attempting to dial out. To get a debugging output, enter the command:

```
cu -x9 phone_number
```

where *phone\_number* is the phone number of the system you wish to dial.

### No OK Message

You get a connected message when you test the modem connection with the command:

```
cu -s1200 -ltty1a dir
```

But, when you enter **AT**, the system does not display the “OK” message. Use the following steps to solve the problem:

1. Verify that the modem switch and software settings are correct.
2. Check the modem cable:
  - If you are using a straight-through cable, try a null modem cable using at least pins 2, 3, 7, 8, and 20.
  - After issuing the **cu** command, watch the lights on the modem and press **<Return>** several times. The “receive” light should flash as you hit **<Return>**. If it does not flash, check your cable to make sure that pin 2 is connected correctly (pin 2 is the data transmission line from the serial port to the modem).

3. If the “send” light flashes on the modem, the local echo may be turned off. Use the **ATE0Q1** command to turn on the modem’s echo capability.
4. Verify that the serial port on the computer is not defective:
  - Attach the modem to a different serial port, or attach a terminal or serial printer to the port to confirm that it is functioning.
  - If the port is not functioning, check your hardware documentation for an appropriate repair facility.
5. If the previous steps do not fix the problem, your modem may be defective. If this is the case, check your hardware documentation for an appropriate repair facility.

### Modem Dials, but Does Not Connect

If your modem dials okay, but the call never connects, check the following:

1. Verify that the phone number is correct and operational and that the phone line to which the modem is attached is not faulty. To do this, unplug the modem from the telephone line and plug in a regular telephone. Manually dial the number to make sure that the modem on the other end of the line is answering the call.
2. Listen carefully to your modem while it dials the call. Some business phone systems require a pause between certain numbers. Use a hyphen in the **cu** command to indicate a pause of two seconds. In this example, the modem pauses 8 seconds after dialing the first number: **9---458--1234**.

The dialer translates the hyphen passed to the **cu** command into the appropriate code for your modem. For example, the dialer translates the hyphen into a comma before sending to a Hayes-compatible modem.

### Connect Failed: NO DEVICES AVAILABLE

When you try to dial out on the modem, the following message is displayed:

```
Connect failed: NO DEVICES AVAILABLE
```

## Troubleshooting Your Modem

Follow these steps to solve the problem:

1. Verify that the modem port has an entry in the */usr/lib/uucp/Devices* file. Here are example entries for a Hayes-compatible modem running at 2400 baud on */dev/tty1A*:

```
ACU tty1A - 300-2400 /usr/lib/uucp/dialHA24
Direct tty1a - 2400 direct
```

Make sure that these lines in *Devices* do not begin with a pound sign (#). There should be no spaces in front of the entries.

2. Verify that the modem port in *Devices* has the correct baud rate associated with it. If you specify the baud rate with the **-s** option to **cu**, verify that there is an entry in *Devices* that corresponds to that baud rate.

### Modem Answers, but Terminal Displays Garbage

If the modem answers, but the terminal displays garbage characters, use the following steps:

1. Verify that the site that you are calling is set to the same data bit and parity values that you are using. By default, **cu** uses 8 data bits, and no parity. To change the values to 7 data bits and even parity, enter **cu -e**. For 7 data bits and odd parity, use **cu -o**, and use **cu -oe** for 7 data bits and no parity.
2. Verify that the remote computer is set to the same baud rate that you are using.

If you are dialing into another XENIX system, you can force the remote site to switch to the next lower baud rate by sending a break signal. Always start at the highest baud rate and move down as necessary. To send the break signal during the login sequence, enter:

```
~%b
```

3. Check for noise on your phone line. Noise becomes more acute when operating at 2400 baud or higher. Normally, when there is a problem with line noise, garbage characters appear on the screen continually, as if a system on the other end of the line is trying to send valid data.

### Modem Does Not Hang Up

If your modem does not hang up at the end of a call, check the following:

1. Verify that you are using a modem control port that is configured in the `/usr/lib/uucp/Devices` file. If you are using a non-modem control port, change the port to the corresponding modem control port. For example, the modem control port associated with `tty1A` is `tty1A`.

---

#### *Note*

Non-modem control ports should only be used with terminals, and when configuring the modem.

---

2. If the CD (Carrier Detect) light on the modem does not go off when the call is disconnected, check the modem switches to verify that the modem is set to detect the incoming carrier. If your modem is a Hayes 2400 or compatible, use the **AT&C1** command. This forces the carrier detect line to follow the presence of a carrier on the line.
3. Check the modem switches to verify that the modem is set to detect DTR (Data Terminal Ready). The modem should hang up when DTR goes from high to low. If the modem is a Hayes 2400 or compatible, use the **AT&D2** command.
4. Some modems have a switch that can be set to ignore DTR; make sure that this switch is off.

### Double Echo

If you get a double echo when you dial out on your modem, check the setting for local echo. If local echo is enabled, disable it.

### Problems Dialing In

This section provides solutions for common problem situations that may occur when dialing in on your modem.

#### Modem Does Not Answer the Phone

If the modem does not answer the phone, check the following:

1. Verify that the modem control port is enabled. To enable the modem port, enter the following commands:

```
disable /dev/tty1a
```

```
enable /dev/tty1A
```

2. Verify that the modem is configured to auto-answer. Check your modem switches. If the modem is a Hayes 2400 modem, enter:

```
cu -ltty1a dir
```

Then, use the **ATS0=1** command to tell the modem to answer the phone on the first ring. (Remember to enter **AT&W** to save modem settings.)

3. Verify that the DTR (Data Terminal Ready) line is connected from the computer to the modem. Make sure that Pin 20 is connected. Pins 2, 3, 7, 8, and 20 are required for modem communication.
4. Make certain the ACU entry for this modem in the *Devices* file precedes any Direct entries for the port or the non-modem control counterpart.

#### Modem Answers, but Hangs Up

If the modem answers, but hangs up immediately upon connection, check the following:

1. If the modem is set to auto-answer and to detect DTR, check to see that the DTR line is asserted.
2. Verify that the modem control port is enabled:

```
disable /dev/tty1a
```

```
enable /dev/tty1A
```

3. Verify that the cable is correct. If you are using a straight-through cable with at least pins 2, 3, 7, 8 and 20 connected, verify that pin 20 (DTR) is properly connected.

### Garbage or Loose Cable

The console displays a message like the following when a call comes into the modem:

```
Garbage or loose cable on /dev/tty1A, port shut down
```

Check the following:

1. Verify that your modem is not set to echo back data or send command responses. If the modem is not set up this way, it may be sending a “RING” signal to indicate that the phone you are calling is ringing. Because the CD signal is not active, `getty` interprets this as random data on the serial line. The proper Hayes 2400 modem command is **ATE0Q1**.
2. If you have an internal modem and the above options do not eliminate the error message, your modem may be incompatible. Replace your modem with a standard Hayes-compatible external modem.

### Modem Answers, But No Login Prompt

If the modem answers, but does not display a login prompt, check the following:

1. Verify that the CD line is being asserted by the modem after the modem has answered the phone. Check the switches on your modem or, if your modem is a Hayes 2400 or compatible, use the **AT&C1** command. (Remember to enter **AT&W** to save modem settings.)
2. Make sure that the port is enabled. Enable the port by entering the following command:

```
enable /dev/tty1A
```

## Troubleshooting Your Modem

3. Verify that the modem is using the correct */etc/gettydefs* entry and is selecting the proper baud rate. The modem port device line in the */etc/ttys* file should look like the following:

```
03tty1A
```

The “3” in the */etc/ttys* line is the pointer to the entry in the */etc/gettydefs* file. Entry 3 cycles the modem from 2400-300-1200 baud. Verify that this entry in */etc/gettydefs* is correct.

### Screen Displays a Series of Login Prompts

If the screen scrolls uncontrollably when you log in, usually displaying a series of login prompts, verify that only the modem device is enabled. If the non-modem device is enabled, disable it:

```
disable /dev/tty1a
```

### System Displays Meaningless Characters

If the system displays the login prompt, but no password prompt, or meaningless characters are displayed after the login prompt, verify that the line settings are correct:

1. Determine the serial line settings on the system that you are calling. The standard settings that **cu** uses are 8 data bits, one stop bit, and no parity.
  - If the remote system uses even parity, use **cu -e**.
  - If the remote system uses odd parity, use **cu -o**.
2. If you are dialing into a XENIX system, check the */etc/ttys* file on the remote system to verify that the “pointer” into the */etc/gettydefs* file is correct. The serial line characteristics may not match between the stty settings defined in the third field of the selected *gettydefs* entry. Change the setup for the port to 8 data bits, one stop bit, and no parity.

The entry should look similar to this:

```
3 # B2400 HUPCL OPOST CR1 ECHOE NL1 #
 B2400 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n@!login: # 1
```

---

## Troubleshooting the Print System

This section covers some common problems that you may have with your printer system:

- The printer does not print.
- The system does not recognize the printer port at bootup.
- Redirecting output to the printer port does not work.
- The output from the printer is illegible.
- The spacing on the output from the printer is wrong.
- A parallel printer is abnormally slow.
- A dialout printer reports UUCP errors.

### Printer Does Not Print

If the printer is sitting idle and there is no output, check the following:

1. Make sure that the printer has power.
2. Verify that the printer hardware is working before continuing. Check the printer documentation for a self-test.
3. Check the printer cable and make sure that it is attached properly to the port and the printer. Refer to the owner's manual for your printer for installation instructions.
4. Make sure that the printer is configured properly. To set up your parallel or serial printer to receive data properly, follow the instructions in the section, "Installing a Printer" in the "Using Printers" chapter of this guide.

If the printer is a serial printer, make sure that the baud rate at which the computer sends data to the printer matches the printer's baud rate. For instructions on how to reset the baud rate, see the section "Printer Output is Illegible" in this chapter.

## Troubleshooting the Print System

5. Make sure that the printer is enabled and prepared to accept jobs:

```
/usr/lib/accept printername
enable printername
```

6. Verify that the system recognizes the port at boot time. After the copyright information, the system should display information like the following for each port (examples are for parallel and serial, respectively):

```
parallel 0x378-0x37A 07 - unit=0
serial 0x03F8-0x03FF 04 - unit=0 type=Standard nports=1
```

If the system does not display a similar message for the printer port, follow the instructions in the “Port Not Recognized at Bootup” section later in this chapter.

7. Make sure that the port is configured for the proper interrupt vector and that no other hardware is using that interrupt vector. For information on the available interrupt vectors, see the section “Tape Not Recognized at Bootup” under “Troubleshooting Your Tape Drive” in this chapter. See your hardware documentation for information on configuring your ports.
8. Test the printer port connection by redirecting the output of a command directly to the device.

- For parallel printer **lp0**, enter:

```
date > /dev/lp0
```

- For serial printer **tty1a**, enter:

```
(stty options;date) > /dev/tty1a < /dev/tty1a
```

where *options* are baud rate, parity, or other settings that you want to pass to the serial printer.

If the output from the redirected command does not print, follow the instructions in the section “Cannot Redirect Output to Printer” in this chapter.

If the output prints try submitting a sample file (like */etc/motd*) for printing by invoking **sysadmsh**, and selecting:

```
Dirs/Files→Print
```

If the hardware connections are good and the printer is properly configured and enabled, but is still idle and print requests are queued, check the following:

1. Verify that the **lpsched** process is running:

```
ps -u lp
```

2. Restart the **lpsched** daemon if it is not present:

```
/usr/lib/lpshut
/usr/lib/lpsched
```

3. Check to see that print requests are being queued:

```
lpstat -t
```

This command displays a detailed description of the status of output requests, printer names, and devices.

4. If the printer detects a fault, it does not immediately continue automatic printing. Force a retry by enabling the printer.
5. Check to see if a dialout printer was busy or did not answer, or all dialout ports are busy. The print service waits five minutes before trying to reach a dialout printer again. Force a retry by enabling the printer.

## Cannot Redirect Output to Printer

If you redirect output directly to the parallel or serial port and nothing happens or the system displays the “cannot create” message, check the following:

1. Verify that the device file for the port exists in */dev*. Make sure that this file is a device file and not a text file.
2. Test the cable connection using a cable from a working system.

## Troubleshooting the Print System

3. Print a file from DOS. If you can print a file under DOS but not under the XENIX System, check the following:
  - Verify that the port is recognized at boot time and that it is configured correctly.
  - If the port configuration is correct, and you still cannot redirect output to the port, try using a different device name. For example, for a parallel port, use **lp1** instead of **lp0**; for a serial port, use **tty2a** instead of **tty1a**.
  - If you still cannot print using a different device name, your printer may be defective; check the hardware documentation that came with your printer.
4. If you cannot print from DOS, check the printer hardware configuration. See the documentation that came with your printer.

If you configured your printer correctly and you still cannot redirect output to it, the problem is most likely a hardware malfunction. Recheck the cables and port configuration and consult your hardware documentation.

## Port Not Recognized at Bootup

If your system does not recognize the port at boot time (the “parallel” or “serial” line for your port is not displayed after the copyright information), check the following:

1. Verify that the parallel or serial card is properly seated:
  - Turn the power off and open the machine.
  - Remove and reseat the card in the bus.
2. If you have more than one parallel card, one may be conflicting with the other; remove the second card.
3. Verify that the card is correctly configured; check the documentation that came with the card. If possible, try setting the card for a different configuration.
4. The card may be defective; replace it.

## Printer Output is Illegible

If the printer prints illegible output, check the following:

1. Determine the baud rate for the serial printer and check to see that it matches the baud rate for the computer. (If the printer is connected with a parallel port, the baud rate does not matter.) Set the baud rate to 9600 baud for optimum performance.

To set a different baud rate for the print service to use, edit the printer interface script for that printer and modify the **stty** line. The interface script has the name you gave the printer and is located in */usr/spool/lp/interface*. When the change is complete, submit a sample file for printing.

2. Determine the parity setting for the printer and check to see that it matches the computer's parity setting. (If the printer is directly connected to the computer with a wire that is less than 50 feet long, it does not have to use the parity bit.)

To set the parity bit, edit the printer interface script for that printer and add one of the following to the **stty** line:

|                |                              |
|----------------|------------------------------|
| <b>oddp</b>    | Sets odd parity generation.  |
| <b>evenp</b>   | Sets even parity generation. |
| <b>-parity</b> | Sets no parity (default).    |

Select the option that matches what your printer requires.

3. Check to see that the tabs are set correctly. See the next section "Printer Output Spacing is Wrong."

## Printer Output Spacing is Wrong

If the printer output is legible, but the spacing is wrong, you should edit the interface script and modify the **stty** line. The interface script has the name you gave the printer and is located in */usr/spool/lp/interface*. For each of the symptoms described, adjust the following **stty** line:

- If the printer output is double-spaced, add either the **-onlcr** or **-tabs** option.
- If there is no left margin and the text runs together, add the **-tabs** option.

## Troubleshooting the Print System

- If the printer output zig-zags down the page, add the **onlyr** option. (This is set by default, but you may have cleared it accidentally.)

### Parallel Printer is Slow

If your parallel printer prints abnormally slow, verify that the configuration settings are correct using the section “Installing Printers” in the “Using Printers” chapter of this guide.

If printing is still slow after verifying that the parallel ports are configured correctly, the problem may be that your parallel port is not capable of generating interrupts.

To speed up printing on your parallel printer, you can alter the way that the hardware and the printer driver communicate. The parallel printer driver can be made to *poll* a parallel port so that the driver does not rely on interrupts from the parallel port.

---

#### Note

When the printer driver polls a parallel port, you may experience a drain on system resources.

---

To set up polling for a parallel port or parallel printer, create what is known as a *special device node*. Use the following procedure:

1. Note which parallel printer ports are recognized during the bootup message. (You can also look at the last boot message in */usr/adm/messages*.) For example:

```
parallel 0x378-0x37A 07 - unit=0
```

`unit=0` refers to *lp0*, and so on.

2. Log in as *root*.

3. Create a special device file for the printer by entering one of the following commands:

For **lp0**, enter: **mknod /dev/lp0p c 6 64**

For **lp1**, enter: **mknod /dev/lp1p c 6 65**

For **lp2**, enter: **mknod /dev/lp2p c 6 66**

4. Enter the following commands for each printer that you want to configure. For example, enter the following for */dev/lp0*:

```
chown bin /dev/lp0p
chgrp bin /dev/lp0p
chmod 222 /dev/lp0p
```

5. If you are using the print spooler, you must now inform the spooler of the new parallel poll device. Invoke **mkdev lp** and select option 3 “Configure an existing printer” and instead of selecting a standard parallel device name, use either */dev/lp0p*, */dev/lp1p*, or */dev/lp2p*.

## Printer Reports UUCP Errors

If UUCP is configured, the print service uses the UUCP software to handle dialout printers. If a dialing failure occurs and you receive printer fault alerts, the print service reports the same error reported by the UUCP software for similar problems. (If you have not arranged to receive fault alerts, this information is mailed to the user **lp**, by default.) For more information, see the UUCP error messages in the “Building a Remote Network with UUCP” chapter of this guide and the section “Troubleshooting UUCP Connections” later in this chapter.

---

# Resolving Process Errors

This section discusses how to recover from situations in which the system runs out of processes (the process table is full), a runaway process locks up a keyboard, or a process is unkillable. This section also covers how to reconfigure the maximum number of processes systemwide and per user.

## No More Processes

The number of simultaneous processes that each (non-super) user can have is limited. Each process running on the system uses one entry in the kernel process table. If an attempt is made to create a new process when there are none available (when the process table is full), or when the process limit for a UID is exceeded, the **fork** fails and the system displays the following error message:

```
No more processes:
```

This error produces no console messages.

When the system displays this error message, use the **ps(C)** command to check for a *runaway process*. To display all the processes on the system, enter:

```
ps -ef | more
```

For more information on this command, see the **ps(C)** manual page.

The system can run out of available processes when a program enters an infinite loop and spawns new processes. If this is the problem, refer to the section “Stopping a Runaway Process” later in this chapter.

If the system consistently displays the “No more processes” error message, use the following procedure to evaluate whether your system needs tuning:

1. Use the following command to determine the current size of the process table:

```
cd /usr/sys/conf
./configure -y NPROC
```

2. Determine the number of processes currently running when the system is under heavy load with the following command:

```
pstat -p | head -1
```

3. The value for NPROC should be at least 10% greater than the number of processes running at peak use time.

If the NPROC value is too small, you should increase the value and thus reconfigure the kernel to allow for more processes. To increase the maximum number of processes per user, you should modify the MAXUPRC parameter.

For information on tuning system parameters, refer to “Reallocating Kernel Resources with configure” in the “Tuning System Performance” chapter of this guide.

## Stopping a Runaway Process

A *runaway process* can cause an error that locks up the keyboard, preventing anything that the user types from reaching the system. Because of this, a runaway process cannot be stopped from the terminal at which it was invoked.

To continue, you must identify and stop the runaway process:

1. Log in as **root** on a terminal (or console screen) that is not locked.
2. Enter the following command and press <Return>:

```
ps -ef
```

The system displays full information on all the current processes and their process identification numbers (PIDs). Find the PID of the runaway program.

3. Enter **kill PID**. The program should stop in a few seconds. If the process does not stop, enter **kill -9 PID**.

The last command may leave temporary files that are usually removed when a program terminates normally, or a nonechoing terminal. To restore the terminal to normal operation, follow the instructions in the section “Restoring a Non-Echoing Terminal” under “Solving Terminal Problems” later in this chapter.

### Removing an Unkillable Process

If the runaway process does not stop when you enter **kill** with the **-9** option, the process is considered *unkillable*. If the unkillable process is a user's shell, you must stop the process before that user can continue working. To stop an unkillable process, you must reboot the system. Use the following procedure:

1. Log in as **root** and start the shutdown process by entering:

**/etc/shutdown**

Specify the number of minutes to shutdown, giving users sufficient time to log out.

2. Reboot the system by pressing <Return> at the boot prompt.

The unkillable process no longer appears when you run **ps**.

---

## Fixing Problems with Schedulers

This section explains how to restart the **cron** and **lpsched** daemons.

### **cron** Daemon is Not Running

The **cron** daemon executes commands submitted with the **at(C)**, **batch(C)**, and **crontab(C)** commands at specified dates and times. (The **cron** daemon is started automatically by a script in the */etc/rc.d* directory at system startup.) If **cron** is not running and you try to submit a job with one of these programs, the system displays the following message:

```
cron may not be running - call your system administrator
```

Another indicator that **cron** is not running is if previously scheduled jobs are not being executed. Use the following command to see if the **cron** daemon is running:

```
ps -ef | grep cron
```

If there is no **cron** process, then the daemon is not running. To start **cron**, log in as the super user **root** and enter **cron** at the system prompt. The system displays a message like the following:

```
!***cron started***pid=4612 Mon Dec 18 17:44:19 PST 1989
```

The jobs scheduled with **at**, **batch**, and **crontab** should now execute properly. If **at** and **batch** continue to report error messages, see section “at Command Fails: Cannot Change Mode of Job” later in this chapter.

### **lpsched** Print Scheduler is Not Running

If the print service stops in the middle of a print request, or does not start any new print jobs, determine if the */usr/lib/lpsched* daemon is running. To do this, enter the following command:

```
lpstat -r
```

## Fixing Problems with Schedulers

(The **lpsched** daemon is started automatically by a script in the */etc/rc.d* directory at system startup.) If the scheduler is down, the system displays the following message:

```
scheduler is not running
```

If this is the case, use these steps to start the **lpsched** daemon:

1. Log in as the super user *root*.
2. Invoke the following commands:

```
/usr/lib/lpshut
/usr/lib/lpsched
```

It may take a minute or two for the printer configurations, to be re-established before any saved print requests start printing. Any print requests that did not finish printing when the scheduler stopped are printed in their entirety when the print service restarts.

---

# Recovering from Other System Failures

This section covers problems caused by system failures, such as relieving resource constraints (messages indicating file table overflow, out of streams or queues), how to map a bad track on the hard disk, how to reset the HZ value, how to tune the system if the inode table overflows, what to do when the system panics, and how to recover from an abnormal shut-down due to power failure.

## File table overflow

This message indicates there are insufficient entries in the kernel's file table. This is determined by the NFILE parameter.

See “Reallocating Kernel Resources with `configure`” in the “Tuning System Performance” chapter of this guide for instructions on adjusting kernel parameters.

## Bad HZ Value

If the `/etc/default/login` system file is incorrect or corrupted at any time during system operation, the system displays the following message,

```
Bad HZ Value
```

The operating system uses the HZ variable to represent the system interrupt clock frequency. The HZ value must be set in the `/etc/default/login` file to 50 cycles per second:

```
HZ=50
```

If the HZ variable is not set correctly in `/etc/default/login`, or the file is corrupted or missing, the system displays the HZ error message.

## Recovering from Other System Failures

If you verify that `/etc/default/login` is present and correct, this error message may indicate that your kernel is not properly serialized. Use the following procedure to re-serialize your kernel:

1. Log in as **root**.

2. Back up the `/xenix` file:

3. Enter:

```
/etc/brand <serial#> <activationkey> /xenix
```

Your serial number and activation key are on the card with your distribution and used at installation time.

4. Reboot the system with the `/etc/shutdown` command.

5. Press `<Return>` at the boot prompt.

The system should no longer display the HZ error message.

## Inode Table Overflow

Each open file requires an inode entry in the inode table. If the inode table is too small, the system displays the following error message:

```
Inode table overflow
```

When the inode table overflows, the specific request is refused. Although not fatal to the system, inode table overflow may damage the operation of various spoolers, daemons, the mailer, and other important utilities. Abnormal results and missing data files are a common result.

If the system consistently displays this error message, use the following procedure to evaluate whether your system needs tuning:

1. Determine the current size of the inode table by entering:

```
cd /usr/sys/conf
./configure -y NINODE
```

2. Determine the number of active inodes by running the following command when the system is under heavy load:

```
pstat -i | head -1
```

3. The value for NINODE should be at least 10% greater than the number of active inodes at peak use time.

If the NINODE value is too small, you should increase the value and thus reconfigure the kernel to reallocate inode table entries.

For information on tuning system parameters, refer to “Reallocating Kernel Resources with configure” in the “Tuning System Performance” chapter of this guide.

## Mapping a Bad Track

During the installation procedure, the system scans the hard disk for flaws and creates a bad track table with this information. The bad tracks listed on the table are aliased to good tracks so that operating system avoids the areas of the disk that cannot be read or written.

If your hard disk develops a bad track after the system is installed and running, an error message, like the following is displayed on the console:

```
wd: ERROR : on fixed disk ctlr=0 dev=0/47 block=31434 cmd=00000020
 status=00005180, sector = 62899, cylinder/head = 483/4
```

If this error occurs, use **badtrk(ADM)** to create a new entry in the bad track table so that the system can avoid the new bad track(s). The **badtrk** utility is a menu-driven utility for viewing, adding, or deleting entries in the bad track table. To use **badtrk**:

1. Log in as **root** and enter single-user mode:

```
/etc/shutdown su
```

2. Enter **badtrk** at the system prompt.
3. Select option **2** to scan the disk, then select option **1** to scan the entire XENIX partition.
4. Indicate whether you want to do a quick or thorough scan and, at the destructive scan prompt, enter **n**.
5. You are also asked if you want to recover data on the bad tracks. Respond **y**.

### *Note*

Run **badtrk** in nondestructive mode to save the data on your hard disk. The thorough scan is recommended if new bad tracks have appeared.

---

6. When the scan is complete, the main menu reappears. Select **q** to return to the system.

The **badtrk** utility automatically enters any flaws it detects in the bad track table. For more information on the options to **badtrk**, see the manual page on **badtrk(ADM)**.

## Recovering from a System Panic

The system “panics” when it encounters a hardware problem or kernel inconsistency that is so severe that the system cannot continue functioning. When this happens, the system displays a message on the console, and all system activity stops.

The system panic messages contain the word “PANIC” in the **severity** field, followed by a diagnostic message. For a list of the system panic messages, see the manual page on **messages(M)**. The system can crash without displaying a PANIC message. When this happens, the system simply refuses to process any input (including BREAK, INTERRUPT, and CANCEL keys) from the system console and all other terminals.

To recover from a system panic:

1. Copy the PANIC message, if any, from the console screen to your system log book.
2. Power-cycle the machine and press **<Return>** at the boot prompt to reboot the system.
3. At the prompt to check the root filesystem, answer **y**.

4. Bring the system up in single-user mode and run **fsck(ADM)** on those filesystems that were mounted when the system panicked. For example, use the following command to check the `/u` filesystem:

```
fsck /dev/u
```

5. Fix the problem that caused the system to panic.

If the system does not restart, or crashes each time you start it, the operating system is corrupted and must be restored or reinstalled. To do this, follow procedures in the section “Restoring a Corrupted root Filesystem” earlier in this chapter.

---

### *Note*

If you cannot start the system from the boot floppy disk in the distribution set for installation, the computer has a serious hardware malfunction. Contact a hardware service representative for help.

---

In most cases, simply rebooting the system solves the problem. However, if your system consistently shuts down with the same PANIC message, you should fix the problem that is causing the system to panic.

For example, if the call-out table contains too few entries, the console consistently displays the following PANIC message when it crashes:

```
PANIC:Timeout table overflow
```

Fix this problem by increasing the value of NCALL kernel parameter. For information on tuning system parameters, refer to the “Tuning System Performance” chapter of this guide.

## **kernel PANIC: Trap E in system: non-recoverable kernel page fault**

The 80386 microprocessor generates this error message when it receives a reference to an invalid or nonexistent memory address. This is usually due to bad hardware generating an illegal memory address, but it could also be caused by a poorly written device driver or another piece of hardware. The next step is to strip your machine down to its minimum hardware components and run the default XENIX kernel. You can then add

## Recovering from Other System Failures

hardware and third-party device drivers one by one to determine when the problem occurs. Replace or repair the defective piece of hardware as needed.

Other possible causes for a Trap E are a bad spot on the hard drive or possible bad memory on the disk controller card.

Also look for a conflict between the system's hardware cache and base addresses of hardware.

## Recovering After a Power Failure

When your system goes down as the result of a power failure, the shutdown is considered "abnormal." When the system is shut down normally, the **shutdown(ADM)**, program stops all daemons, kills the active processes, unmounts any mounted filesystems, tells **init** to enter single-user mode, and runs the **sync** command. If the system goes down before this shutdown procedure completes, the following may occur:

- Filesystems may be inconsistent or corrupted, resulting in lost data.
- Ongoing work by users and other data may be lost because the buffer cache was not flushed to disk.

Because XENIX writes to disk every 30 seconds, the amount of data lost due to system shutdown without the **sync** command should be minimal. However, because the filesystems were not unmounted properly before the system shut down, filesystem corruption may be extensive. If the root filesystem is corrupted, the system does not function properly.

When the power fails, turn the machine off. This minimizes damage to your system if the power fluctuates.

Once the power comes back on, restore your system using the following steps:

1. Turn on the computer and press `<Return>` at the boot prompt.
2. If the root filesystem is corrupted, the system prompts you whether to check the `/dev/root` filesystem; enter `y` to invoke **fsck(ADM)**. The system may display the following message:

```
FREE INODE COUNT WRONG IN SUPERBLK (FIX?)
```

This message is routine when the system is not shut down properly. Enter **y** and **fsck** fixes the problem.

The **fsck** command automatically fixes any problems with the root filesystem when autobooting, if the */etc/default/boot* file contains the following line:

```
FSCKFIX=YES
```

If */etc/default/boot* does not contain this line, **fsck** prompts you for instructions on how to fix the problems that it encounters.

3. If any non-root filesystems are corrupted when the system enters multi-user mode, the system prompts you whether to clean the filesystem; enter **y**.

The **fsck** command automatically fixes any problems with the filesystem at this time if the entry for that filesystem in the */etc/default/filesys* file reads:

```
fsckflags=y
```

If the filesystem line in */etc/default/filesys* does not contain this line, **fsck** prompts you to fix any problems with the corrupted filesystem.

---

# Troubleshooting Your Tape Drive

This section covers some common problems that you may have with your cartridge, floppy, or SCSI tape drive on your system. Note that these problems generally occur immediately after using **mkdev tape** to install your tape drive.

For more information about installing and configuring tape drives, see the “Using Floppy Disks and Tape Drives” chapter in this guide.

## Tape Not Recognized at Bootup

If your system does not display a message like the following at boot time, the system does not recognize the tape controller card:

```
%tape 0x338-0x33C 05 1 type=W
```

Instead of this hardware recognition message, the system displays a message like the following:

```
ct: ERROR: Tape controller (type=W) not found
```

## Cartridge Tape

For a cartridge tape drive, check the following:

1. Verify that the tape controller card is physically configured to the base address that you gave when you ran **mkdev tape**. To check this, run **hwconfig(C)**. If this value is incorrect, either physically reconfigure these settings on the tape controller so that it agrees with the values that you indicated when you ran **mkdev tape**, or run **mkdev tape** again and change the parameters.
2. Use **hwconfig** to verify that the DMA (Direct Memory Access) Channel, base address, and interrupt vector do not conflict with any other hardware device on your system. If one or more of these values conflicts, run **mkdev tape** and modify the settings.

Table 19.1 lists the interrupt (IRQ) vectors that are commonly used by system hardware.

**Table 19.1**  
**Interrupt Vector Usage**

| Interrupt | Device                                        |
|-----------|-----------------------------------------------|
| 0*        | clock                                         |
| 1*        | console                                       |
| 2         | available (often entered as <b>25</b> )       |
| 3         | COM2 ( <b>tty2a</b> )                         |
| 4         | COM1 ( <b>tty1a</b> )                         |
| 5         | <b>lp2</b> (alternate parallel port)          |
| 6*        | floppy disk controller                        |
| 7         | <b>lp0</b> or <b>lp1</b> (main parallel port) |

\* Do not use these interrupts.

---

### Note

For more information about configuring interrupts on your cartridge tape drive, see the section “Using Cartridge Tape Drives” in “Using Floppy Disks and Tape Drives” in this guide.

- 
3. Verify that the tape controller is seated properly on the motherboard and, if necessary, insert the controller in a different slot.
  4. Some tape drives (particularly external drives) require that the drive be attached to the tape controller and powered on at boot time.
  5. Verify that the cartridge tape drive is supported. See Appendix A, “Compatible Hardware,” of the *Release Notes* for a list of the compatible tape drives.
  6. If your system still does not recognize the tape controller card at boot time, the tape controller is broken. See the documentation that came with your hardware for more information.

## Troubleshooting Your Tape Drive

### Irwin and QIC-40/80

For an Irwin (mini-cartridge) or QIC-40/80 tape drive, verify that the jumper is set for the correct floppy position on the back of the floppy tape drive.

### SCSI Tape

For a SCSI tape drive, run **mkdev tape** to check the following:

1. Verify that the ID number for the controller of the device is correct. The ID number is determined by the jumper settings on the controller. The valid range is 0-7.
2. Make sure that the host adapter number is correct. The first SCSI host adapter is 0; the second is 1.
3. Check that the LUN (Logical Unit Number) is correct. In most cases, the controller is embedded in the same physical unit as the device and supports one device with LUN 0. If the controller is not embedded, it supports up to eight devices. If this is the case, the LUN is determined by the jumper settings on each device. The valid range is 0-7.
4. Verify that the tape drive is supported. See Appendix A, "Compatible Hardware," of the *Release Notes* for a list of the compatible tape drives.

## Tape Commands Hang

If the tape drive hangs when you test it with the following command:

```
tape reset
```

check the following:

1. Verify that the cartridge tape controller card is physically configured for the DMA value that you gave when you ran **mkdev tape**. To fix this, either physically reconfigure the tape controller DMA setting to agree with the address that you gave with **mkdev tape**, or run **mkdev tape** again and specify the DMA value on the tape controller. Do the same for the interrupt vector. Remember that hardware interrupt 2 is entered as 25.
2. Verify that the cable between the tape controller and the tape drive is properly attached.

3. If the **tape reset** command still hangs, either the tape controller, the tape drive, or the cable is broken. See the documentation that came with your hardware for more information.

### Cannot Open /dev/rct0

The the **tar(C)** command or other tape commands fails to write the contents of a directory to a tape and the system displays the following message:

```
Cannot open /dev/rct0
```

Check the following:

1. Use **hwconfig** to verify that the interrupt vector for a cartridge tape drive does not conflict with any other device on your system. See the section “Tape Not Recognized at Bootup” earlier in this chapter for a list of the available interrupt vectors. To change the interrupt vector, run **mkdev tape** and indicate a new interrupt vector value.
2. Verify that the actual physical interrupt (IRQ) setting on a cartridge tape controller card agrees with the interrupt vector that you specified with **mkdev tape**.
3. Make sure that the proper device for the tape drive is located in the */dev* directory. Enter:

```
l /dev/rct0
```

The output should look like the following:

```
crw-rw-rw 1 root root 10, 0 Feb 14 12:00 rct0
```

If the listing of */dev* does not contain a line similar to this, run **mkdev tape** again to create the device.

4. Verify that the cable between the tape controller card and the tape drive is connected correctly.
5. If the **tar** command still does not work, either the tape controller, the tape drive, or the cable is broken. See the documentation that came with your hardware for more information.

---

# Solving Terminal Problems

This section explains how to solve the following problems with terminals:

- Terminal is completely non-functional
- Terminal is hung
- Terminal display is scrambled
- Terminal is locked
- Terminal does not echo

## Restoring a Non-Functional Terminal

A completely non-functional terminal displays no login prompt and does not respond to keyboard input. This situation is usually caused by hardware failure or configuration problems. Follow these steps to check a non-functioning terminal:

1. Check the brightness control on the terminal.
2. Check the power and communication connections at the terminal and computer.
3. If applicable, enter set-up mode on the terminal and verify the terminal configuration settings. The settings should include 9600 baud, 8 data bits, 1 stop bit, and no parity.
4. Enable the port to which the terminal is connected. For example, to enable tty007, use the following command:

```
enable tty007
```

5. Verify that there is a **getty** process associated with the terminal port. For example, enter:

```
ps -t tty007
```

6. Test the hardware communications by disabling the port and redirecting output to the non-functional terminal. For example, to test `tty007`, use the following commands:

```
disable /dev/tty007
echo hello > /dev /tty007
```

If these steps do not restore the non-functional terminal, check the documentation that came with the terminal hardware for troubleshooting suggestions.

### Fixing a Hung Terminal

A terminal is considered *hung* if the previous work session is still visible on the display, but it does not respond to keyboard input. Use the following steps to fix a hung terminal:

1. Wait a minimum of 60 seconds before trying to resurrect the terminal. (If the system is busy, the terminal may not respond immediately to keystrokes because the system response time has increased.)
2. Press `<Ctrl>q` to re-enable transmission in case the `<Ctrl>s` (transmit off) signal was inadvertently pressed.
3. Check to see that all power cords, keyboard cords, and communications cables are connected.
4. Reset the terminal hardware by recycling power to the terminal and then reinitialize it by running `tset(C)` with no arguments.
5. Verify the terminal set-up mode configuration settings (if available) as described in step 3 of the previous section.
6. Test the hardware communications by redirecting output from an operating terminal to the locked one as described in step 6 of the previous section.

## Solving Terminal Problems

7. Check the processes that are running on the locked terminal port with the following command:

```
ps -t ttyn
```

*Substitute n* with the actual port number. Stop the process that the user was running when the terminal hung using **kill(C)**. For information on how to do this, see the section “Stopping a Runaway Process” earlier in this chapter. If the program does not die, you must reboot the system to stop the process. See the section “Removing an Unkillable Process” earlier in this chapter.

8. Determine whether the current line characteristic parameters are correct. For example, use the following command to display these values for tty007 :

```
stty -a < /dev/tty007
```

You can also compare the **stty** settings with those of a working terminal.

9. Reset the serial line characteristics with the following command:

```
<Ctrl>j stty sane <Ctrl>j
```

If you cannot enter the command on the terminal, you can redirect the **stty** command from another terminal as follows:

```
stty sane < /dev/tty007
```

If the **ps -t** command shows only a **getty** program, the terminal should display a login prompt. If it does not, check the terminal hardware again.

## Fixing a Scrambled Terminal Display

A scrambled terminal responds to keyboard input but the display is incorrect. Follow these steps to fix a scrambled terminal:

1. Check the terminal type (TERM) for the user with the **env** command. If the terminal type is incorrect, reset it. For example, to set the terminal type to **wyse60**, enter one of the following set of commands:

*Bourne or Korn Shell:*

```
TERM=wy60; export TERM
TERMCAP=/etc/termcap; export TERMCAP
```

*C Shell:*

```
setenv TERM wy60
setenv TERMCAP /etc/termcap
```

After resetting the terminal type, reinitialize the terminal by entering **tset** with no arguments.

2. Reset serial line characteristics with the following command:

```
<Ctrl>j stty sane <Ctrl>j
```

## Restoring a Non-Echoing Terminal

A non-echoing terminal is a terminal that responds to keyboard input but does not display the characters entered at the keyboard. (This is different from a *locked* or non-functional terminal that does not respond to input at all; see “Restoring a Non-Functional Terminal” earlier in this chapter for a solution to this problem.)

Sometimes, when a program stops prematurely as a result of an error, or when the user presses the BREAK key, the terminal stops echoing. To restore the terminal to normal operation, enter the following:

```
<Ctrl>j stty sane <Ctrl>j
```

Enter this command accurately because the terminal does not display what you enter at the keyboard.

The terminal should now display keyboard input. If it does not, follow the steps outlined in the section, “Restoring a Non-Functional Terminal” earlier in this chapter.

---

# Troubleshooting Network Connections

This section discusses common problem situations that you may experience with UUCP and covers the following topics:

- How to check for a faulty ACU/modem
- Error messages that **cu** displays
- Common errors when running **uucp**
- How to check the status of a **uucp** request
- Using **uutry** to debug UUCP communications
- Common messages that appear in the UUCP log and status files
- How to check permissions settings on UUCP files
- Checking that your sitename is unique
- What to check if UUCP is very slow
- What to do when **uucp** works, but **uux** does not
- A description of the UUCP troubleshooting utilities, **uucheck**, **uulog**, **uuname**, **uustat**, and **uutry**

Before troubleshooting the UUCP system, make sure that the physical connection works. If you are connecting to UUCP with a modem, verify that the modem is installed and configured correctly. If you are using a direct line, make sure that the computers are connected correctly. Use the instructions in the sections “Connecting Two Local Systems Using a Direct Wire” and “Connecting Remote UUCP Systems with a Modem” in “Building a Remote System with UUCP.”

### *Note*

**cu** must work in order for UUCP to function.

---

## Check for Faulty ACU/Modem

The following are two methods for checking whether the ACU (Automatic Call Unit) or modems are working correctly:

- To display a list of queued requests, the time of the last request attempt, and the contact status, enter:

```
uustat -q
```

- To use a specific line and print debugging information during the contact attempt, enter:

```
cu -x9 -lline
```

---

### *Note*

To protect the modem from interference from unqualified users, this command is only permitted for those who have write access to the *Devices* file.

---

For information on solving common modem problems, see the section “Troubleshooting Your Modem” earlier in this chapter.

### Errors When Testing the Connection with cu

This section describes messages that the system displays when testing the connection with the **cu** command fails.

#### Connect Failed: CANNOT ACCESS DEVICE

If the connection fails and the system displays the CANNOT ACCESS DEVICE message, check the permissions on the device file. For example, to check the device file for **tty1a**, enter:

```
l /dev/tty1a
```

The ownership and permissions settings should look like the following:

```
crw--w--w- 1 uucp uucp 5, 0 Feb 14 12:00 /dev/tty1a
```

If the ownership or permissions differ, change them. You should also check the *Systems* and *Devices* files for consistency. Make certain there are no spaces in front of *Systems* and *Devices* entries.

#### Connect Failed: SYSTEM NOT IN Systems FILE

This message is displayed by **cu** if the */usr/lib/uucp/Systems* file on your computer does not contain an entry for the system that you are trying to access.

To display a list of all the systems that your system is connected to, enter **uname**.

The system may display this error message if you used the **-l** option to specify a serial port and you entered the wrong line number. Verify that the line is configured properly.

#### Connect Failed, NO DEVICES AVAILABLE

If **cu** fails to connect and displays the NO DEVICES AVAILABLE message, check to see that the */usr/lib/uucp/Devices* file is set up correctly.

Verify that the line that corresponds to the device that you are using is uncommented. For example, the entry in *Devices* for a direct line using **tty1a** at 9600 baud looks like this:

```
Direct tty1a - 9600 direct
```

If the *Device* file looks correct, the remote line may be busy. Try again later.

### Connected, but No Login Prompt

If **cu** displays the **Connected** message, but not the login prompt, the line for the remote system may be busy. To exit **cu**, enter `~.` and press `<Return>`.

If everything appears to be working on your system but the login prompt for the remote machine does not appear, check with the remote system administrator to verify that the **getty** on the remote system is set up with the same communications parameters that you are using. By default, **cu** uses 8 data bits, and no parity. To change the values to 7 data bits and even parity, enter **cu -e**. For 7 data bits and odd parity, use **cu -o**, and **cu -oe** for 7 data bits and no parity.

### Connected, but Screen Displays Garbage

If you connect to the remote system, but your screen displays garbage, the connection may be bad. Exit **cu** by entering `~.` and try again later.

Another possibility for this situation is that the communications settings on your system are different from the settings on the remote system. Check with the system administrator on the remote system.

## UUCP Failed Messages

Errors that UUCP displays immediately after you enter the **uucp** command are generally syntax or permissions errors, and include either of the following messages:

```
uucp failed completely
uucp failed partially
```

The following are some common problems and the error messages that the system displays:

- UUCP displays the following message when the permissions are set so that **uucp** cannot access the source file:

```
can't read file (filename)
uucp failed partially
```

Change the permissions on the source file to allow all users read permissions.

## Troubleshooting Network Connections

- The following message is displayed when you do not enter both a *source* and *destination* with the **uucp** command:

```
usage uucp from ... to
uucp failed completely
```

- If you enter a sitename that is not in the */usr/lib/uucp/Systems* file, UUCP displays:

```
bad system name: sitename
uucp failed completely
```

Use **uname** to display a list of the sites to which your system is connected.

## Checking the Status of a uucp Request

If the system does not display an error message immediately after entering the **uucp** command, and the system prompt appears, the **uucp** request is queued. Use the **uustat** and **uulog** utilities to check the status of your **uucp** job.

Use the **uustat** command to display the status of the currently queued **uucp** requests or display the status of connections to other systems. For example, to display a list of all the queued jobs for the user *robertm*, enter:

```
uustat -u robertm
```

The **uustat** utility displays user status information in the following format:

```
couscousN266 01/26-15:43 S couscous robertm rmail edwarda
```

To list the status of the accessibility of all the remote machines, use the following command:

```
uustat -m
```

The following example shows the output from the **uustat** command:

```
scooter 01/26-15:43 CAN'T ACCESS DEVICE
disco 01/27-12:01 SUCCESSFUL
obie 01/25-05:12 CALL FAILED, JOB DELETED
```

For more information on the options to **uustat**, see **uustat(C)** in the manual pages.

The **uulog** command displays the status messages (most recent last) that the UUCP programs write to the files *uucp/sys*, *uux/sys*, *uucico/sys*, and *uuxqt/sys*, in the */usr/spool/uucp/.Log* directory (*sys* is the name of the system that **uucp** is trying to access).

For example, to display status information about file transfers involving the system *scooter*, enter the following:

```
uulog -s scooter
```

The **uulog** utility displays information about the system in the following format:

```
uucp scooter (01/26-15:43:00, 304, 5) COPY (SUCCEEDED)
```

---

### Note

The **uulog** command displays all the status messages that **uucp** logs in the *.Log* files; the messages are not necessarily error messages.

---

### Debugging UUCP Communications

If you are unable to contact a particular machine, use the following procedure to debug the communications to that machine with **uucp** and **uutry**.

1. Log in as **root**.
2. Enter the following commands, pressing  $\langle$ Return $\rangle$  after each line:

```
uucp -r testfile machine! /usr/tmp/testfile
/usr/lib/uucp/uutry -r machine
```

Where *testfile* is a file that you want to transfer and *machine* is the sitename of the machine that you are trying to contact. The **-r** option to **uutry** overrides the retry time specified in */usr/lib/uucp/Systems*.

3. When you see one of the following messages, press the  $\langle$ Del $\rangle$  key:

```
Conversation Complete: Status SUCCEEDED
Conversation Complete: Status FAILED
```

If your system displays the first message, check the remote system, *machine* for the file */usr/tmp/testfile*. If the file exists and is not empty, the UUCP connection functions. If the system displays the second message, the connection transfer failed.

If the UUCP system is set up correctly, the system attempts to transfer the *testfile* file to the remote system, *machine*. The **-r** option to **uucp** queues the file without starting the transfer. This allows **uutry** to invoke **uucico** with debugging.

---

#### Note

The **uutry** command displays more debugging information if you run it as **root**.

---

The **uutry** command saves the debugging output to the file */tmp/machine* on the local machine and prints the debugging output to your terminal using **tail -f**. Copy the output from */tmp/machine* if you wish to save it. You can change the debugging level from the default level (five) using the **-x** option.

If the system displays the “Status FAILED” message, you can look through this file for status and error messages.

See the following section, “UUCP Log and Status File Messages,” for an explanation of some common UUCP messages.

If you still cannot solve the problem, you may need to call support personnel. Save the debugging output for diagnosing the problem.

### UUCP Log and Status File Messages

UUCP keeps track of activity in log and status files in the `/usr/spool/uucp` directory. The UUCP programs **uucp**, **uux**, **uucico**, and **uuxqt** write status information for each system to files in the `/usr/spool/uucp/Log` directories. The **uucico** utility writes messages to the files in the `.Status` directory. These messages describe the status of the transfer request, such as **uucico** failures, completions, and the time until the next allowable call for each remote system with which you communicate.

The following sections describe common error messages that UUCP programs write to these files.

#### DEVICE LOCKED

The **uucico** creates a lock file named `LCK..sitename` for the remote system in the `/usr/spool/uucp` directory. If a file for the system that you are trying to call exists, **uucico** thinks that the device is in use.

To use the device, enter the following command on the remote system:

```
rm /usr/spool/uucp/LCK*
```

This can also be caused by a modem that does not do CD (carrier detect) properly.

## Troubleshooting Network Connections

### LOGIN FAILED

If this error message appears, one of several things may be wrong:

1. Check the */usr/lib/uucp/Systems* file on the local machine to verify that the information in this file, particularly the chat script, is current. Some things that may be out-of-date are the phone number, login, and password.
2. Verify that the modem setup on both ends is correct. See the section, "Troubleshooting Your Modem," earlier in this chapter for more information.
3. Check the phone connection; try a different phone line.

After each step, invoke the **uucp** command.

### NO DEVICES AVAILABLE

If the system displays this message, one of several things may be wrong:

1. The system thinks that the modem is in use because there is a lock file for it. Enter the following from the local machine:

```
rm /usr/spool/uucp/LCK*
```

2. There may be no valid device for the calling system to use. Verify that the device named in the */usr/lib/uucp/Systems* file corresponds to the entry in */usr/lib/uucp/Devices*.
3. If the message persists, reboot the system. Enter the **rm** command again, and then invoke the **uucp** command again.
4. If the message persists after rebooting the system, the modem may be configured incorrectly. Verify that you can use **cu** to call out. For more information, see the section "Troubleshooting Your Modem" earlier in this chapter.

### REMOTE DOES NOT KNOW ME

If the system displays this message, the */usr/lib/uucp/Systems* file on the machine that you are trying to call does not contain an entry for your machine. Check the *Systems* file on the remote machine.

### REMOTE HAS A LCK FILE FOR ME

If your system displays this message, either the remote system is trying to call your system or there is a lock file for your system on the remote system.

1. Remove the lock file from the remote system by entering:

```
rm /usr/spool/uucp/LCK*
```

2. Invoke the **uucp** command again.

### SYSTEM NOT IN Systems FILE

If the */usr/lib/uucp/Systems* file on your system does not contain an entry for the system that you are trying to access, this error message is displayed. Verify that the system name is in the *Systems* file using the **uname** command.

### RETRY TIME NOT REACHED

When UUCP requests fail, retries are not executed immediately. After an attempt to contact a remote system, a status file remains in */usr/spool/uucp/.Status/nodename* (*nodename* is the name of the remote system that you are trying to reach). This file contains information about the last request and does not allow another request until the minimum retry period (specified in the */usr/lib/uucp/Systems* file) is reached. If you try to use UUCP again, the system displays an error message like the following:

```
RETRY TIME NOT REACHED
```

To enable another call attempt immediately, remove the status file for the remote system from the */usr/spool/uucp/.Status* directory:

```
rm /usr/spool/uucp/.Status/nodename
```

## Troubleshooting Network Connections

### CANNOT ACCESS FILE

If the system displays this message, it cannot access the calling device port. Check the permissions and ownership on the device file:

```
l /dev/ttyxx
```

where *xx* is actual device number. The ownership and permissions settings should look like the following:

```
crw-r--r-- 1 uucp uucp 5, 0 Feb 14 12:00 /dev/ttyxx
```

### WRONG TIME TO CALL

The */usr/lib/uucp/Systems* file may restrict outgoing calls at this time.

## Checking UUCP Files Permissions Settings

UUCP does not work correctly if it cannot read or execute its files. Because virtually all of the UUCP files are writable only by the super user, and many of them are also readable and executable only by **root** and **uucp**, you should log in as **root** to install and modify the UUCP system. When you have finished, verify that all of the UUCP files are owned by **uucp** and not **root**.

To check the permissions of the UUCP files, use the following commands:

```
cd /
fixperm -n -v -dUUCP /etc/perms/*
```

This **fixperm** command displays a list of any UUCP files with incorrect permissions.

---

#### Note

The *Systems* and *Permissions* files contain unencrypted passwords, and therefore should be readable only by **uucp** (and **root**). Note also that, in order to work correctly, the */usr/bin/ct* program must be owned by **root** and not by **uucp**.

---

## Verifying That Sitename Is Unique

Make sure that the first seven characters of your sitename name are unique. If your machine is connected directly to a machine with the same sitename, UUCP does not allow communication with that machine. If both your machine and another machine with the same name are not directly connected, but are on the same UUCP network, electronic mail may go to the wrong machine.

To check the sitename, enter:

```
cat /etc/systemid
```

To change the sitename, either edit the */etc/systemid* file manually, or invoke **uuninstall** and select option **1** to display or change your sitename.

## UUCP is Abnormally Slow

Because UUCP is a batch network, requests are not executed immediately. Therefore, when users report that UUCP does not work, the problem may be that the system is slow. If the system is slower than usual, check for the following problems:

1. Check the spool directories, */usr/spool/uucppublic* and */usr/spool/uucp* for an overload of old work files and remove them.
2. Use **ps(C)** to verify that there are not too many **uucico** processes going at once. The limit to the number of communications processes is specified in */usr/lib/uucp/Maxuuscheds* and */usr/lib/uucp/Maxuuxqts*. Check to see that the limit is not too high; the default is 1.
3. Make sure that the filesystem that contains the spool directory is not out of space. For more information, see the section “Out of Space on Filesystem” under “Fixing Filesystem Problems” earlier in this chapter.
4. The */usr/lib/uucp/uudemon.clean* script performs general cleanup by removing old files that are trying unsuccessfully to execute. To change the frequency of the cleanings, edit the **find** statement in the *uudemon.clean* script.

### uucp Works, But uux Does Not

If you can use **uucp** to transfer files between two systems, but you cannot use **uux**, there is a problem with the `/usr/lib/uucp/Permissions` file. When you use the **uucp** utility, the remote system requires only the LOGNAME entry in *Permissions*; **uux** also requires the MACHINENAME entry.

To fix this problem, add the MACHINENAME entry, with the name of the remote system, to *Permissions*. For example, if your local machine, *couscous* set up to call *obie*, the entry for *couscous* in the *Permissions* file on *obie* should look like this:

```
LOGNAME=uucouscous MACHINE=couscous \
COMMANDS=ALL \
READ=/ \
WRITE=/ \
NOREAD=/etc \
NOWRITE=/etc \
SENDFILES=yes REQUEST=yes
```

---

#### Note

The permissions granted in the example above are very liberal and should only be used in closely coupled systems where there is no security risk.

---

## UUCP Troubleshooting Utilities

Table 19.2 lists several commands you can use to check for basic communications information.

**Table 19.2**  
**UUCP Troubleshooting Tools**

| Command        | Description                                                                                                                                                                                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>uucheck</b> | Allows you to check for the presence of files and directories required by <b>uucp</b> . This command also checks the <i>Permissions</i> file for obvious errors.                                                                  |
| <b>uulog</b>   | Displays the contents of the log directories for specific hosts.                                                                                                                                                                  |
| <b>uuname</b>  | Lists the machines that you are set up to contact.                                                                                                                                                                                |
| <b>uustat</b>  | Display the status of the currently queued <b>uucp</b> requests or connections to other systems.                                                                                                                                  |
| <b>uutry</b>   | Invokes <b>uucico</b> with debugging, saves the information to the file <i>/tmp/machine</i> , and directs the last 10 lines of the output to the terminal. The <b>-x</b> option changes the debugging level (default is level 5). |



## **Appendix A**

# **Customizing System Startup**

---

Introduction A-1

Changing Scripts in /etc/rc.d A-2

Modifying .profile and .login Files A-5

Changing the /etc/motd File A-6



---

# Introduction

When your system is switched on and booted, certain aspects of system operation are set up. The system reads initialization files at startup, and whenever a user logs in. By modifying these files, you can adapt system startup.

The system initialization files contain commands or data that set the system clock, enable terminals, start programs, check and mount specified filesystems, clean up temporary directories, set home directories and terminal types for users, and display system messages. The files discussed here are the scripts in the */etc/rc.d* directory, *.profile*, *.login*, *.cshrc*, and */etc/motd*. The system administrator can modify the startup files to create any desired initial system and user environment. For example, by adding or changing entries in the */etc/ttytys* file, specific terminals can be enabled (or disabled).

---

## Note

An */etc/inittab* file is provided for limited compatibility with UNIX System V. It can be used to automate the enabling and disabling of ttytys. Under XENIX, the */etc/ttytys* file is used to enable and disable ttytys. (To generate and use an *inittab* file, see the **telinit**(ADM) manual page.) XENIX */etc/inittab* does not have the ability to specify programs to be executed at start up. Instead, */etc/gettydefs* includes a field wherein programs other than **login**(M) can be specified. See “The gettydefs File” in the “Adding Terminals and Modems” chapter in this guide and **gettydefs**(F) for details.

---

By changing a script in the */etc/rc.d* directory, process accounting can be started automatically at system startup. The administrator can also customize a specific user’s environment by modifying the *.profile* or *.login* file in their home directory.

The initialization files are ordinary text files and can be modified using a text editor such as **vi**(C) (see the *XENIX User’s Guide*). The scripts in */etc/rc.d* and the */etc/profile* and */etc/cshrc* files contain commands and comments in the command file format described in “The Shell” chapter in the *XENIX User’s Guide*.

---

## Changing Scripts in /etc/rc.d

Upon entering multiuser mode from single-user mode, **init** executes the **/etc/rc** script. The **rc** script sets certain environment variables and runs scripts in the */etc/rc.d* directory. The scripts that **rc** runs are executed in alphabetic order to ensure proper initialization.

This section describes the scripts in the */etc/rc.d* directory that are run by **rc** and explains the steps for adding your own script. Table A.1 lists the subdirectories in */etc/rc.d* and their purpose.

**Table A.1**  
*/etc/rc.d* Directories

| Directory Number | Purpose              |
|------------------|----------------------|
| 0                | System init          |
| 1                | System daemons       |
| 2                | Mount filesystems    |
| 3                | Crash recovery       |
| 4                | Clean                |
| 5                | Hardware download    |
| 6                | Kernel software init |
| 7                | User daemons         |
| 8                | User definable       |
| 9                | System reserved      |

Table A.2 gives a brief description of the scripts in */etc/rc.d* in the order in which they are executed.

**Table A.2**  
**Default */etc/rc.d* Files**

| Filename                    | Function                                    |
|-----------------------------|---------------------------------------------|
| <i>/etc/rc.d/0/sysinit</i>  | system initialization after normal boot     |
| <i>/etc/rc.d/1/sdaemons</i> | system daemons executed after normal boot   |
| <i>/etc/rc.d/2/mntfs</i>    | mount file systems after normal boot        |
| <i>/etc/rc.d/3/recovery</i> | crash recovery after boot                   |
| <i>/etc/rc.d/4/cleanup</i>  | removal of temporary or lock files          |
| <i>/etc/rc.d/5/hdload</i>   | hardware download after normal boot         |
| <i>/etc/rc.d/6/kinit</i>    | kernel software init after normal boot      |
| <i>/etc/rc.d/7/udaemons</i> | user daemons started after normal boot      |
| <i>/etc/rc.d/8/userdef</i>  | user definable commands executed after boot |
| <i>/etc/rc.d/9/reserved</i> | system reserved commands after normal boot  |

The */etc/rc.d* subdirectories on your system may contain scripts other than the ones listed in the previous table. The reason for this is that, during installation, many add-on programs insert their own daemon-initialization scripts in these directories. These directories may also include scripts that clean up the temporary or lock files for an add-on program.

You can write your own scripts to run when the system enters multiuser mode. For example, you can write a script that sets up a RAM disk or starts a network and add it to a subdirectory of */etc/rc.d*.

Note that your script may rely on the existence of services or daemons (such as network services, the **cron** daemon, or the print scheduler) started by other scripts. For example, if your script depends upon certain filesystems being mounted, make sure that your script runs after the **mntfs** script. (When you add a new filesystem with **mkdev fs**, the appropriate files are updated with the information necessary to mount the new filesystem when the system enters multiuser mode.)

## Changing Scripts in */etc/rc.d*

To add a function to the initialization procedure, follow these steps:

1. Write a script that performs the desired function.
2. Copy the script into the proper subdirectory of */etc/rc.d* to have it executed by **rc** when the system enters multiuser mode. Note that the scripts are executed in alphabetical order by directory and within each directory.

If the function that you want to add is in the same category as functions performed by a script already located in */etc/rc.d*, simply edit the existing script to add the new function. You can also edit any script to tailor it to your needs. For example, to start process accounting, uncomment the appropriate lines from the *kinit* file. Remember to back up the original script before modifying it.

---

## Modifying `.profile` and `.login` Files

The `/etc/profile` and `/etc/cshrc` files set the default environment for all users of the Bourne/Korn and C shells, respectively. The `.profile` and `.login` files in the C-shell user home directories contain commands that initialize the environment for each individual user. When a Bourne or Korn shell user logs in, the shell first executes the commands in `/etc/profile` and then executes the commands in the `.profile` file in the user's home directory. When a C-shell user logs in, the shell executes the commands in `/etc/cshrc` and then the commands in the user's `.login` file in their home directory. (Depending on the login shell, other files may apply.)

The files contain commands that set various system variables (for example, `TERM`, `PATH`, and `MAIL`). These variables give the system information such as what terminal type is being used, where to look for programs that the user runs, where to look for the user's mailbox, and what keys to expect for the "kill" and "backspace" functions. For more information about these environment variables, see the chapters about the Bourne shell and C shell in the *XENIX User's Guide*.

There is one `.profile` and/or `.login` file for each user account on the system. The files are placed in the user's home directory when the account is created. Users can modify their own `.profile` or `.login` files or allow the system administrator to make modifications. In either case, these files are ordinary text files and can be modified using a text editor; commands can be added or removed as desired.

---

# Changing the */etc/motd* File

The message of the day file, */etc/motd*, contains the greeting displayed whenever a user logs in. Initially, this file contains the name and version number of the operating system. Because */etc/motd* is an ordinary text file, the system administrator can change the message by editing the file with a text editor. In general, you should limit the size of the */etc/motd* file to include no more than a single screen of information.

You can modify this file to include messages such as a reminder to clean up directories to preserve disk space, a notice of the next periodic backup, a description of the latest system upgrade, or information about upcoming scheduled system down times. When you announce an unscheduled system shutdown with **wall**, you should edit *motd* to include the shutdown information so that users who log in after you run **wall** are notified of the impending shutdown.

## **Appendix B**

# **Using the Job Scheduling Commands: at, cron and batch**

---

Introduction B-1

Automatic Program Execution with cron B-2

Automatic System Cleanup with cron B-4

Delaying Program Execution with batch and at B-5

Permitting Users Access to Job Scheduling B-9

Managing cron Jobs with the queuedefs File B-10



---

## Introduction

This chapter explains how to use the **cron(C)**, **at(C)**, and **batch(C)** programs to schedule or delay the execution of programs (jobs). Each of these utilities is subject to authorization by the system administrator, who can permit or deny their use by ordinary users.

The role of the job scheduling programs is as follows:

- |              |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| <b>cron</b>  | Executes programs repeatedly at a specified time.                                            |
| <b>at</b>    | Delays the execution of programs until a time specified by the user.                         |
| <b>batch</b> | Delays the execution of programs until the system load is low (as determined by the system). |

This appendix explains how the user or system administrator can use each of the job scheduling programs to automate regular operations or delay execution of programs that would otherwise slow down the system during peak usage.

---

# Automatic Program Execution with cron

XENIX systems allow you to have programs run automatically at specified times. This is done with the **cron** program. The **cron** program and, more specifically, the **crontab** command allow you to run programs during off-hours such as

- filesystem administration
- long-running, user-written shell procedures
- cleanup procedures

Any task that needs to be done repeatedly at a specified time is a candidate for your *cron* file located in the */usr/spool/cron/crontabs* directory. You can use the **crontab** command to establish the entries you want.

The **crontab** command is used as follows:

```
crontab file
crontab -r
crontab -l
```

The **crontab** command copies the specified *file* (or standard input if no file is specified) into a directory that holds all users' crontabs. Each user has one crontab file. The **-r** option removes the user's crontab from the *crontab* directory. The **-l** option will list the *crontab* file for the invoking user.

---

## Note

Invoking **crontab** *file* overwrites the existing crontab file for that user. To modify an existing file, use **crontab -l** and direct the output into a file so that it can be edited first before using it to replace the old one.

---

See the **crontab(C)** command in the *XENIX Reference* for additional information.

Each line in the *crontab* file defines one procedure. The line entry format looks like the following:

*minute hour day month day-of-week command*

**B**

Each field is defined as follows:

*minute* (0-59),  
*hour* (0-23),  
*day* (1-31),  
*month* (1-12),  
*day-of-week* (0-6 with 0=Sunday)  
*command* (the command to be executed at the time specified)

The following rules apply to the first five fields:

- Two numbers separated by a hyphen indicate a range of numbers between the two specified numbers.
- A list of numbers separated by commas indicates only the numbers listed will be used.
- An asterisk specifies all legal values.

For example, 0 0 1,14 \* 2 indicates a command will be run on the first and fourteenth of each month, as well as on every Tuesday.

If a percent sign (%) is placed in the command field (sixth field), the operating system will translate it as a new-line character. Only the first line of a command field (character string up to the percent sign) is executed by the shell. Any other lines are made available to the command as standard input.

For example, let a file called *anyfile* contain the following **cron** entry:

```
0 0 1 * * mail $LOGNAME % Subject: Call zursch! % now
```

When the command line **crontab anyfile** is executed, the user whose login is **\$LOGNAME** will get a reminder mail message with “Call zursch!” as the subject the first of every month.

### Automatic System Cleanup with cron

The XENIX System has to be cleaned up occasionally. Fortunately, you can get out of some cleaning with the help of the **crontab** command and the *crontab* file. You can specify cleanup jobs (e.g., remove aged files) and the time you want them to execute in the *crontab* file.

The section “Automatic Cleaning of Log Files” in the “Maintaining Filesystems” chapter of this guide contains a good example of a cleanup script and a *root* crontabs entry. The example uses the following line to execute a script called */usr/bin/cleanlog* each Friday at 11 P.M.:

```
0 23 * * 5 /usr/bin/cleanlog
```

The following files are cleared by the *cleanlog* script:

- */usr/adm/messages*: This file contains a history of boot messages and other console messages.
- */etc/wtmp*: This file contains a history of system logins. Every time a user logs in, a record is made in this file. As you can see, the size of this file grows forever, and it needs to be limited. Instead of deleting the contents of this file yourself, you can have **cron** do it for you.
- */usr/lib/cron/log*: This file contains a history of all actions taken by **cron**.
- */usr/spool/lp/log*: This file contains a history of all print jobs submitted to the system.

## Delaying Program Execution with batch and at

The **batch** and **at** commands allow you to specify a command or sequence of commands to be run at a later time. With the **batch** command, the system determines when the commands run; with the **at** command, you determine when the commands run. Both commands expect input from standard input (the terminal); the list of commands entered as input from the terminal must be ended by pressing **<Ctrl>d**.

The **batch** command is useful if you are running a process or shell program that uses a large amount of system time. The **batch** command submits a batch job (containing the commands to be executed) to the system. The job is put in a queue and runs when the system load falls to an acceptable level. This frees the system to respond rapidly to other input and is a courtesy to other users. Note that if the system load is light, the submitted **batch** job is executed immediately.

The general format for **batch** is

```
batch
 first command
 .
 .
 .
 last command
<Ctrl>d
```

If there is only one command to be run with **batch**, you can enter it as follows:

```
batch command_line
<Ctrl>d
```

The next example uses **batch** to execute the **grep** command at a convenient time. Here **grep** searches all files in the current directory for “dollar” and redirects the output to the file *dol.file*:

```
$ batch "grep dollar * > dol-file"
<Ctrl>d
job 155223141.b at Fri Dec 7 11:14:54 1990
$
```

## Delaying Program Execution with **batch** and **at**

After you submit a job with **batch**, the system responds with a job number, date, and time. This job number is not the same as the process number that the system generates when you run a command in the background.

Figure B-1 summarizes the syntax and capabilities of the **batch** Command.

**Figure B-1** Summary of the **batch** Command

| <b>batch</b> - executes commands at a later time |                                                                                                                                     |                      |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <i>command</i>                                   | <i>options</i>                                                                                                                      | <i>input</i>         |
| <b>batch</b>                                     | none                                                                                                                                | <i>command_lines</i> |
| <b>Description:</b>                              | <b>batch</b> submits a batch job, which is placed in a queue and executed when the load on the system falls to an acceptable level. |                      |
| <b>Remarks:</b>                                  | The list of commands must end with a <Ctrl>d.                                                                                       |                      |

The **at** command allows you to specify an exact time to execute the commands. The general format for the **at** command is

```
at time
first command
.
.
.
last command
<Ctrl>d
```

The *time* argument consists of the time of day and, if the date is not today, the date.

## Delaying Program Execution with batch and at

The following example shows how to use the **at** command to mail a happy birthday banner to login **emily** on her birthday:

```
$ at 8:15am Feb 27
banner happy birthday | mail emily
(ctrl)d
job 453400603.a at Wed Feb 27 08:15:00 1991
$
```

**B**

Notice that the **at** command, like the **batch** command, responds with the job number, date, and time.

If you decide you do not want to execute the commands currently waiting in a **batch** or **at** job queue, you can erase those jobs by using the **-r** option of the **at** command with the job number. The general format is

```
at -r jobnumber
```

Try erasing the previous **at** job for the happy birthday banner. Enter:

```
at -r 453400603.a
```

If you have forgotten the job number, the **at -l** command will give you a list of the current jobs in the **batch** or **at** queue, as the following screen example shows:

```
$ at -l
user = mylogin 168302040.a at Sat Nov 29 13:00:00 1990
user = mylogin 453400603.a at Wed Feb 27 08:15:00 1991
$
```

Notice that the system displays the job number and the time the job will run.

## Delaying Program Execution with batch and at

Using the **at** command, mail yourself the file **memo** at noon to tell you it is lunch time. Then try the **at** command with the **-l** option:

```
$ at 12:00pm
mail mylogin < memo
(Ctrl)d
job 263131754.a at Jun 30 12:00:00 1990
$
$ at -l
user = mylogin 263131754.a at Jun 30 12:00:00 1990
$
```

Figure B-2 summarizes the syntax and capabilities of the **at** command.

| <b>at</b> - executes commands at a specified time |                                                                                                                                                                                                                                                                                                                                                                  |                                        |
|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| <i>command</i>                                    | <i>options</i>                                                                                                                                                                                                                                                                                                                                                   | <i>arguments</i>                       |
| <b>at</b>                                         | <b>-r</b><br><b>-l</b>                                                                                                                                                                                                                                                                                                                                           | <i>time (date)</i><br><i>jobnumber</i> |
| <b>Description:</b>                               | at executes commands at the time specified. You can use between one and four digits, and am or pm to show the time. To specify the date, give a month name followed by the number for the day. You do not need to enter a date if you want your job to run the same day. See the <b>at(C)</b> manual page in the <i>XENIX Reference</i> for other default times. |                                        |
| <b>Options:</b>                                   | The <b>-r</b> option with the job number removes previously scheduled jobs.<br><br>The <b>-l</b> option (no arguments) reports the job number and status of all scheduled <b>at</b> and <b>batch</b> jobs.                                                                                                                                                       |                                        |
| <b>Remarks:</b>                                   | Examples of how to specify times and dates with the <b>at</b> command are as follows:<br><br><b>at 08:15am Feb 27</b><br><b>at 5:14pm Sept 24</b>                                                                                                                                                                                                                |                                        |

**Figure B-2** Summary of the **at** Command

---

## Permitting Users Access to Job Scheduling

**B**

Users are permitted to use **at**/**batch**, or **crontab** if their name appears in the files */usr/lib/cron/at.allow*, or */usr/lib/cron/cron.allow*, respectively. If the allow file for **at** or **cron** does not exist, the file */usr/lib/cron/at.deny* */usr/lib/cron/cron.deny*, respectively, are checked to determine if the user should be denied access to the appropriate command. If neither the deny or allow file exists, only *root* is allowed to submit a job. If only the deny file exists, global usage is permitted except for those users listed in the deny files. The allow/deny files for **at** and **cron** consist of one user name per line.

Global usage of the **crontab** command is permitted by the existence of an empty *cron.deny* file. *cron.deny* is checked only if *cron.allow* does not exist.

A history of all actions by *cron* can be recorded in */usr/lib/cron/log*. This logging occurs only if the variable **CRONLOG** is set to **YES** in */etc/default/cron*. By default this value is set to **NO** and no logging occurs. If logging should be turned on, be sure to check the size of the log file regularly.

---

## Managing cron Jobs with the queuedefs File

All jobs scheduled by **cron**, **at**, and **batch**, are executed at the proper time by the **cron** daemon. Normally, **cron** maintains two lists, or queues, of tasks waiting to be executed. One queue contains all tasks run by the users' **crontab** entries as well as the **at** utility. The second queue contains all jobs submitted using **batch**. Each queue has certain parameters assigned to it: the number of jobs in any one queue that may run simultaneously; the **nice**(C) value of the jobs (the value that determines priority in relation to other running processes); and when to retry a job if there are already the maximum number of jobs running.

---

### Note

**batch** is actually a script that calls **at** on the "b" queue. The system administrator can set up other queues with similar program interfaces.

---

However, **cron** is not limited to two queues. The system administrator can define up to 26 queues to manage different types of tasks. The **-q** flag to **at** is used to select which queue a job is submitted to. The characteristics of each queue are defined in the file */usr/lib/cron/queuedefs*.

By defining the **nice** value and number of concurrent jobs, the system administrator can manage how a queue of jobs affects system performance. Large number-crunching mathematical programs, for example, could be limited to one or two at a time. UUCP jobs could be limited to the number of available dial-out lines.

The queuedefs file contains one line for each queue definition. The format of each line is:

*x.njn<sup>n</sup>nw*

where:

- x* is the queue's letter (a through z, lower case only).
- nj* specifies the maximum number of jobs in the queue that may run at the same time.
- nn* specifies the **nice** value given to jobs in the queue when they are executed.
- nw* indicates how many seconds in the future a job should be scheduled to try again if the allowed number of running jobs has been reached.

Any of these three fields can be omitted and will then assume default values (100 maximum jobs, **nice** value of 2, and retry in 60 seconds). The default */usr/lib/cron/queuedefs* is:

```
a.4j1n
b.2j2n90w
```

This default configuration defines two queues: the first (a.4j1n) is used for all jobs submitted to **cron** and by the **at** command; the second queue (b.2j2n90w) is used for jobs submitted with the **batch** command or using **at** with the **-qb** option.

The **cron** and **at** queue (a.4j1n) can have up to four processes running at once. The **nice** value of each process is set to 1. If another job attempts to run while four are already executing, it will be rescheduled for one minute later.

The batch queue (defined by the second line, shown above) can have only two jobs running simultaneously. Its jobs receive a **nice** value of 2, and will be rescheduled for 90 seconds later if the maximum number of jobs is reached.

When rescheduling occurs, the following message is displayed on the console:

```
n queue max run limit reached date
rescheduled at job date
```

## Managing cron Jobs with the queuedefs File

Other queues can be defined by the system administrator by adding lines to the `/usr/lib/cron/queuedefs`. Before the new queues may be used, **cron** must be stopped and then restarted, either by rebooting the system, or killing and restarting the **cron** process itself. Jobs may be submitted to other queues using the **at** command with the **-qx** option, where *x* is the letter of the desired queue.

Job queues can be used to manage system resources such as printers, modems, or archive devices. For example, the following queuedefs entry might be used for communications programs on a system that has four modems:

```
m.4j0n300w
```

For this queue, there can be up to four communications tasks running, at normal priority (**nice** value of 0). If a fifth job is entered in the queue, it is scheduled for 5 minutes later, when a free modem might be available.

## Appendix C

# Using the System Console and Color Displays

---

Introduction C-1

Console Keyboard Type Selection C-2  
    Switching Keyboard Modes Manually C-2  
    Changing Modes Permanently C-3

Using the Console Screen Protection Feature C-4

Using Multiscreen C-5  
    Reducing the Number of Multiscreens C-6  
    Multiscreens and Multiple Video Adapters C-7

Changing Video Fonts C-8

Switching Video Adapters C-9

Controlling Color Displays with setcolor C-10  
    Changing the Foreground and Background Colors C-10  
    Changing Reverse Video Colors C-11  
    Changing the Screen Border Color C-11  
    Sounding the Keyboard Bell C-11  
    Resetting the Screen C-12



---

# Introduction

This appendix is concerned with utilities and features that affect the use of the system console and other color displays. Console displays are connected to a standard display adapter, while color terminals (such as Sun River terminals) are connected to the system by special adapters.

**C**

This chapter explains how to do the following:

- Set or change the console keyboard type (XT or AT) using **kbmode(ADM)**
- Protect the console screen from excessive wear by blanking it when not in use
- Use the **multiscreen(M)** facility to control multiple screens from a single display
- Change the font used on the screen display using **vidi(C)**
- Set the colors displayed on color screens using **setcolor(C)**.

If you wish to set up a serial console, refer to “Setting up a Serial Console” in the “Adding Terminals and Modems” chapter in this guide.

# Console Keyboard Type Selection

The operating system supports two keyboard modes: AT and XT. By default, the system is configured for use with an XT keyboard. AT mode supports the extended keyset found on an AT 101 or 102 key keyboard. An AT keyboard will work properly in XT mode, but the extended keyset is not accessible. You cannot configure an XT or other non-AT keyboard to work in AT mode. The system will not recognize keyboard input if this is done. The **kbmode**(ADM) utility is used to test and set the keyboard mode.

Some keyboards have an AT keyboard layout, but do not support AT mode. To test your keyboard to determine if it supports AT mode, invoke **kbmode** with the test option as follows:

### **kbmode test**

A sample session with **kbmode** in test mode is shown below, complete with user input in boldface:

```
kbmode test
Current keyboard mode is XT
Do you want to determine if your keyboard supports AT mode? y
During the test the keyboard will be put into AT mode.
You should then press the space bar two or three times.
Are you ready to start? y
Please hit the space bar now!

The keyboard has been returned to its default mode.
It supports AT mode.
#
```

The keyboard will be temporary initialized to AT mode.

## Switching Keyboard Modes Manually

The **kbmode** utility is also be used to set the mode. Use one of the following commands for switching to AT and PC/XT mode, respectively:

**kbmode at**

**kbmode xt**

### Changing Modes Permanently

To change the default mode permanently, the kernel parameter `KBTYPE` must be set to the proper keyboard. To change `KBTYPE`, run the **configure(ADM)** utility and select option 13, “Hardware Dependent Parameters.” Change the parameter value and relink the kernel as described in the “Reallocating Kernel Resources with configure” section of the “Tuning System Performance” chapter in this guide.

C

---

## Using the Console Screen Protection Feature

The console can be set up to blank after a certain number of seconds to protect the screen from excessive wear. (This is similar to a feature available with most terminals.) The kernel parameter TBLNK controls the console screen protection feature. The default behavior is not to blank the screen.

---

### *Note*

The screen protection feature is not available on XENIX-286 distributions.

---

To enable this feature, you must invoke the **sysadmsh** selection System→Configure→Kernel→Parameters and select category 6: “Multi-Screens,” and change the value of TBLNK to the number of seconds that the system should wait before blanking the screen. The kernel must then be relinked and booted for the new behavior to take effect. Use the **sysadmsh** System→Configure→Kernel→Rebuild selection to relink the kernel. See “Reallocating Kernel Resources with configure” in the “Tuning System Performance” chapter of this guide for complete instructions.

---

## Using Multiscreen

With Multiscreen, you can use your console as several terminals at one time. Pressing a simple key combination switches you from one screen to another, with each screen acting as an independent terminal.

Because each multiscreen is independent, you can log in and run programs on each screen. Because output from your programs is saved in a screen buffer, you see the most recent output for whichever screen you look at. If you stop output to one screen, as when you press the `<Ctrl>`s key combination, only that screen is affected.

The amount of memory in your computer determines the number of multiscreens available on your system. When you boot your system, the number of automatically-enabled multiscreens is displayed. Your machine can have up to twelve if your system has sufficient memory. To increase the number of multiscreens on your system, add to your system's memory; additional screens will be enabled automatically.

Although all of the multiscreens can be open and active at once, you see only one screen at a time. The selected multiscreen is like a terminal that is "connected" to the keyboard. Switching between screens is like moving to another terminal because each multiscreen has its own device file.

The multiple screen feature uses the `/dev/tty[01..12]` device files. These files provide character I/O between your system and your computer screen and keyboard.

To select any active screen, press `<Alt>-Fn`, where `Fn` is one of the function keys on your keyboard. Function keys are generally located across the top or down the far left side of the keyboard. `tty01` is the `<Alt><F1>` screen, `tty02` is the `<Alt><F2>` screen, `tty03` is `<Alt><F3>`, etc. For example, this key-stroke switches you to screen 6, corresponding to `/dev/tty06`:

`<Alt><F6>`

You can also rotate through the screens by pressing the Control and Print Screen key combination, `<Ctrl><PrtSc>` (using the `<Ctrl>` key and the `<PrtSc>` key). Use this combination to access screens for which you do not have function keys. For example, if you have twelve multiscreens enabled, but



## Using Multiscreen

your computer keyboard has only ten function keys, display screen eleven by pressing `<Alt><F10>` to get to screen 10, and then pressing `<Ctrl><PrtScr>` to rotate to screen 11. To access screen 12, press `<Ctrl><PrtScr>` again. Pressing `<Ctrl><PrtScr>` again rotates back to the first multiscreen, *tty01*.

Note that you can use `<Ctrl><Alt>` *function-key* combinations in addition to `<Alt>` *function-key* combinations to change multiscreens. This is especially useful in applications that reserve the `<Alt>` *function-key* combinations for their own use. This can be configured using the `mapkey(ADM)` utility.

For more information, refer to `multiscreen(M)` and `screen(HW)`.

## Reducing the Number of Multiscreens

If you have sufficient memory, the system is configured with 12 `multiscreen(M)` console screens by default. Although this does not significantly affect performance, you can reduce the number of screens if desired. To configure the system for fewer screens, follow the instructions found in “Reallocating Kernel Resources with `configure(ADM)`” in the “Tuning System Performance” chapter of this guide and use the following information:

1. Select category 6, Multiscreens.
2. The first parameter displayed is the `NSCRN` parameter. Enter the value corresponding to the number of screens you wish to configure.
3. Calculate the amount of screen memory in Kbytes (controlled by the `SCRNMEM` parameter) set aside for screens as follows:

*For 25-line displays:*

$$\text{SCRNMEM} = 10\text{K} + 4\text{K} * \text{NSCRN}$$

*For 43-line displays:*

$$\text{SCRNMEM} = 10\text{K} + 8\text{K} * \text{NSCRN}$$

4. Change the value of `SCRNMEM` to match the value you calculated in the previous step.

5. After leaving **configure** and relinking the kernel, you must then disable the unused **multiscreen** ttys. For example, if you reconfigured for 5 screens after having 12, you would enter the following command at the system prompt to disable ttys 6 through 12:

```
disable tty06 tty07 tty08 tty09 tty10 tty11 tty12
```

6. You should then reboot the system for the changes to take effect.

C

## Multiscreens and Multiple Video Adapters

Video adapters can be assigned to **multiscreens** dynamically. All start on the primary adapter, but any screen can be moved to the other video card with the **vidi(C)** command.

Valid adapter names are “mono”, “cga”, “ega”, and “vga”.

For example, if you have an EGA adapter as your primary adapter, and a MONO was secondary then any screen you wish to be on the MONO card can be moved with the following command.

```
vidi mono
```

This affects only the multiscreen that is currently in use.

---

# Changing Video Fonts

You can display the full range of characters on a display adapter by using the **vidi(C)** utility. Normally, if you have a console with a display adapter which has a character set defined in a ROM, you will be able to display only those characters defined in that ROM. In addition, the **mapchan** file for the console must correspond to the character set defined in the display adapter ROM to display the entire font set.

In addition to using **vidi** to override ROM, you can use it to define certain display fonts on some display adapters. For example, the VGA adapter will allow you to display fonts in the sizes 8x8, 8x14, and 8x16.

The **vidi** utility defines the font for one of these six character sets. The font definition files are available in the directory */usr/lib/vidi*. For more information, see the **vidi(C)** manual page.

---

## Switching Video Adapters

The **vidi** utility also allows the switching between different display adapters installed on your system. For example, to switch from the internal display on a portable to an external display attached to the portable, you enter:

**vidi external**

To switch to a VGA display connected to your system, you enter:

**vidi vga**

For more information, see the **vidi(C)** manual page.

A small black square containing a white letter 'C', representing a copyright symbol.

---

## Controlling Color Displays with `setcolor`

`setcolor(C)` is a simple utility that enables you to control the colors used on the display screen. (The `setcolor` command usually has no effect on monochrome displays or terminals.) Both foreground and background colors can be set independently in a range of 16 colors. `setcolor` can also set the reverse video and graphics character colors.

The following colors are available:

|         |            |          |        |
|---------|------------|----------|--------|
| blue    | magenta    | brown    | black  |
| lt_blue | lt_magenta | yellow   | gray   |
| cyan    | white      | green    | red    |
| lt_cyan | hi_white   | lt_green | lt_red |

To display these colors, simply invoke `setcolor` without options.

The following flags are available. In the arguments below, “color” is taken from the above list.

### Changing the Foreground and Background Colors

You can set both background and foreground colors with a single command as in the following example:

```
setcolor red white
```

This results in red characters on a white background. If only one color is specified, the foreground color is changed. To change only the background color, use the `-b` option, as in the following:

```
setcolor -b red
```

This simply changes the background color to red.

## Changing Reverse Video Colors

Reverse video normally inverts the foreground and background colors. **setcolor** allows you to set these independently. For example:

```
setcolor -r blue red
```

This command sets the foreground reverse video color to blue and the background to red.

C

## Changing the Screen Border Color

You can also change the color of the square border that defines the text region of the display:

```
setcolor -o green
```

The above example changes the border to green without affecting the rest of the display. Changing the border color does not work with all types of color displays.

## Sounding the Keyboard Bell

One of the less obvious functions of **setcolor** is to control the sound of the bell that is usually built into the display or keyboard. To change the bell tone, you must supply a pitch and duration. (Pitch is the period in microseconds, and duration is measured in fifths of a second.) When using this option, a control-G (bell) must be echoed to the screen for the command to work. For example:

```
setcolor -p 500 2
```

This command sets the bell to a high pitch of short duration. The higher the pitch number, the lower the sound generated. For example, this command sets the bell to a sustained low tone:

```
setcolor -p 7000 8
```

Note that each time <Ctrl>g is pressed, the bell will sound the tone most recently set.

## **Resetting the Screen**

The **-n** option returns the screen to “normal” white characters on black background.

# Index

---

## Special Characters

{ } (brackets), find program 4-10  
; (semicolon), find(C), command 4-10  
/ (slash), root filesystem 4-2  
1640, printer interface script 15-29  
19.2 K baud 14-9  
38.4 K baud 14-9

## A

accept(ADM), command 15-26  
Access time, locating files by 4-10  
Account  
  adding 10-2  
  changing user ID 10-11  
  comments 10-4  
  explained 10-1  
  group, changing 10-9  
  login group, changing 10-9  
  login name 10-3  
  removing 10-13  
  shell type 10-4  
Accounting, record of processes 5-26, 5-27  
acctcom(ADM), command 5-26, 5-27  
accton(ADM), command 5-26  
Adding  
  hard disk 7-1  
  printer 15-3  
  tape drive 16-2  
Address, serial board 17-7  
(ADM), reference section 1-10  
AD\_SCATTER parameter 8-11, 8-32  
Alt keys 19-11  
ANSI terminals 14-14  
Archiving  
  defined 16-1  
  floppy disk 16-15, 16-18  
  tape 16-11  
ASSERT error messages 11-55  
AT console keyboard 19-14  
AT console keyboard C-2

at(C), command  
  at.allow file B-9  
  at.deny file B-9  
  defined B-6  
-atime option, find(C), command 4-10  
AUTO keyword 14-8, 14-9  
Auto-answer modem configuration 14-32  
autoboot(ADM) 3-15  
Autologic APS-5 phototypesetter 15-29  
Autologout. *See* idleout(ADM), utility  
Automatic Call Unit (ACU) 11-5, 11-31, 11-32, 11-33, 14-29  
  Devices file 11-18  
  fixing problems with 19-63  
AUX port 15-12

## B

B1200, gettydefs flag 14-8  
Backspace defining (stty echoe) 14-13  
backup account 6-2  
Backup descriptor table 6-6  
backup(ADM), utility 6-3  
Backups  
  *See also* Filesystems  
  archiving 6-10  
  creating 6-1, 19-18  
  /etc/default/archive file 6-29  
  /etc/default/filesys file 6-29  
  getting a listing 6-18  
  incremental, explained 6-31  
  labeling 6-9  
  levels 6-7  
  log book 6-10  
  restoring a filesystem 6-24, 6-34, 19-18  
  restoring files or directories 6-20  
  root filesystem 6-33  
  rotating media 6-10  
  schedule 6-4  
  schedule examples 6-32  
Bad block 4-27  
Bad HZ value, error message 19-47  
Bad system name, error message 19-66  
Bad track

## Index

Bad track (*continued*)  
  boot problems 19-3  
  mapping with badtrk(ADM) 19-49  
badtrk(ADM), utility 7-13  
BANNERS variable 15-31  
Base address, tape drive conflict 19-55  
batch(C), command B-5  
Battery, clock 3-5  
Baud rate 14-2, 15-31  
  changing 14-12  
  error 11-11  
  modem 14-29  
  modem problems 19-30  
  setting in /etc/ttyS file 19-34  
Bidirectional port 11-38  
/bin directory 13-2  
BIOS  
  described 7-4  
  parameter table 7-9  
  shadowing 19-27  
Block  
  bad 4-27  
  defined 4-7  
  displaying 4-8, 4-9  
  duplicate 4-27  
  free-block list 4-28  
  per sector 13-10  
  size 13-9  
  structure 4-23  
block out of range, error message 19-24  
Boot  
  automatic 3-15  
  configuring 3-14, 3-15  
  display 3-12  
  steps 3-2  
/boot file, restoring 19-4  
/boot not found, error message 19-4  
Boot problems  
  bad track 19-3  
  corrupted super block 19-9  
  installation 19-2  
  missing system files 19-4, 19-5, 19-7,  
    19-8, 19-9  
  NO OS 19-9  
  RAM (Random Access Memory)  
    19-2  
  system hangs 19-8  
  system hangs at the letter Z 19-6  
Boot screen  
  memory recognition 19-26  
  port not recognized 19-38  
  tape drive recognition 19-54  
  tape drives 16-9  
Bootable floppy disk 17-3

boot(HW), command 3-14  
Bootstrap program 3-2, 3-14  
Bourne shell  
  .profile file 14-15  
  TERM assignment 14-14  
Break, login script 11-20  
Break key 1-13  
Buffers  
  allocating character list 8-22  
  allocating physical I/O buffers 8-21  
  boot display 17-5  
  buffer cache 8-8, 8-21  
  cache hit rate 8-20  
  defined 4-23, 8-1  
  parameters 8-20  
  tape size 8-21  
Bus, defined 17-1  
Bus card 17-1, 17-2  
Busmouse  
  card 18-8  
  installing 18-5

## C

(C), reference section 1-10  
C runtime library files 13-5  
C. (work) files 11-4, 11-7, 11-46, 11-56  
Cabling  
  DB-25 connector 14-4  
  DB-9 connector 14-4  
  planning 1-18  
  printer 15-3, 15-5  
  terminal 14-2, 14-4  
  unterminated wire problem 14-27  
CALLBACK option, Permissions file  
  11-27  
Call-out table 8-25  
Canceling, print request 15-20  
CANNOT ACCESS FILE, error  
  message 19-72  
Cannot access special file, error  
  message 19-19  
Cannot create, error message 19-37  
Cannot load floating-point emulator,  
  error message 19-7  
Cannot open /dev/hd0root, error  
  message 19-21  
Cannot open: device busy, error  
  message 14-24  
Cannot open /dev/rct0, error message  
  19-57

- Cannot set up /dev directory on new device, error message 19-26
- Can't exec /etc/inir, error message 19-8
- Can't exec /etc/init, error message 19-8
- Carrier detection (CD) 14-32
- Carrier Detection (CD), light, modem problems 19-31
- Cartridge tape, *See* Tape
- Cartridge tape drive, *See also* Tape drive problems
- Character buffers parameters 8-22
- Chat script. *See* Login script
- Checking, filesystems 4-21
- chgrp(C), command 5-15
- chmod(C), command 4-11, 5-10, 5-12
- Choosing a password 5-4
- chown(C), command 5-14
- Circuit boards 17-1
- Class, printer 15-21, 15-23
- Cleaning filesystem. *See* Filesystem
- Clear to Send (CTS) 15-36
- Clock 3-5
  - parameters 8-25
- CMASK parameter 8-23
- cmos(HW) 17-9
- cmp(C), utility 6-21
- Color displays
  - changing display C-10
  - color list C-10
  - foreground and background C-10
  - reverse video C-11
  - screen border C-11
- COLUMNS setting 14-23
- COM ports
  - cabling 14-27, 14-28
  - COM3 and COM4 17-7
  - dialing in 14-33
  - interrupt assignments 14-30, 17-7
  - modem use 14-24
  - printer 15-5
- Commands
  - editing 1-13
  - entering 1-13
  - locating 1-10
  - location 13-2, 13-5
  - printer 15-16
  - UUCP 11-3, 11-4
- COMMANDS option, Permissions file 11-27, 11-28
- Computer, bus 17-1
- configure(ADM), utility
  - changing kernel parameters 8-1
  - command-line interface 8-5
  - defined 8-3
  - configure(ADM), utility (*continued*) menu 8-4
- Configuring
  - hard disk 7-1, 7-4
  - modem *See* Modem
  - printer 15-7
  - serial port 17-7
- Connect failed: no devices available, error message 19-29
- Console
  - See also* System
  - changing colors C-10
  - color list C-10
  - device file 13-10
  - device name 13-3
  - keyboard type C-2
  - multiscreen C-5
  - resetting screen C-12
  - screen border color C-11
  - screen protection C-4
  - serial 14-16
  - switching fonts C-8
  - video adapters C-7
- Console problems
  - keyboard lockup 19-11, 19-12
  - wrong keyboard type 19-14
- Context indicator (sysadmsh) 2-3
- Controller, ST506 7-7
- copy(C), command 7-20
- Copying
  - floppy disks 16-16, 16-18
  - remote files 11-4
- core file 1-13, 4-13
- Corruption
  - filesystem 4-21
  - repairing 4-26
- (CP), reference section 1-10
- Crash. *See* System
- Creating
  - Emergency Boot Floppy 19-3
  - filesystem 7-15
  - group 10-7
  - User 10-2
- crlmap, printer interface script 15-29
- cron may not be running, error message 19-45
- cron(C), daemon 19-45
  - clearing log files 4-15
  - clearing log files B-4
  - cron.allow file B-9
  - cron.deny file B-9
  - crontabs file 4-15
  - log file 13-8
  - reducing system load 8-13

## Index

cron(C), daemon 19-45 (*continued*)  
    UUCP 11-4, 11-7, 11-44  
cron(C), dameon  
    log file B-4  
crontabs file B-4  
crontabs file B-9  
crontabs file, sample 4-15  
CTBUFSIZE parameter 8-21  
ct(C), command 11-3  
Ctrl-q key, unlocking keyboard 19-12  
Ctrl-s key, stopping screen scrolling  
    19-12  
CTS (Clear to Send) 15-36  
cu: dir permission denied 14-29  
cu(C), command 11-3, 19-63, 19-64,  
    19-65  
    dialing out 14-31, 14-34  
    exiting 14-26  
    login sequence 11-20  
    syntax 14-25  
    testing a modem 14-29  
custom(ADM), utility 8-3

## D

D. (data) files 11-4, 11-46, 11-47, 11-56  
Daemons  
    cron(C) B-10  
    cron(C) B-4  
    cron(C) B-9  
    Irwin drive(mcd daemon) 16-7  
    printer 15-10, 15-18, 15-20  
    UUCP 11-6, 11-7, 11-22, 11-44,  
        11-46  
DASI terminal 15-29  
Data bit settings 19-30, 19-34  
Data bits 14-2  
Data block, defined 4-23  
Data Communications Equipment  
    (DCE) 14-28, 15-36  
Data storage 4-22, 4-23  
Data Terminal Equipment (DTE) 14-28,  
    15-36  
Data Terminal Ready (DTR)  
    auto-answer modem 14-32  
    defined 14-26  
    modem problems 19-31, 19-32  
Date 3-5  
DB-25 connector 14-4  
DB-9 connector 14-4  
DCE (Data Communications  
    Equipment) 15-36

ddate log file 13-7  
dd(C), command 16-16, 16-18  
Defaults  
    /etc/default directory 13-4  
    modem 14-29  
    printer 15-22  
DEFBOOTSTR option 3-14  
Defective tracks 7-14  
Delete key 1-13  
Description line (sysadmsh) 2-3  
/dev directory 13-2  
    security 5-12  
Device  
    cannot access 19-64  
    defined 11-31  
    dialout modem problems 19-29  
    directories 13-2  
    filenames 13-9  
    files, serial cards 17-8  
    fixing file permissions 19-64  
    locked 19-69  
    nodes 13-3  
    none available 19-29, 19-64, 19-70  
    printer 15-2  
Device kernel parameters 8-31  
devices, hardwired 13-10  
Devices file  
    baud rate 11-10  
    dial-out entries 11-31, 14-34  
    error 11-57  
    example 11-8  
    format 11-31  
    LAN switch 11-19  
    Local Area Network (LAN) 11-32  
    matching Systems file 11-6  
    modem 11-8, 11-10  
    modem installation 14-29  
    speed field 11-33  
    testing modem 11-14  
    type field 11-32  
    UUCP configuration 11-5  
    write permissions 14-29  
df(C), program 4-7, 13-9  
Dialcodes file 11-18, 11-39  
Dialer binaries  
    defined 14-34  
    example usage 11-34  
    /usr/lib/uucp directory 11-8  
Dialers file 11-39  
    dialer binaries 14-34  
    dialer programs 11-34  
    dialer types 11-33  
    dialer-token entries 11-37  
    entries 11-34

- Dialers file 11-39 (*continued*)
    - error 11-60
    - modem 11-35
  - Dialer-token 11-32, 11-35, 11-37, 11-59
  - dialHA12 dialer binary 14-34
  - dialHA24 dialer binary 14-29
  - dialHA24 dialer binary 14-34
  - dialHA96V dialer binary 14-34
  - Dial-in line, password 5-20
  - Dial-in modem. *See* Modem
  - Dial-in site 11-2, 11-12, 11-15
  - Dialing out, command 14-31
  - dialMUL dialer binary 14-34
  - Dial-out modem. *See* Modem
  - Dial-out site 11-2
  - dialTBIT dialer binary 14-34
  - dialVA3450 dialer binary 14-34
  - dialVA96 dialer binary 14-34
  - diff(C), utility 6-21
  - Dip switches 17-2
  - dir permission denied 14-29
  - Direct, keyword 11-32
  - Direct line 14-29
  - Direct wire, UUCP 11-41
  - Directory
    - finding 4-9, 4-10, 4-11
    - lost+found 4-27
    - permissions 5-4, 5-7
    - root 13-2
    - size 1-16, 4-8, 4-18
    - size consideration 1-16
  - disable(C), command 13-10, 14-3
    - modem use 14-28
    - printer) 15-20
    - terminals 14-15
  - Disabling
    - port 15-3
    - serial line 11-42
  - Disk
    - See also* Hard disk
    - adding 4-20
    - bad track table 19-49
    - bad tracks 7-15
    - block number and size 13-10
    - changing layout 4-20
    - configuring 4-20, 7-1, 7-3
    - cylinder 7-12
    - errors 13-5
    - fragmentation 4-7, 4-17, 19-18
    - free space 4-6
    - Free space 4-7
    - gap and block numbers 13-10
    - gap number 13-10
    - installing 7-6
  - Disk (*continued*)
    - nonstandard 7-9
    - repairing 4-21
    - security 5-3
    - size 7-9
    - usage 4-8, 8-8
    - user accounts 7-19
  - Disk parameters 8-20
  - diskcp(C), command 16-16, 16-17
  - Display
    - adapter, characters C-8
    - adapter, switching C-9
    - console 1-14
    - scan window, sysadmsh(ADM) 2-22
    - sysadmsh(ADM), utility 2-3
  - divvy(ADM), command
    - error during installation 19-22, 19-26
    - explained 7-15
    - restoring root filesystem 19-20
  - dkinit(ADM), command 7-9
  - DMA (Direct Memory Access) channel,
    - tape drive conflict 19-55, 19-56
  - DMA requests, allowing simultaneous
    - 8-31
  - DMAEXCL parameter 8-31
  - DOS
    - coexistence 9-1
    - device names 9-12
    - files and directories 9-12
    - hard disk devices 9-7
    - partition 7-11
    - partition, removing 9-10
    - switching from XENIX 9-5
    - two hard disks 9-9
    - utilities 9-1, 9-11
    - XENIX partition 9-8
  - (DOS), reference section 1-10
  - Driver, vendor-supplied 17-7
  - DSTFLAG parameter 8-25
  - DTE (Data Terminal Equipment) 15-36
  - DTR (Data Terminal Ready), serial
    - printer 15-3
  - du(C), command 4-8, 19-16
  - dumb, printer interface script 15-29
  - Dummy user account 5-21
  - Duplex 14-2
- ## E
- EAGAIN error 8-23
  - EBUSY error 8-23
  - Echo problems, modem 19-31

## Index

- echoe, stty(C) flag 14-13
- ECON-80 mode 14-23
- Editing, command line 1-13
- Embedded controller 7-5
- Emergency Boot Floppy 19-4, 19-5, 19-7, 19-8, 19-9
  - contents 19-3
  - restoring root filesystem 19-18
  - restoring system files 19-5, 19-7, 19-8
  - system corruption 19-4
- emulator, printer interface script 15-29
- enable(C), command 13-10, 14-4
  - auto-answer modem 14-33
  - printer 15-20
- Enter key 1-13
- Enter new time ([yymmdd]hhmm):
  - message 3-5
- env(C), command 14-15
- environ(M) 6-4
- Environment variable
  - initialization A-5
  - PATH 8-13
  - SA\_EDITOR 2-21, 6-4
  - SA\_MAIL 2-21
  - SA\_PRINT 2-21
  - TERM 14-13
  - TZ (timezone) 8-25
- Epson serial or parallel printer 15-29
- Erase key 1-13
- Error
  - checking (fsdb), disabling 19-24
  - message. *See* Error message
- Error Correction Code (ECC) 16-14
- Escape, login script 11-21
- Escape key 1-13
- ESDI
  - configuration 7-3
  - disk controller 7-8
  - root disk 7-2
- /etc directory 13-3
- /etc/cshrc A-5
- /etc/ddate file 4-14
- /etc/default directory 3-14, 13-4, 16-9
- /etc/default/archive file 6-29
- /etc/default/boot 14-16
- /etc/default/cron file B-9
- /etc/default/filesys file 5-22, 6-29
- /etc/default/filesys file, example 4-4
- /etc/default/format, file 16-16
- /etc/default/idleout file 5-19
- /etc/default/login file, fixing HZ value 19-47
- /etc/default/lpd 15-31
- /etc/default/mcconfig 16-6
- /etc/default/mkuser file 7-19
- /etc/default/msdos file 9-11
- /etc/default/msdos files 9-12
- /etc/default/passwd file 5-16
- /etc/default/su file 5-25
- /etc/default/tar file 16-9
- /etc/d\_passwd file 5-20
- /etc/emulator file, restoring 19-7
- /etc/gettydefs file
  - changing 14-9
- /etc/group file, modifying 10-7
- /etc/init file 19-8
- /etc/init file, restoring 19-8
- /etc/login file, restoring 19-9
- /etc/motd file 4-6
- /etc/motd file A-6
- /etc/msscreencap file 14-20
- /etc/passwd file 7-21, 10-5, 10-11
- /etc/profile A-5
- /etc/rc scripts
  - execution 3-3, 3-4
  - explained 13-3
  - Micnet network startup 12-16
  - modifying A-2
  - starting cron daemon 19-45
  - starting lpsched daemon 19-46
- /etc/rc.d directory A-1
- /etc/rc.d directory A-3
- /etc/rc.d/6/kinit file 5-27
- /etc/shutdown program 4-25
- /etc/systemid file 12-2, 12-16, 14-7, 19-73
- /etc/termcap file 14-13, 15-12
- /etc/ttys file
  - modem lines 14-28
  - relationship to inittab A-1
  - setting baud rate in 19-34
  - terminal lines 14-3
- /etc/ttytype 14-15
- /etc/wtmp file 4-14, 5-24
- /etc/wtmp file B-4
- EVDEVS parameter 8-31
- EVDEVSPERQ parameter 8-31
- Event queue kernel parameters 8-31
- EVQUEUES parameter 8-31
- exec option, find program 4-10, 4-12
- Execute permission 5-8
- Executing, recursive commands 4-12
- Exiting, shell 1-13
- Expanded memory 17-6
- Expanding, system 1-18
- Expansion card 17-7
- Expansion slots 17-3
- Expect string 11-19, 11-20

Export 14-14  
 expr(C), command 14-22  
 EXTA 14-9  
 EXTB 14-9  
 Extended memory 17-6

## F

-f (follow) option, tail program 4-15  
 (F), reference section 1-10  
 fdisk utility, XENIX and DOS versions  
   9-1  
 fdisk(ADM), command  
   activating partition 9-4  
   creating partition 7-11, 9-3  
 File  
   archiving 16-1, 16-11, 16-18  
   backups. *See* Filesystem  
   clearing log files 4-14, 4-15  
   /etc/cshrc A-5  
   /etc/motd A-6  
   /etc/profile A-5  
   /etc/rc.d scripts A-2  
   finding. *See* find command  
   last access time 4-10  
   locating 19-17  
   log files B-4  
   .login A-5  
   name *See* Filename  
   ownership 5-14, 5-15, 5-21  
   parameters 8-22  
   permissions. *See* Permission  
   process accounting 4-14  
   .profile A-5  
   recursive removal 4-13  
   repairing 4-26  
   restoring 6-20  
   restoring critical system files 19-3  
   size 4-8  
   structures 4-22  
   system initialization A-1  
   transfer, UUCP 11-1, 11-5, 11-7  
   transferring ownership 4-12  
   UUCP log files 4-14  
 File creation mask 5-11  
 File table, allocating open entries 8-22  
 File table overflow, error message 8-6,  
   8-23, 19-47  
 File transfer, UUCP background dae-  
   mons 11-6  
 Filename, device special files 13-9  
 Filesystem

Filesystem (*continued*)  
*See also* Backups  
 adding 4-20, 7-18  
 backups  
   backup account 6-2  
   contents 6-18  
   defined 6-1  
   device list 6-29  
   filesystem list 6-29  
   frequency 6-2  
   media labeling 6-17  
   media storage 6-11  
   restoring 6-20  
   scheduling 6-11, 6-12  
 checking 4-25, 4-28  
 cleaning 3-3  
 corruption 4-21, 4-24  
 creating 4-17, 7-15  
 defined 4-2  
 expanding 4-20  
 floppy disk 16-19, 16-21  
 free space 4-6, 4-7  
 maintaining 4-17  
 mounting 4-3, 7-19  
 parameters 8-22  
 problems 19-4, 19-15, 19-16, 19-18,  
   19-21, 19-22, 19-24, 19-26  
 removing 4-17, 4-25  
 reorganizing 8-11  
 repairing 4-21, 4-25, 4-26  
 restoring 6-24  
 root 4-2, 4-26  
 size 1-16  
 space, maintaining 1-16  
 structures 4-22, 4-23  
 sysadmsh(ADM) options 7-18  
 unmounting 7-19  
 user mount privileges 5-22  
 find(C), command 4-9, 4-10  
   executing shell commands 4-10, 4-12  
   options 4-11, 4-12, 4-13  
 Finding files. *See* find command  
 finger(C), command 5-21  
 fixperm(ADM) command, checking  
   UUCP file permissions 19-72  
 Floating-point emulator error 19-7  
 Floating-Point Unit (FPU) 3-13  
 Floppy disk  
   backups 6-11  
   blocks 13-9  
   booting 19-4  
   copying 16-16  
   device names 13-3  
   filesystem 16-19, 16-21

## Index

Floppy disk (*continued*)  
  formatting 16-15  
  gap and block numbers 13-10  
  Micnet file 12-13  
  security 5-3, 5-4  
  using 16-15, 16-18  
Floppy tape drive, *See also* Tape drive problems  
Flow control using RTS/CTS 15-36  
Font  
  changing display C-8  
  definition files C-8  
Format, floppy disk 16-15  
Formatting, tape 16-13  
Fragmentation 4-7, 4-17  
Free list 4-28  
Free space 4-6, 4-7, 4-20  
Free-block list 4-28  
Front-end accounts 5-32  
fsave(ADM), utility 6-3  
fsck(ADM), command  
  abnormal exit 19-21  
  boot option 3-3  
  defined 4-21, 4-25, 4-26  
  failed mount command 19-24  
  gap and block numbers 13-10  
  phases 4-27  
  Proceed with cleaning 3-3  
  quits at size check 19-21  
  restoring root filesystem 19-18  
  scratch file 19-19  
  single-user mode 4-25, 19-51  
  wrong ISIZE and FSIZE values 19-21  
fsdb(ADM), command 4-27, 19-21, 19-23, 19-24  
FSIZE value 19-21, 19-22, 19-23  
fsphoto(ADM), utility 6-3  
fssize parameter 19-20

## G

Gap number 13-10  
Garbage or loose cable 19-33  
getty process 14-4  
getty(C), command, -c (check) option 14-11  
gettydefs file  
  alternate login programs 14-9  
  autologin 14-9  
  changing 14-9  
  explained 14-7  
  EXTA and EXTB entries 14-9

gettydefs file (*continued*)  
  relationship to inittab A-1  
getty(M) 14-6  
Ground 14-2, 17-2  
Group  
  access 10-8  
  creating 10-7  
  defined 10-7  
  ID 10-3, 10-7  
  ID, changing 10-9  
  name 10-3, 10-7  
  number 10-3  
-group option, find program 4-10  
Group password 5-21  
Group permissions. *See* Permission  
grpcheck(C), utility 10-8  
Guest, UUCP transaction 11-8

## H

haltsys(ADM), command 3-10  
Hard disk  
  *See also* Disk  
  adding 7-1  
  blocks 13-9  
  configuring 7-3, 7-4  
  device names 13-3  
  fdisk(ADM) utility 9-3  
  gap and block numbers 13-10  
  heads 13-10  
  installing 7-6  
  mkdev(ADM), command 7-3  
  nonstandard 7-9  
  non-standard 9-14  
  partition table 9-5  
  partitions 7-11, 9-4  
  removing a partition 9-10  
  size 7-9  
  switch settings 7-4  
  tracks 9-3  
  user accounts 7-19  
Hard disk problems  
  bad track table 19-49  
  checking with fsck 19-7, 19-8, 19-10  
  checking with fsck(ADM) 19-19  
  detecting flaws 19-50  
  mount unsuccessful 19-19  
  mounting and unmounting 19-7  
  reducing disk fragmentation 19-18  
Hardware  
  displaying 3-12, 3-13  
  removing 17-6

**Hardware failure**  
 defective printer 19-35, 19-38  
 modem port problems 19-29  
 non-functional terminal 19-58  
 super block damage 19-21  
 system does not boot 19-3  
 system panic 19-50  
 tape controller card 19-55, 19-57  
 terminal, hung 19-59, 19-60  
**Hardware-dependent parameters** 8-31  
**Hardwired devices** 13-10  
**Hash queues, increasing with system buffers** 8-21  
**Hayes (and compatible) modem configuring** 11-10  
**Hayes (and compatible) modem configuring** 14-26  
 switch settings 14-25  
**Hayes (and compatible) modem testing** 11-13  
**hd(C), program** 4-18  
**hdinstall(ADM), utility** 8-4  
**Header files directory** 13-5  
**Hierarchy, sysadmsh(ADM), utility** 2-22  
**Highlighting, sysadmsh** 2-5  
**Hog factor.** *See* Use factor  
**Home directory, removing** 10-13  
**Host, UUCP transaction** 11-8  
**Host adapter**  
 SCSI hard disk 7-4  
 SCSI tape drive 16-8, 19-56  
**hp, printer interface script** 15-29  
**hp2631a line printer** 15-29  
**Hung process** 1-17  
**HUPCL, gettydefs flag** 14-7  
**(HW), reference section** 1-10  
**hwconfig(C), command** 3-13, 15-5, 15-6, 16-9, 18-8, 19-54  
**HZ value, fixing in /etc/default/login file** 19-47

## I

**IBM Proprinter XL** 15-29  
**ID number** 7-4  
**ID number, SCSI tape drive** 19-56  
**idleout(ADM), utility** 5-4, 5-19  
**IDLETIME variable** 5-19  
**Incremental backups** 6-31  
**init program** 19-8  
**Initialization files** A-1, A-2, A-4, A-6

**Inode table, allocating entries** 8-22  
**Inode table overflow, error message** 8-6, 19-48  
**Inodes**  
 active 19-48  
 allocation 19-21  
 available 19-23  
 clearing 4-27  
 defined 4-23  
 directory 4-18  
 filesystem problems 19-15  
 null numbers 4-18  
 reconfiguring with mkfs(ADM) 19-16  
 removing extra (shadow) numbers 4-18  
 shadow 4-18  
 structure 4-23  
**Inodes parameters** 8-22  
**Installation** 19-3, 19-26  
**Installing** 19-2  
 bus card 17-2, 17-3, 17-7  
 cable connections 14-2  
 COM ports 17-8  
 Emergency Boot Floppy 19-3  
 hard disk 7-3, 7-6  
 memory 17-5  
 modem 14-24, 14-27  
 mouse 18-2, 18-5, 18-6  
 multiport card 17-7  
 planning system 1-18  
 printer 15-3  
 pseudo-tyts 14-20  
 tape drive 16-2  
 terminal 14-2  
 terminals 14-4  
**Intel inboard** 3-14  
**Intelligent serial board** 15-6  
**Interface**  
 printer 15-9, 15-10  
**Interface programs** 15-29  
**Internal memory, adding** 17-5  
**Internal modem** 14-27  
**Interrupt, key** 1-13  
**Interrupt vector**  
 boot display 3-12  
 COM ports 14-30, 17-7  
 conflicts 14-27  
 mouse 18-5  
 parallel port 15-3  
 printer 15-6  
 printer port conflict 19-36  
 tape drive conflict 19-55, 19-57  
**Invoking programs, users** 1-16

## Index

IRQ (physical interrupt). *See* Interrupt vector  
Irwin  
  mcconfig file 16-6  
  mcd daemon 16-7  
  problems 19-56  
  tape commands 16-13  
ISIZE value 19-21, 19-22, 19-23  
IXANY, gettydefs flag 14-7

## J

Jobs. *See* Process  
Jumper settings, tape drives 19-56  
Jumpers, bus card 17-2

## K

kbmode(ADM), command C-2  
KB\_MOUSE parameter 8-32  
KBTYPE kernel parameter C-3  
KBTYPE parameter 8-31  
Kernel  
  mouse driver 18-6  
  relinking 8-4  
kernel PANIC: Trap E in system, error message 19-51  
Kernel parameters  
  *See also* System  
  additional memory 8-10  
  buffers 8-20, 8-22  
  changing 8-1, 8-3  
  clock 8-25  
  defined 8-1  
  devices 8-31  
  disks 8-20  
  error messages 8-6  
  event queues 8-31  
  files 8-22  
  filesystems 8-22  
  inodes 8-22  
  memory management 8-23  
  message queues 8-26  
  multiscreens 8-25  
  processes 8-23  
  semaphores 8-27  
  shared data 8-28  
  streams data 8-28  
  swapping 8-23

Kernel parameters (*continued*)  
  system name 8-28  
  system performance 8-7  
  tradeoffs 8-7  
Kernel problems 19-43  
  backing up 19-13  
  bad HZ value 19-47  
  booting from alternate kernel files 19-5  
  inconsistencies causing system panic 19-50  
  inode table overflow 19-48  
  keyboard lockup patch 19-13  
  reconfiguring 19-42, 19-49  
  serializing 19-48  
  warning out of inodes on /dev 19-15  
Kernel resources. *See* System resources  
Keyboard  
  AT mode C-2  
  bell, sounding C-11  
  default mode C-3  
  end input 1-13  
  input, no response 19-11  
  lockup 19-11, 19-12, 19-43, 19-44  
  selecting types C-2  
  substituting mouse 18-10  
  switching modes C-2  
  test with modem 14-30  
  type, console 19-14  
  XENIX keystrokes 1-13  
Keystrokes, XENIX 1-13  
kill command, explaining to users 1-17  
Kill key 1-13  
Korn shell, TERM assignment 14-14

## L

LAN (Local Area Network)  
  planning 1-18  
last(C), command 5-24  
l(C), command 5-7, 5-14  
LCK (lock) files 11-8, 11-14, 11-46, 11-55, 11-60  
Level 0 backup 6-7, 6-11  
/lib directory 13-5  
Link Kit (LINK package) 8-3  
link\_xenix(ADM), utility 8-4  
LOADXENIX 3-15  
Local Area Network (LAN)  
  UUCP 11-19, 11-31, 11-32  
Local printing: lprint(C) 15-12  
Locating files. *See* find command

- lock(C), utility 5-3
  - Locked
    - device 19-69
    - keyboard. *See* Keyboard
  - Log book 1-2, 1-4, 6-10
  - .Log directory, UUCP programs 19-69
  - Log files
    - clearing 4-14, 4-15
    - clearing system files 19-17
    - cron B-4
    - cron log file 13-8
    - directories 13-7
    - error messages 13-5
    - /etc/ddate 13-7
    - /etc/ddate file 4-14
    - /etc/wtmp 13-7
    - /etc/wtmp B-4
    - lp B-4
    - Micnet 12-18
    - Micnet log file 13-8
    - monitoring 4-15
    - /usr/adm/messages 13-7
    - /usr/adm/messages B-4
    - /usr/adm/pacct 13-7
    - /usr/adm/sulog 13-8
    - /usr/adm/sulog file 4-14
    - UUCP 11-4
    - UUCP log files 13-8
    - UUCP status 19-69
  - Logical Unit Number (LUN), SCSI hard disk 7-4
  - Login
    - alternate, gettydefs file 14-9
    - creating dial-in accounts 11-12
    - directory size 4-18
    - ending session 1-13
    - group 10-9
    - Micnet network 12-12, 12-13
    - multiscreen 1-14
    - name 10-3
    - problems 19-9, 19-58, 19-65
    - procedure 1-16
    - prompt 3-7
    - security 11-40
    - sequence, UUCP 11-5, 11-6, 11-8
    - system boot 3-4
  - LOGIN FAILED, error message 19-70
  - Login script (chat script) 11-16, 11-19, 11-20, 11-21
  - login sessions, monitoring 5-24
  - login(M) program, restoring 19-9
  - LOGNAME entry, Permissions file 11-23
  - lost+found directory 4-27
  - lp\* commands 15-17
  - lp log file B-4
  - lp0, lp1, lp2 15-3
  - lpadmin(ADM), command 15-10
  - LPDEST shell variable 15-22
  - lpmove(ADM), command 15-26
  - lpsched(ADM), daemon 15-20
    - print problems 19-45
    - starting 15-10, 15-18, 19-37, 19-46
  - LQP-40 printer 15-29
  - LUN (Logical Unit Number), SCSI tape drive 19-56
  - LUN. *See* Logical Unit Number
- ## M
- MACHINE entry
    - OTHER option 11-30
    - Permissions file 11-23
  - Machine name 6-4
  - Mail, /usr/spool directory 13-5
  - mail, Micnet network command 12-1
  - mail program, defined 4-9
  - Mailbox, removing 10-15
  - mail(C), command
    - sending to remote sites 11-1
    - testing UUCP 11-41
  - maliases file 11-40
  - Man pages, *See* Manual pages
  - Manual pages 1-10
  - mapkey(M) 1-15
  - Master tty
    - create example 14-18
    - defined 14-17
  - Masterboot block 9-14
  - MAXBUF parameter 8-21
  - Maximum screens and pseudo-ttys 14-18
  - MAXSEPGCNT parameter 8-30
  - MAXUPRC parameter 19-43
  - MAXWEEKS variable 5-16, 5-17
  - mcconfig file (irwin tape) 16-6
  - mcdemon process 16-7
  - Media
    - administering 1-18
    - types 16-1
  - Memory
    - BIOS shadowing 19-27
    - expanded 17-6
    - extended 17-6
    - installing 17-5
    - installing additional 8-8, 8-10

## Index

### Memory (*continued*)

- management 8-23
- parity errors 17-6
- requirements for XENIX 19-26

### Menu, sysadmsh(ADM), utility 2-3, 2-4, 2-5, 2-22

### Message, system-wide 3-9

### Message of the day file. *See* /etc/motd file

### Message queue parameters 8-26

### messages file 15-5, 15-6

### messages log file 13-7

### Micnet network

- activity records 4-14
- alias 12-7, 12-13
- aliases, creating 12-8
- checking 12-17
- /etc/systemid file 12-2, 12-16
- forward alias 12-7
- group alias 12-8
- handshake message 12-18
- install option 12-9
- LOG file 12-18
- log file 13-8
- machine, name 12-2
- machine name 12-10
- mail command 12-1
- modifying 12-20
- netutil program 12-2, 12-9
- planning 1-18, 12-2
- propagating 12-9, 12-15
- restore option 12-9, 12-15
- restoring 12-15
- save option 12-9, 12-13
- serial line 12-4, 12-11
- standard alias 12-7
- star topology 12-2
- start option 12-16
- startup procedure 12-16
- stop option 12-20
- testing 12-17
- topology 12-2, 12-3, 12-9
- transmission speed 12-6, 12-11
- UUCP links 11-40

### Mini-cartridge

- formatting 16-13
- sizes 16-10
- tape drives 16-6

### MINWEEKS variable 5-16, 5-17

### mkdev(ADM), command

- fd 16-19
- fs 7-18
- hd 7-3, 7-6
- lp 15-3, 15-7

### mkdev(ADM), command (*continued*)

- mouse 18-3, 18-5, 18-6
- serial 17-7
- tape 16-2

### mkfs(ADM), command

- device names 13-9
- gap and block numbers 13-10
- reconfiguring free inodes 19-15

### mkuser(ADM), command 10-2

### mnt directory 13-5, 16-19

### mnt(C), command 4-4, 5-22, 7-19

### Mode

- changing 1-13
- single user 3-4
- system maintenance 3-4

### Modem

- auto-answer 14-32
- checking baud rate 19-30
- checking /etc/gettydefs file 19-34
- checking modem cable 19-28
- commands 14-26
- configuring 14-25
- dialers 14-34
- Dialers file 11-8, 11-35
- dial-in 11-2, 14-28, 14-31, 14-33
- dialing configuration 14-25
- dialing in 19-32, 19-33, 19-34
- dialing out 19-28, 19-29, 19-30, 19-31
- dial-out 11-2, 14-28, 14-31, 14-34
- disabling control line 11-42
- Hayes 2400 dialer 11-10
- Hayes (and compatible), *See* Hayes incompatible 19-33
- installing 14-24, 14-25, 14-27
- internal 14-27
- local network switch 11-37
- login sequence 11-20
- null modem 14-28
- pin connections, cabling 14-27
- planning 1-18
- problems in UUCP 19-70
- supported 11-2
- switch settings 14-25
- telephone line 11-1
- testing 11-13, 14-29
- testing phone line 19-29
- Trailblazer 14-26
- tty identification 17-8
- UUCP 11-10, 14-28
- variable baud rate 11-10
- volume 14-26

### Monochrome card 15-3

### Motherboard 17-1, 17-9

- Mount on /mnt failed, error message 19-26
  - Mount point 7-18
  - Mount table 8-23
  - mount(ADM), command 4-3, 16-21
    - cannot access special file, error message 19-19
    - device names 13-9
    - failed 19-24
    - floppy filesystem 19-4
    - hard disk root filesystem 19-7, 19-19
  - Mounting
    - defined 4-3
    - directory 13-5
    - filesystem 4-4, 7-18
    - floppy disk 16-21
  - Mounting filesystems, user privileges 5-22
  - Mouse
    - busmouse 18-5
    - installing 18-2, 18-3, 18-5, 18-6
    - multiscreens 18-9
    - problems 18-8
    - removing 18-8
    - restricted configurations 18-2
    - sharing 18-9
    - testing 18-7
    - tty 18-9
    - using 18-9, 18-10
  - mscreen(M), utility
    - configuring 14-17
    - example 14-22
    - mscreencap file 14-20
    - NSPTTYS 14-18
    - troubleshooting 14-20
    - tuning 14-21
  - MSGMAP parameter 8-26
  - MSGMAX parameter 8-26
  - MSGMNB parameter 8-26
  - MSGMNI parameter 8-26
  - MSGSEG parameter 8-26
  - MSGSSZ parameter 8-26
  - MSGTQL parameter 8-26
  - Multiplexer links 8-29
  - Multiport card 14-24, 15-6, 17-7
  - Multiscreen
    - boot display 17-5
    - console C-5
    - defined 1-14
    - example 1-14
    - kernel parameters 8-25
    - keyboard lockup 19-11
    - mouse 18-5, 18-9
    - multiple video adapters C-7
  - Multiuser mode
    - defined 3-3
    - system hangs at login prompt 19-9
- ## N
- N01 (boot) floppy 19-2
  - name option, find program 4-10
  - NBLK<sub>n</sub> parameter 8-29
  - NBUF parameter 8-20
  - NCALL parameter 8-25
  - NCLIST parameter 8-22
  - NDISK parameter 8-20
  - NEMAP parameter 8-22
  - netutil program. *See* Micnet network
  - Network, planning 1-18
  - Network printer 15-8
  - New user 10-2
  - newgrp(C), utility 10-8
  - NFILE parameter 8-22
  - NFLOCKS parameter 8-23
  - NHBUF parameter 8-21
  - nice(C), command B-10
  - NINODE parameter 8-22, 19-49
  - NMOUNT parameter 8-23
  - NMUXLINK parameter 8-29
  - NO DEVICES AVAILABLE, error message 19-70
  - No more processes, error message 19-42
  - NO OS, error message 19-9
  - No space left on device, error message 19-16
  - NODE parameter 8-28
  - Nodes, device 13-3
  - Nonstandard disk 7-9
  - NOREAD option, Permissions file 11-26
  - Normal operation 3-3, 3-9
  - NOWRITE option, Permissions file 11-26
  - NPBUF parameter 8-21
  - NPROC parameter 8-7, 8-23, 19-42
  - NQUEUE parameter 8-29
  - NSABUF parameter 8-21
  - NSCRN parameter 8-25
  - NSPTTYS parameter 8-25, 14-18
  - NSTREAM parameter 8-29
  - NSTREVENT parameter 8-29
  - NSTRPUSH parameter 8-29
  - NSXT parameter 8-22
  - Null modem 14-2, 14-28

## Index

### O

- Octal number, onum option 4-11
- OK message, cu(C) command 14-30, 19-28
- ok option, find program 4-10, 4-12
- Open event driver failed, error message 18-7
- Operating system
  - See also* System
  - loading 3-2
  - reinstalling 19-4
- Operating system, reinstalling 19-19
- Operation modes 3-3
- OS/2 coexistence 9-2
- OTHER option, MACHINE entry 11-30
- Out of inodes on device, error message 19-15
- Out of queues, error message 8-6
- Out of streams, error message 8-6
- Ownership. *See* File ownership

### P

- pacct log file 13-7
- Packets, UUCP 11-8, 11-40
- PANIC message. *See* System
- Panic reboot 3-15
- PANICBOOT 3-15
- parallel, port 15-3
- Parallel
  - printer 15-3, 15-6, 15-40
  - See also* Printer problems
- Parameters, boot 3-15
- Parameters, kernel. *See* Kernel parameters
- Parity
  - errors, memory 17-6
  - modem problems 19-30, 19-34
  - terminals 14-2
- Partition
  - defined 9-3
  - DOS 7-11, 9-3, 9-9
  - hard disk 7-15
  - removing 9-10
  - switching 9-5
  - table 9-5
  - two hard disks 9-9
- PASSLENGTH variable 5-5
- Passthrough mode. *See* Transparent mode

- passwd file, *See* /etc/passwd file
- passwd(C), command 5-6, 5-18, 10-4
- Password
  - aging 5-16, 5-17, 5-18
  - assigning 10-4
  - changing 5-4, 5-6, 5-16, 5-17
  - dial-in 14-33
  - dial-in lines 5-20
  - expect string 11-19, 11-20
  - group 5-21
  - guidelines 5-4, 5-5
  - length 5-5
  - preventing changes 5-18
  - root 3-7
  - security 5-4, 5-16
  - super user 1-12
  - Systems file 11-5, 11-8
  - user account 10-1
  - user generating 1-16
- PATH environment variable 8-13
- Pathname, { } command argument 4-10
- Performance, vmstat(ADM) 8-14
- perm option, find program 4-10
- Permission
  - changing 5-10
  - default 5-9, 5-10
  - description 5-7
  - directory 5-4, 5-7
  - displaying 5-7
  - example 5-9, 5-10, 5-11
  - execute permission 5-8
  - file creation mask 5-11
  - files 5-7, 5-21
  - group field 5-8
  - group permissions 5-7
  - locating files 4-10, 4-11
  - no access 5-8
  - other field 5-8
  - other permissions 5-7
  - problems 19-64, 19-65, 19-72
  - read permission 5-8
  - serial port 14-28
  - special files 5-12
  - system utilities 5-4
  - type field 5-7
  - user field 5-8
  - user permissions 5-7
  - write permission 5-8
- Permissions file
  - CALLBACK option 11-27
  - combining entries 11-31
  - COMMANDS option 11-27
  - defined 11-5, 11-22
  - error 11-57, 11-59

- Permissions file (*continued*)
  - example 11-8, 11-12
  - file transfer 11-7
  - format 11-22
  - LOGNAME entry 11-23
  - MACHINE entry 11-23
  - NOREAD option 11-26
  - NOWRITE option 11-26
  - options 11-24
  - READ option 11-25
  - REQUEST option 11-24
  - SENDFILES option 11-25
  - VALIDATE entry 11-28
  - WRITE option 11-25
- ph.daps, printer interface script 15-29
- Phone line
  - checking for noise 19-30
  - modem problems 19-29, 19-70
- Pin connections
  - serial printers 15-5
  - terminals 14-2
- pkttime parameter 11-40
- Planning, installation 1-18
- PN and PS termcap entries 15-12
- Polling, setting for UUCP 11-22
- Port
  - parallel 15-3
  - printer 15-33
  - serial 14-24, 17-7
  - serial printer 15-3
- Port card, installation 17-8
- Port problems 19-36, 19-38
- Ports, settings changed with stty(C) 14-12
- PostScript printer 15-29
- Power cycle, terminal 1-17
- Power failure, recovering from 19-52
- Power supply 1-18
- Powering on the system 3-2
- pprx, printer interface script 15-29
- print option, find program 4-10
- PRINT port 15-12
- Printer
  - acceptance status 15-28
  - adding 15-3
  - administering 15-26
  - administrative commands 15-16
  - BANNERS variable 15-31
  - cabling 15-12
  - checking spooler 1-5
  - class 15-2, 15-21
  - configuring 15-7
  - defaults 15-7, 15-22
  - destination 15-2
  - Printer (*continued*)
    - disabling 15-20
    - dumb model interface 15-30
    - enabling 15-20
    - exit codes 15-32
    - init device file 15-40
    - initialize port 15-34
    - installing 1-19
    - interface program 15-9, 15-10, 15-29, 15-30, 15-35
    - local printing: lprint(C) 15-12
    - LPDEST shell variable 15-22
    - lpsched(ADM), daemon 15-18
    - moving jobs 15-26
    - network 15-8
    - parallel printers 15-40
    - port 15-33, 15-34
    - queue 15-1
    - removing 15-23
    - RS-232 interface 15-1
    - RTS/CTS protocol 15-36
    - serial printer 15-36
    - spooler 15-39
    - spooling, explained 15-1
    - status 15-23
    - terms defined 15-2
    - testing 15-4
    - user command 15-16
  - Printer problems
    - cannot redirect output to 19-37
    - checking device file 19-37
    - creating special device node 19-40
    - idle printer 19-35
    - illegible output 19-39
    - lpsched(ADM) daemon 19-45
    - reconfiguring slow 19-40
    - testing 19-38
    - UUCP errors 19-41
    - wrong output spacing 19-39
  - Printing
    - administering 15-25, 15-26
    - command summary 15-16
    - explaining to users 1-16
    - load management 15-25
    - requests 15-20
    - starting 15-18, 15-19, 15-26
    - stopping 15-18, 15-25
  - Printronix line printer 15-29
  - Proceed with cleaning (y or n)? message 3-3
  - Process
    - batch(C) B-5
    - batch(C) B-6
    - delayed execution B-5

## Index

### Process (*continued*)

- increasing limits 8-8
- maximum user size 17-6
- memory management and swapping 8-23
- parameters 8-23
- problems 19-42, 19-43, 19-44
- queuedefs file B-10
- runaway 1-17
- Process accounting 5-26, 5-27
- Process table
  - EAGAIN error 8-23
  - size 8-23
- procl(S) 17-6
- .profile file 14-14, 14-15
- Program, simultaneous execution 1-14
- Prompt, login 3-7
- prx, printer interface script 15-29
- ps(C), command 1-17, 8-12
- Pseudo-tty, limit 8-25, 14-18
- pstat(C), command
  - active inodes 19-48
  - active processes 19-43
- ptys 14-17
- pwadmin(ADM), command 5-16, 5-17, 5-18
- pwcheck(C), utility 10-10

## Q

- QIC-02 tape drive 16-3
  - base address 16-6
  - controller type 16-4
  - DMA channel 16-4
  - interrupt vector 16-5
  - size 16-10
- QIC-40/80 tape drive
  - described 16-7
  - problems 19-56
  - size 16-11
- queue max run limit reached, error message B-11
- queuedefs file B-10
- Quitting, login session 1-13
- quot(C), command 4-9

## R

- RAM (Random Access Memory)
  - fully recognized 19-26
- Random Access Memory (RAM), minimum and recommended 19-2
- rc directories 13-3
- rcp command 12-1
- READ option, Permissions file 11-25
- Read permission 5-8
- Ready to Send (RTS) 15-36
- Reboot, *See also* Boot
- Receive data 14-2
- Recursion 4-10, 4-12
- reject(ADM), command 15-25
- remote command 12-1
- REMOTE DOES NOT KNOW ME, error message 19-71
- REMOTE HAS A LCK FILE FOR ME, error message 19-71
- remote.unknown script 11-40
- Removing
  - memory 17-6
  - printer or class 15-23
  - terminal 14-15
- Request ID 15-1
- Request ID, printing 15-20
- REQUEST option, Permissions file 11-24
- Resetting terminal 1-17
- Resources. *See* System resources
- restore(ADM), utility 6-3, 19-21
- Restoring
  - boot file 19-4
  - /etc/init file 19-8
  - /etc/login file 19-9
  - filesystems 19-18
  - free space 19-17
  - super block, root filesystem 19-9
  - system files 19-4, 19-5, 19-9
  - /xenix file 19-5
- Restricted accounts 5-32
- Restricted shell 5-30, 5-31, 5-32
- Retry period, UUCP 11-17
- RETRY TIME NOT REACHED, error message 19-71
- Reverse video, colors C-11
- rksh(C), utility 5-30
- rmail(ADM), command 11-23, 11-27, 11-28, 11-30
- rmuser(ADM), command 10-13, 10-15
- ROM tables 7-4

- Root
    - See also* Super user
    - directory 13-2
    - filesystem 4-2
    - filesystem backups 6-1
    - hard disk 7-2
    - login 3-7
    - password 3-7
  - Root filesystem 19-7, 19-8, 19-10, 19-19
    - backup 6-33
    - mount table 8-23
  - Root password 3-4
  - RS-232 15-36
    - UUCP 11-2, 11-41, 11-42
  - RS-232 interface 15-1
  - rsh(C), utility 5-30, 5-31
  - RTS/CTS protocol 15-1, 15-36
  - Runaway process, killing 19-43
- ## S
- SA\_EDITOR environment variable 6-4
  - Safe to Power Off message 3-10, 3-11, 17-6
  - SANE, gettydefs flag 14-8
  - Scan, destructive 7-13
  - Scan window 2-13, 2-22
  - Scatter/gather feature 8-11, 8-32
  - schedule(ADM), file 6-3, 6-4, 6-5
  - Scheduler is not running, error message 19-46
  - Scheduler problems 19-45
  - Scheduling backups 6-4
  - scopatch(ADM), command 11-40
  - Scratch file 19-19
  - Screen protection, console C-4
  - Screen saver, TBLNK parameter 8-25
  - Screens, multiple 1-14
  - Screens, maximum 14-18
  - Screens, multiple. *See* Multiscreen C-5
  - SCRNMEM parameter 8-25
  - SCSI
    - disk configuration 7-4
    - Embedded controller 7-5
    - ID number 7-4
    - Logical Unit Number (LUN) 7-4
    - root disk 7-2
    - tape drives 16-8
  - SCSI tape drive, *See also* Tape drive problems
  - Search, *See also* find
  - Search permission. *See* Execute permission
  - Sector, track 13-10
  - Security
    - floppy disk 5-3, 5-4
    - hardware 5-3
    - special file 5-12
    - super user 1-12
  - SEMAEM parameter 8-27
  - Semaphores 8-27
  - Semicolon (;), find(C), command 4-10
  - SEMMAP parameter 8-27
  - SEMMNI parameter 8-27
  - SEMMNS parameter 8-27
  - SEMMNU parameter 8-27
  - SEMMSL parameter 8-27
  - SEMOPM parameter 8-27
  - SEMUME parameter 8-27
  - SEMVMX parameter 8-27
  - Send string 11-19
  - SENDFILES option, Permissions file 11-25
  - Serial
    - cable 11-42, 11-43
    - console 14-16
    - line 14-2
      - Micnet network 12-4, 12-11
    - line, *See* Serial line
    - multiscreens (mscreen) 14-17
    - port, *See* Serial port
    - printer 15-3
    - terminal 18-9
  - Serial card
    - attaching a printer 15-6
    - installing 17-7
    - multiport 14-24
  - Serial line
    - conflicts 14-33
    - connecting 1-19
    - disabling 11-42
    - displaying characteristics 14-12
    - filenames 13-10
    - lock file 11-8
    - modem connection 14-24
    - settings, standard 19-34
    - testing 11-42
  - Serial port
    - access error 17-9
    - adding 17-7
    - choosing 14-24
    - permissions 11-42, 14-28
    - UUCP 11-41
  - Serial printer 15-5
  - Serial printer. *See* Printer problems
  - serial(HW) 17-7

## Index

- set init example 14-22
- set noglob example 14-22
- set prompt example 14-22
- setcolor(C), command C-10
- setcolor(C), command C-11
- setcolor(C), command C-12
- setenv TERM 14-14
- Setting, terminal type 14-13, 14-15
- Settings, switch 17-9
- Setup disk 17-3
- sh, TERM assignment 14-14
- Shadow inodes 4-18
- Shared data parameters 8-28
- Shell, exiting 1-13
- Shell escape, sysadmsh(ADM) 2-20
- shl(C) 14-20
- /shlib, directory 13-6
- SHMALL parameter 8-28
- SHMMAX parameter 8-28
- SHMMIN parameter 8-28
- SHMMNI parameter 8-28
- SHMSEG parameter 8-28
- Shutdown
  - abnormal 19-21, 19-52
  - procedure 3-9, 3-10
  - single-user option 4-25
  - sysadmsh(ADM) utility 3-4
- shutdown(ADM), command 3-4, 3-9, 4-25, 19-52
- Signal ground 14-2
- Single-user mode
  - changing kernel parameters 8-3
  - defined 3-3
  - fixing a bad track 19-49
  - restoring /xenix 19-5
  - running fsck(ADM) 19-51
  - shutdown(ADM) command 19-52
  - shutdown(ADM) option 4-25
- Sitename 6-4, 19-73
- size option, find program 4-10
- Slash (/), root filesystem 4-2
- Slave tty
  - create example 14-18
  - defined 14-17
- Slots 17-3
- Smart card 17-7
- Smartmodem 1200, 2400, V-series 9600 14-25
- SMS-OMTI, disk controller 7-8
- Special device node, creating 19-40
- Special file, protection 5-12
- Special filenames 13-9
- Spooler
  - printer 15-39
- Spooler (*continued*)
  - UUCP 11-2
- srmount: panic cannot cvtv7superb()
  - yet, error message 19-25
- ST506
  - configuration 7-3
  - disk controller 7-7, 7-8
  - root disk 7-2
- Stage 1 boot failure, error message 19-4
- Starting, terminal output 1-13
- Static electricity 17-2
- Status, printer 15-28
- STATUS error messages 11-58
- .Status file, error 11-57
- Status files, UUCP 19-69
- Status Line (sysadmsh) 2-4
- Stop bits 14-2
- Stopping
  - login session 1-13
  - print request 15-20, 15-25
  - program 1-13
  - system 3-9, 3-10
  - terminal output 1-13
- Storing, *See also* Archiving
- Straight-through cables 14-28
- STRCTLSZ parameter 8-30
- Stream-head structures 8-29
- Streams parameters 8-28, 8-29
- STRLOFRAC parameter 8-30
- STRMEDFRAC parameter 8-31
- STRMSGSZ parameter 8-30
- stropen: out of queues, error message 8-6
- stropen: out of streams, error message 8-6
- stty(C), command 14-10, 15-31, 15-33
  - displaying settings 14-12
  - limitations 14-12
- su attempts, monitoring 5-25
- su, *See* Super User
- sulog log file 13-8
- SULOG variable 5-25
- Super block
  - correcting values 19-23
  - defined 4-24
  - repairing damage with fsdb 19-21
- Super block, restoring 19-9
- Super user
  - defined 1-12
  - exiting 3-8
  - login 3-7
  - password 3-4, 5-4
  - password, restricted use 1-12
  - precautions 3-8

- Super user (*continued*)
  - prompt (#) 3-7
  - security 1-12
  - single-user mode 3-4
  - special file access 5-12
- Swap space 17-6, 19-26
- Swapping, kernel parameters 8-23
- Switch settings 7-4
  - hard disk 7-4
  - modem 14-25
  - motherboard 17-9
- Switching operating systems 9-5
- Switching screens 1-14
- sync(ADM), command 4-24, 4-25
- sysadmsh(ADM), utility
  - backups 6-2, 6-18, 6-20
  - creating backups 6-11
  - defined 1-8, 2-1
  - Display Area 2-4
  - error messages 2-4
  - Filesystem options 7-18
  - HardDisk options 7-6
  - help 2-15
  - keystrokes 2-9, 2-11, 2-12, 2-14, 2-19
  - menu 1-8, 2-3, 2-4, 2-5, 2-6, 2-22
  - options 2-3, 2-5, 2-6, 2-7
  - shell escape 2-20
  - shortcuts 2-22
  - unscheduled backup 6-15
- Sysfiles file 11-39
- System
  - See also* Console; Kernel, parameters
  - administrator, *See* System administrator
  - administrator
    - board 17-9
    - customizing startup A-1
    - diagnosing inefficiency, vmstat(ADM) 8-14
    - efficient usage 8-12
    - error log 13-5
    - /etc/cshrc A-5
    - /etc/profile A-5
    - exceeding limits 8-6
    - files 19-4, 19-5, 19-7, 19-8, 19-9, 19-17, 19-47
    - hardware 3-12, 3-13
    - initialization files A-1
    - initialization files A-2
    - initialization files A-5
    - initialization files A-6
    - maintenance 1-2, 1-3, 1-8, 1-12, 4-6
    - name parameters 8-28
    - panic 8-6, 19-50, 19-51
    - panic. *See also* System problems
  - System (*continued*)
    - performance 8-7, 8-8, 8-20
    - planning 1-18
    - problems 19-8, 19-9, 19-19, 19-21, 19-43, 19-47, 19-48
    - reducing load 8-12
    - safety 1-18
    - security 5-3
    - shutdown 3-9, 3-10, 19-52
    - special tuning needs 8-8
    - starting 3-2
    - stopping 3-9
    - time 3-5
  - System administrator
    - backups 6-2
    - defined 1-2
    - file access 5-12
    - front-end account 5-32
    - log book 1-2, 1-4
    - Micnet network maintenance 12-1
    - super user login 1-12
    - tasks 1-3
    - tools 1-8
    - user account creation, maintenance 10-1
  - System maintenance mode 3-3, 3-4
  - SYSTEM NOT IN Systems FILE, error message 19-71
  - System resources
    - buffers 8-1
    - common needs 8-7
    - kernel 8-3
    - limits 8-1
    - reasons for reallocating 8-2
    - special 8-1
    - table entries 8-1
  - System utilities, permissions 5-4
  - Systems file 11-6
    - baud rate 11-10
    - device field 11-18
    - device requirements 11-6
    - error 11-59, 11-60
    - error with cu command 19-64
    - example 11-8, 11-12
    - format 11-15
    - LAN switch 11-19
    - login sequence 11-5, 11-8
    - modem 11-10
    - phone field 11-18
    - schedule field 11-16
    - security 11-15
    - speed field 11-18
    - testing modem 11-14
    - UUCP 19-70

## Index

Systems file 11-6 (*continued*)  
  UUCP configuration 11-5  
  UUCP messages 19-71  
  verifying phone number 11-14  
System-wide message 3-9  
SYSTTY 14-16

## T

TAB3, gettydefs flag 14-7  
Table entries 8-1  
Tabs, changing 14-13  
tail program, monitoring log files with  
  4-15  
Tandy DMP printers 15-29  
Tape  
  archiving files 16-11  
  device names 13-3  
  /etc/default files 16-9  
  formatting 16-13  
  QIC-02 16-3  
Tape buffers, values 8-21  
Tape controller not found, error message  
  19-54  
Tape drive  
  backups 6-3  
  boot screen messages 16-9  
  defined 16-2  
  Error Correction Code (ECC) 16-14  
  /etc/default files 16-9  
  installing 16-2  
  Irwin 16-6, 16-13  
  maintaining 16-12  
  mcconfig file (irwin) 16-6  
  mcdaemon (irwin) 16-7  
  mini-cartridge 16-6, 16-10  
  QIC-02 16-3, 16-10  
  QIC-40/80 16-7, 16-11  
  SCSI 16-8  
  sizes 16-10  
Tape drive problems 19-54, 19-55, 19-  
  56, 19-57  
tape(C), utility 16-12  
tar(C), command 19-57  
  archiving files 16-11, 16-18  
  default settings 16-9  
Tasks, administrative 1-4  
TBLNK kernel parameter C-4  
TBLNK parameter 8-25  
Telebit modem 14-26  
Temporary files  
  locating 4-11

Temporary files (*continued*)  
  removing 4-13  
TERM environment variable 14-13,  
  19-61  
termcap(F) 14-2  
Terminal  
  adding 14-2  
  cabling 14-2  
  characteristics, displaying 14-12  
  disabling 14-15  
  display 1-13  
  installing 14-2  
  lines, setting 14-6  
  local printer: lprint(C) 15-12  
  mouse use 18-5, 18-9  
  settings, checking 14-11  
  transparent mode 15-13  
  type, setting 14-13, 14-15  
Terminal problems  
  configuration settings 19-58  
  hung 19-59  
  no login prompt 19-58  
  non-echoing 19-61  
  stopping a runaway process 19-60  
Testing, UUCP 11-41  
Time, system 3-5  
TIMEOUT 3-15  
Timeout table overflow, error message  
  8-6, 8-25  
TIMEZONE parameter 8-25  
TM. (temporary data) files 11-8, 11-46  
/tmp, directory 13-6  
Training, for new users 1-16  
Transmit data 14-2  
Transparent mode 15-13  
Trap E in system, error message 19-51  
Troubleshooting utilities, UUCP 19-75  
tset(C), command 1-17, 14-15, 14-22,  
  14-23, 19-59, 19-61  
tty line. *See* Serial line  
tty ports, setting with stty(C) 14-12  
tty(C) 14-15  
ttys  
  device file 13-10  
  modem 14-24  
  mouse 18-5  
ttys file 14-3, 14-28  
Tunable kernel parameters. *See* Kernel  
  parameters  
-type option, find program 4-10  
TZ (timezone) variable 8-25

## U

- umask(C), command 5-11
- umount(ADM), command 4-3, 4-25, 7-19, 16-21
  - floppy filesystem 19-5
  - hard disk root filesystem 19-7
- /unix file 13-2
- Unkillable process, removing 19-44
- Unlocking, console keyboard 19-11
- Unscheduled backups 6-2, 6-15
- update program 4-24
- Use Entire Disk for XENIX, fdisk
  - option 9-8
- Use factor 5-28
- usemouse(C), command 18-7
- User
  - See also* Account
  - adding 10-2
  - changing user ID 10-11
  - disk usage 4-9
  - group, changing 10-9
  - home directory 7-19
  - ID 10-3, 10-11
  - login group, changing 10-9
  - login name 10-3
  - password. *See* Password
  - permissions. *See* Permission
  - print commands 15-16
  - process size 17-6
  - removing 4-12, 10-13
  - shell type 10-4
- user option, find program 4-10, 4-12
- /usr directory 13-5
  - /usr/adm directory 13-5
  - /usr/adm/lastlog file 5-21
  - /usr/adm/messages file 4-14, 15-5, 15-6
  - /usr/adm/pacct file 4-14, 5-26
  - usr/adm/sulog file 4-14
  - /usr/bin directory 11-3, 13-5
  - /usr/include directory 13-5
  - /usr/lib directory 13-5
  - /usr/lib/sysadmin/schedule 6-4
  - /usr/lib/uucp directory 11-4, 11-5, 11-10
  - /usr/lib/vidi, directory C-8
  - /usr/spool directory 13-5
  - /usr/spool/lp/log file 4-14
  - /usr/spool/lp/model file 15-30
  - /usr/spool/micnet/remote log files 4-14
  - /usr/spool/uucp directory 11-5
  - /usr/spool/uucp directory, UUCP status files 19-69
  - /usr/spool/uucp/.Log directory 19-69
  - /usr/spool/uucp/LOGFILE file 4-14
  - /usr/spool/uucp/.Log/Old files 4-14
  - /usr/spool/uucppublic directory 11-5
  - /usr/sys/conf directory 8-3
  - /usr/tmp directory 13-5
- uuchat program 11-38
- uucheck(ADM), command 11-4, 19-75
- uucico(ADM), daemon 11-2, 11-4, 11-6, 11-7, 11-8, 11-40, 19-69
- uuclean(ADM), command 11-4
- UUCP
  - access 11-5
  - ACU (Automatic Call Unit) 11-18
  - administering 11-44
  - allowing remote commands 11-27
  - commands 11-9
  - configuring 11-10, 11-15
  - connecting 11-2, 11-3
  - creating login accounts 11-12
  - daemons 11-6, 11-29
  - default paths 11-28
  - defined 11-1
  - Direct link 11-32
  - direct wire 11-41
  - directories, *See* UUCP directories
  - error messages 11-55
  - examples 11-49
  - files 19-69, 19-72
  - files, *See* UUCP files
  - getty program 11-38
  - granting access 11-28, 11-29
  - Hayes 2400 dialer 11-10
  - Local Area Network (LAN) 11-31
  - login 11-13, 11-16, 11-19, 11-21
  - login script 11-20
  - login sequence 11-20
  - messages 19-69, 19-70, 19-71, 19-72
  - micnet sites 11-40
  - modem 11-2, 11-10, 11-13, 14-27
  - node name 11-15
  - passwd file 11-12
  - password 11-13
  - polling 11-22
  - printer 15-8
  - problems 19-41, 19-64, 19-65, 19-68, 19-73, 19-74, 19-75
  - remote commands 11-46
  - retry period 11-17
  - sample transaction 11-7
  - security 11-22, 11-26, 11-27, 11-29, 11-40
  - setting up 11-2
  - shared dial-in/dial-out 11-38, 11-39
  - special login shell 11-13

## Index

UUCP (*continued*)  
  uuchat 11-38  
UUCP daemons 11-28  
UUCP directories  
  public directory 11-23  
  security 11-25, 11-26  
  spool directory 11-46  
  /usr/bin/uucp 11-5  
  /usr/spool/uucp 11-5  
  /usr/spool/uucppublic 11-5  
UUCP files  
  C. (work) files 11-4, 11-7, 11-46,  
    11-56  
  control files 11-2, 11-8, 11-10, 11-11  
  D. (data) files 11-4, 11-46, 11-47,  
    11-56  
  Devices file 11-5, 11-10, 11-32  
    ACU (Automatic Call Unit) 11-31  
  Dialcodes file 11-39  
  Dialers file 11-33, 11-34, 11-35, 11-  
    39, 11-60  
  editing 11-11  
  LCK (lock) files 11-8, 11-14, 11-46,  
    11-55, 11-60  
  log files 4-14, 11-4, 11-5, 11-9, 11-44,  
    11-45  
  maliasies file 11-40  
  Permissions file 11-5  
  Sysfiles file 11-39  
  Systems file 11-15  
  TM. (temporary data) files 11-8,  
    11-46  
  uudemon.hour 11-7  
  X. (execute) files 11-4, 11-6, 11-9,  
    11-46, 11-48, 11-56  
UUCP log files 13-8  
UUCP programs  
  rmail(ADM) 11-23  
  uuname(C) 11-54  
uucp(C), command 11-1, 11-3, 11-4,  
  11-7, 14-34, 19-66, 19-68  
uucppublic directory 11-25  
uudemon.admin script 11-44  
uudemon.clean script 4-14, 11-4, 11-44,  
  11-46  
uudemon.hour script 11-7, 11-44  
uudemon.poll2 script 11-44  
uuiinstall(ADM), command, changing  
  sitename 19-73  
uulog(ADM), command 11-4, 11-45,  
  19-66, 19-75  
uuname(C), command 11-54, 19-75  
uupick(C), command 11-3  
uusched(ADM), daemon 11-6, 11-7

uustat(C), command 11-3, 19-66, 19-75  
uuto(C), command 11-4, 11-45  
uutry(ADM), command 11-14, 11-21,  
  11-60, 19-68, 19-75  
uux(C), command 11-3, 11-4, 11-5,  
  19-74  
  defined 11-1, 11-27  
  records 11-4, 11-45  
uuxqt(ADM), daemon  
  defined 11-28  
  executing remote programs 11-6,  
    11-9  
  MACHINE option 11-29

## V

VALIDATE entry, Permissions file  
  11-28  
Variables. *See* Environment variable  
Ventilation 1-18  
Video adapter. *See* Display  
Video fonts, changing C-8  
vidi(C), utility  
  switching adapter C-7  
  switching adapter C-9  
  switching fonts C-8  
vmstat(ADM), utility 8-14

## W

wall(ADM), command 3-9, 4-8  
Warning! All data on your disk will be  
  lost! message 7-12  
Window, scan window 2-13  
windows parameter 11-40  
Wiring, modems 14-27  
WRITE option, Permissions file 11-25  
Write permission 5-8  
WRONG TIME TO CALL, error  
  message 19-72  
wtrmp log file 13-7  
Wyse 60 terminal 14-23, 15-13

**X**

- X. (execute) files 11-4, 11-6, 11-9, 11-46, 11-48, 11-56
- xargs(C), command 4-12
- XENIX, *See also* System
- /xenix file, restoring 19-5
- xenix not found, error message 19-5
- /xenix.old, alternate kernel file 19-5
- XON/XOFF
  - serial printer 15-3
  - terminal 14-2
- XT console keyboard 19-14
- XT console keyboard C-2







AZ01201P002

64361



MANUAL



AU01203P001