START HERE !

This binder contains everything you need to know about installing XENIX and running XENIX programs.

- Before You Begin:

  1 - Check the Compatible Hardware section of the *Release Notes* in this binder before making any hardware modifications to your system.

  2 - Read the "Installation Notes" in the *Release Notes.*

  3 - Glance through Chapter 2, "Installation Procedure," in the *Installation Guide*, also in this binder. Then, use that chapter to help you install XENIX.

  4 - Refer to the appropriate section in the Release Notes on any question you might have. The Release Notes are made up of different sections grouped by topic.

- If you've never used XENIX, refer to the *Introduction To XENIX* in this binder. It shows you how to use some of the basic XENIX commands.

- The *Release Notes* contain important information about your system. Refer to them if you have any questions.

- For information on operating your system, including how to add terminals, line printers and other devices, and perform system backups, see the *Operations Guide*.

The XENIX® System V

Operating System

Release Notes

Version 2.2.3

The Santa Cruz Operation, Inc.

This document was developed and typeset using The SCO XENIX System V Operating System and printed on an IMAGEN® 8/300 Laser Printer.

Document Number: XG-2-29-88-4.1.1/2.2.3

Processed Date: Tue Mar 15 15:03:02 PST 1988

Contents

B.1 Common Questions and Answers

SCO XENIX System V
Operating System Release Notes

Release Notes
SCO XENIX System V
Operating System
Release 2.2.3
February 29, 1988

1. Preface

This document contains important information about the SCO XENIX System V Operating System Release 2.2:

- Installation Notes
- Software Notes
- Documentation Notes
- Compatible Hardware
- Common Questions and Answers
- Updated replacement manual pages.

These notes are organized into sections by topic. There is a section for notes about system installation, a section for system user's notes that should be read by every user on the system, a section of notes of interest to the System Administrator, and sections on using various tools and peripherals such as DOS, networks, printers, tape drives, hard disks, etc.

Note

> Please read through the Notes About Installing Your Software section of these Release Notes before installing the XENIX System V Operating System. Also, please read the sections of these notes pertaining to any peripheral devices such as printers, tape drives or any other hardware you are installing.
>
> Also, please note that certain hardware configuration information included in Appendix A of these notes may be required for successful XENIX installation. Please refer to those sections of Appendix A that apply to the hardware in your configuration if you have difficulty.

We are always pleased to hear of users' experiences with our product, and recommendations of how it can be made even more useful. All written suggestions are given serious consideration.

The manual pages **fd**(HW), **diskcp**(C), **dos**(C), **dtype**(C), and **mkdev**(C) have been updated and replacement pages are included with these notes. Please replace the existing pages with these updated ones.

1.1 Contents of the Distribution

The XENIX System V Operating System Release 2.2 is distributed on the following 96tpi floppies:

Volumes N1-N3 (N1-N2 on 286 machines)
Volume B1
Volumes X1-X3
Games (Optional) - Volume 1

The XENIX System V Operating System Release 2.2 is also available on the following 48tpi floppies:

Volumes N1-N7
Volumes B1-B3
Volumes X1-X10
Games (Optional) - Volume 1

The XENIX System V Operating System Release 2.2 is also available on the following 135tpi 3.5 inch floppies:

Volumes N1-N6
Volumes B1-B2
Volumes X1-X5
Games (Optional) - Volume 1

The software is grouped into *packages* that are listed in the next section, "Packages In This Set."

1.2 Packages In This Set

The XENIX System V Operating System software is grouped into *packages*. This makes customizing your system easier, since you can use the **custom**(C) utility to add or delete groups of programs that have related functions.

For example, if you do not want to use your system for communications, you would not install the UUCP package. If you install a package, then change your mind later, use **custom** to remove that package. **custom** can locate all the files that belong in a certain package and delete them, thus removing the package.

These are the packages in the XENIX System V Operating System: (the numbers are the approximate sizes of the individual packages in 512 byte blocks)

XENIX System V Operating System Packages

Package	Size (in blocks)	Use/contents
ALL	13028	Entire Operating System set
RTS	3648	XENIX run time system
LINK	1686	The link kit
BACKUP	282	System backup and recovery tools
BASE	926	Basic extended utility set
CSH	106	The C-shell
DOS	346	DOS utilities
EX	336	The ex and vi editors
FILE	436	File manipulation tools
IMAGEN	272	Imagen Laser Printer Support
INITTAB	10	Terminal initialization
LPR	462	Multiple line printer spooler
MAIL	628	Electronic mail and local area networking
MAPCHAN	630	International character set mapping
PERM	52	XENIX contents and permissions lists
SYSADM	1024	System administration tools
TERMINF	440	Terminfo Database
UUCP	792	Uucp and cu communications utilities
VSH	256	The visual shell

1.3 Software Support

Software support is available to customers who purchased XENIX for use in the United States and Canada. If you purchased XENIX for use outside of the US or Canada, contact your distributor or retailer for support information.

Software support is described on an insert in the back of the XENIX documentation.

2. Notes About Installing Your Sofware

Please refer to the *Installation Guide* in the binder marked "XENIX Operating System *Run Time Environment*" to install the XENIX System V Operating System. Chapter 2, "Installation Procedure," describes the actual installation steps.

If you have all three systems, (the XENIX Operating System, Development System and Text Processing System) you can use the XENIX *Installation Guide* to install them all at once. If you wish to install the Development System and/or the Text Processing System at a later time, refer to the *Release Notes* at the beginning of the appropriate guide (XENIX *Programmer's Guide Volume I* or XENIX *Text Processing Guide*). See also the manual page for **custom**(C). With **custom** you can install all or portions of the XENIX System.

This installation is not an update, and it will overwrite the present contents of the hard disk drive 0. If you wish to update from an earlier release or version of XENIX, and you have not yet obtained an update, contact your sales/support center.

If you are upgrading your 286 AT to 2.2, you can continue to use a pre–2.2 XENIX filesystem on a second hard disk by following the instructions in the section "Adding a 2.1 Hard Disk to 2.2" later in these *Release Notes*.

Read the *Release Notes* and *Installation Guide* and make sure you completely understand the installation process before installing the product.

There are some specific points to be aware of before you start:

- Do not abort the installation process by using the DEL or Ctrl-\ keys. If you need to stop because you enter incorrect information or for some other reason, press the computer "reset" button or power cycle your system and start the process again from the beginning rather than trying to proceed from the stopping point.

- Users upgrading their XENIX 3.0 system to XENIX System V should refer to Appendix A, "Upgrading Your System" in the XENIX *Installation Guide*. Users installing XENIX for the first time can ignore Appendix A of the *Installation*

Guide. Do not confuse Appendix A of the *Installation Guide* with Appendix A of the *Release Notes*, "Compatible Hardware."

- XENIX and DOS must boot from the physical drive 0 (hard disk or diskette). Keep this in mind when planning for extra hardware.

- XENIX installs with a minimum of screen prompts and typing, and is set up with ample defaults for most uses and sites. However, you can customize the operating system during installation to meet a wide variety of needs.

  If you know that you need to customize your installation, go over this next list and decide if any of the items apply to you. If they do, read any suggested sections and make sure you understand the procedures involved before installing XENIX:

  - Install all or only parts of the operating system.

    This is described during the installation process, but the section of these *Notes* "Contents of the Distribution" can help you decide what software you need, and what software you may not need.

  - Create additional filesystems, besides the *root*.

    This is described in Chapter 2 of the *Installation Guide*, "Installation Procedure" and in the section "Deciding Whether to Have a Separate /u Filesystem" later in these *Notes*.

  - Add a second hard disk that contains filesystems created under XENIX 2.1.

You can continue to use the filesystems on a second hard disk that you created under XENIX 2.1 or later.

Do not install XENIX 2.2 right away. Instead, you should read the subsequent section of these *Release Notes* "Adding a 2.1 Hard Disk to 2.2" before installing XENIX.

If you intend to preserve and use 2.1 filesystems on drive 0 after installing 2.2, you must obtain the 2.1 to 2.2 update from your sales/support center. The XENIX 2.2 installation will completely replace a 2.1 installation on drive 0.

- Set up your hard disk to accommodate other operating systems, such as DOS.

  This is discussed in Chapter 2 and Chapter 3 of the *Installation Guide*, "Installation Procedure" and "Using DOS and XENIX on the Same Disk."

As stated before, most users and sites do not need to extensively customize their installations since the defaults satisfy most requirements. If you need to customize your installation, though, make sure you have a clear idea of what you want before you start the actual installation process.

- If you plan to use DOS and XENIX on the same hard disk, install DOS first, then install XENIX. This is not required by XENIX but by some versions of DOS.

2.1 Keeping Consistent Systems

We do not recommend using mismatched portions of XENIX Releases. For example, we do not recommend installing a 2.2.2 Link Kit on a 2.2.1 XENIX Operating System. Note that you can use different releases of different systems, for example a 2.2.2 Operating System with a 2.2.0 Development System. This warning only applies to mismatched releases of the same system.

This warning applies to both the Operating System and the Development System.

To make sure that all of the pieces of the XENIX System work together properly, either reinstall a new release over a previous one, or use standard update software available from your dealer. Any other approach may cause system errors and other problems.

Information on performing updates is furnished with the appropriate update software. Also, general information on upgrading your system to a new release is described in Appendix A, "Upgrading Your System" in the XENIX Installation Guide.

2.2 Memory Requirements

Please use the following table to determine the amount of memory you need to run the XENIX System V Operating System and Development System:

System	Requirements
XENIX 86/286 Operating System	**minimum**: 512K **recommended**: 640K **multiuser**: 1-2MB, or more for optimum performance
XENIX 386 Operating System	**minimum**: 1MB **recommended**: 2MB **multiuser**: 2MB or more for optimum performance
XENIX 86/286 Development System	**minimum**: 640K **recommended**: 1MB
XENIX 386 Development System	**minimum**: 1MB **recommended**: 2MB

When you invoke a utility that needs more memory than is available, the message "Killed" or "too big" displays on your screen.

Note

"Hidden" Memory on 386 Machines

Certain manufacturers reserve memory, such as the upper 384K of the first megabyte of memory, for DOS. On some machines, XENIX cannot access this "shadow" RAM. One megabyte of "visible" RAM is required in order to install the Operating System. Should the upper RAM be inaccessible on your machine, you will have to install additional memory to compensate for this loss. If you install a second megabyte of RAM in this example, you would have 1.6 megabytes of usable memory.

2.2.1 vi(C) and vsh(C)

vi(C) and **vsh**(C) are not supported for 8086 or 80286 machines with 512 Kilobytes(K) or less of main memory. If you have only 512K of main memory installed, use of any medium model program may severely impair system performance.

2.2.2 Display Adapters

If you have two display adapters on a machine running with less than 640K of RAM, you see the following message at boot time:

```
Insufficient memory for second display adapter
```

In that case, the second adapter is not used by the XENIX System V Operating System.

2.2.3 Running Out of Swap Space

On 286 machines, running out of swap space causes an immediate *panic* and total system halt. On 386 machines, the system can recover. If you see the kernel warning: "out of swap" on a 386 machine, you must kill some processes quickly. A 386 machine will *panic* and halt if it runs out of both memory and swap space. Refer to "Killing a Process" in Chapter 4 of *Introduction to XENIX* for instructions.

Be sure to allocate plenty of swap space during installation, as the only way to add more swap space is to reinstall.

2.3 Swap Space Allocation on 386 Machines

The virtual memory feature of 386 machines allows the maximum size of a user process to be much larger than physical memory. The default formula for this release is that "a process may occupy all of user memory plus 30 percent of the swap area." You can alter these limits with **configure**(C) by changing the parameters MEMLIM and SWPLIM, which have default values of 100 and 30, respectively.

The maximum user process size is displayed at boot time. If you intend to run very large applications, you should allocate swap space according to the formula below. For typical multiuser operation, other considerations apply.

Ideally, all processes should fit in memory. For best results, swap space should be at least as large as memory. If it is smaller, you may occasionally see the message: "warning: out of swap". This is a warning, not a *panic*. It means that the paging mechanism is not operating efficiently due to lack of swap space.

To calculate multiuser swap size (286 or 386):

1. Multiply the number of users on the machine by the size of the largest process normally run on the machine, in K bytes. If no unusual processes apply, use 512K.

2. Take the amount of memory installed in your machine and add the figure 256K. Compare this result with the result from step 1. Use whichever number is larger as your multiuser swap size.

3. Adjust this number upward a bit if multiple users are running different large applications. (Try adding 512K per different large application.)

For example, for an 8-user machine with 4Mb of memory and a typical departmental computing mix of a 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 above.

3. Add 4 x 512K, which yields a swap space allocation of 6400K.

2.4 80386 32-bit Multiply Bug

There is a serious, but not immediately apparent problem that affects some 386 CPU chips. Although the computer and operating system appear to function normally, this hardware problem can cause damage to data and programs.

According to Intel, the microprocessor may incorrectly multiply very large numbers during 32-bit operations. The hardware fix requires a new CPU chip or motherboard.

The kernel will not boot if it detects this hardware error, and this message is displayed:

HARDWARE FAILURE:
386 incorrectly multiplies 32 bit numbers

In addition, the system will *panic*. At your own risk, you may override this protection mechanism by adding the keyword *mulbug* to the boot line:

Boot
: xenix mulbug

We do **NOT** recommend using a machine with this hardware problem and systems with this hardware problem are not supported.

80386 chips marked "16 bit only" might not present the multiply bug but might also fail with XENIX 386. Chips marked "Σ Σ" have been tested by Intel to function correctly.

For more information about this problem, refer to the "Administering Your System" section of these *Release Notes*.

2.5 Deciding Whether to Have a Separate /u Filesystem

If you install XENIX on a hard disk that is larger than 20 Mb, you may be prompted to decide if you wish to divide the XENIX partition into two filesystems. Traditionally, the second filesystem is called */u*, although you can name it anything you like, and it is typically used for user accounts and files. Throughout these *Release Notes* the second filesystem is referred to as */u*.

Note that any figures given are rough guidelines, since XENIX installations vary widely depending upon their use.

The advantages to having more than one filesystem are:

- It can make system backups easier.
- It is faster to clean smaller filesystems (using **fsck**(C)).
- Your system is easier to manage.

The only disadvantage is that once you make the division it is time consuming to change the size of each of the filesystems. If you discover that you have made a filesystem too small, or that you no longer want multiple filesystems, you must backup your entire XENIX partition, change the filesystems with the **divvy**(C) utility, then restore the entire system from the backups.

If you have a large hard disk, it might be a good idea to create a */u* filesystem if:

- You are running a typical multiuser system where user files change often but system files remain fairly static.

  Most users create and modify files in their home accounts frequently and as a result, user accounts must be backed up often. However, once XENIX and your applications are installed, most of the utilities and system files remain unchanged, or do not change as often. Thus, it can make sense to put all the user accounts together on a separate filesystem.

  If you have a separate */u* filesystem, less media (e.g. fewer floppies, fewer tapes, smaller tape size, etc.) is required to make backups because you can tailor your backup schedule more closely to your system needs.

- There is more than 10 Mb left on the *root* filesystem after you account for all of the system utilities and applications you plan to install.

  The exact amount of extra room you should allow varies depending upon what the system is used for and what applications you plan to install. While some applications programs can be installed on any file system, others might not be so flexible.

  You should leave the XENIX system files in the relative locations in which they are installed. Be sure to leave enough space on your *root* filesystem for the system's temporary directory */tmp*. Many XENIX programs and applications use this directory for temporary storage, so, depending upon the number of users you have and how active they are, the amount of temporary space you need can vary. A comfortable margin is half a megabyte for each user who will be logged in at any time. Add more to this figure depending upon how many of those users are likely to use large databases, edit large files, or work on software development.

  Databases that are maintained on the *root* filesystem can grow very large. If you plan to use a database, allow extra room to store them.

  Finally, you should allow several megabytes of room "just in case." Systems always grow, so plan for that growth.

  Check the section "Packages In The Distribution" in these *Release Notes* to see how much room is required for the entire Operating System and how much for individual parts. Disk space requirements are usually furnished with most applications.

- There is adequate room on the */u* filesystem.

  Allow a minimum of 1.5 - 2 Mb disk space per user on the */u* file system. If you have a database that uses home accounts for file storage, more room is required, depending upon the number of people who will use the database, and how large the files typically become.

- If your disk is so large that it would be difficult to manage as a single file system.

  If your system is equipped with a very large disk, such as an 80 Mb or larger disk with most or all of it used for XENIX, it may become difficult to manage. The larger the disk, the more media is required to perform system backups, and the longer that process takes. Also, cleaning the filesystem after crashes takes longer with large filesystems. Finally, maintaining enough free space on the filesystem takes more work with very large disks.

We **do not** recommend a separate filesystem if:

- Your system is a runtime system, with a dedicated application or with little user file storage.

- The root filesystem cannot be kept large enough to maintain 10-20% of its size as free space.

2.6 Installing terminfo (M)

If you choose to install **terminfo**, which is in the TERMINF package, you see some messages that describe the installation process. You are prompted as to whether you want the **terminfo** database compiled at this time (a lengthy process). If you choose to defer this to a later time, you are given instructions for doing so.

2.7 Installing on a Serial Console

If you are installing the XENIX System V Operating System on a system that does not have a video display adapter, and you are using a serial console, you will see this message:

```
No display adapter present
```

This is a normal part of the installation.

2.8 Include Files and Utilities

Machine dependent Development System include files and utilities are included on the XENIX System V Operating System N Volumes. If you do not install the Link Kit package or the XENIX System V Development System set, you may not be prompted to extract any files off some of the N volumes.

2.9 Games Support

Games are supplied, as is, for your pleasure and enjoyment. They are not supported. On line documentation is supplied for some of the games in the directory */usr/games/doc*. Use **custom**(C) to install the Games floppy:

1. Enter **custom**.

2. Select the option to "Add a Supported Product."

3. Insert Games Volume 1 when prompted, then follow the screen prompts as they appear.

Refer to the **custom**(C) manual page in the XENIX *Reference* for more information on installing all or parts of a package.

Note

> */usr/games/doc/rogue* contains control characters on line 20 that may affect terminals or printers when this file is displayed or printed. Edit the file to remove the characters if they cause a problem.

2.10 Restoring Lost Files

If you accidentally erase one or more of the files included with the distribution after you have installed the XENIX System V Operating System, you can use **custom**(C) to restore the file, or files. The **custom**(C) utility can list, install, or remove individual files, sets within the XENIX System packages, or entire packages of the XENIX

System V complete system.

Refer to **custom**(C) in the XENIX *Reference* for information on using **custom**. You cannot use **custom** to restore any special modifications you made to a file between the time when you first installed it and the time when the file is lost. It is wise to perform regular backups of your system to prevent such losses.

3. Using Your System

3.1 csh(C) - sh(C) Conflicts

The C-shell, **csh**(C), has many built-in commands with the same name and functionality, but different syntax, than those used by the **sh**(C) shell. The **sh**(C) commands are those documented in the XENIX *Reference*. An example of two commands that exhibit this behavior are **echo** and **nice**. Refer to the manual page **csh**(C) for information on built-in C-shell commands.

An */etc/cshrc* file is included with this release. You can customize the C-shell environment for all **csh**(C) users by using this file, which is similar to */etc/profile* for **sh**(C) users.

3.2 Using csh(C) Under Heavy Load

During intervals of heavy system load on a 386 machine, repeatedly pressing the delete key while at a C-shell prompt (%) may cause the shell to exit. If **csh** is the login shell, the user is logged out.

3.3 vsh(C)

The visual shell, **vsh**(C), does not currently support the use of special characters when specifying file names. The use of these special characters (*, [,], and ?) is explained in the chapter "Basic Concepts" in the *Introduction to XENIX*.

vsh(C) resets the **swtch** character. It is not possible to switch to the session manager, **shl**(C), while running **vsh**. However, switching from **vi**(C) or **sysadmsh**(C) works. Note however, that the default switch character (control-Z) is a command to **sysadmsh**, so you may experience difficulty in running **sysadmsh** under shell layers.

In addition, it is necessary to run **vsh** as superuser and select "help" in order to initialize the help files.

3.4 shV(C)

If the **shV**(C) manual page is present, remove it from your *User's Reference*. The functionality of **shV** has been merged with **sh**(C).

3.5 vi(C)

On page 10 of the **vi**(C) man page, the page and scroll command descriptions are reversed.

3.6 mapkey(M)

The following list of files replaces the list on the current **mapkey**(M) manual page:

keys	keys.chw	keys.fra
keys.ger	keys.ita	keys.nor
keys.spa	keys.swe	keys.ukd
keys.usa		

3.7 fsname(M)

The "Syntax" line of the **fsname**(M) manual page is incomplete. It should read:

fsname [**-p**] [**-s** *name*] /dev/*device*

3.8 Introduction to XENIX, "Listing Directory Contents"

On page 4-13, the explanation of the **−F** option of **lc**(C) is incorrect. "...with a backslash (\)" should read "...with a slash (/)."

3.9 pack(C)

The description of the "**−**" option on the **pack**(C) manual page is incorrect. It should read: "If the **−** argument is used, an internal flag is set that causes packing statistics to be printed on the standard output."

3.10 crypt(C)

The **crypt**(C) command is not distributed with the XENIX System V Operating System. If you want the **crypt**(C) utility and associated **crypt**(S) libraries, and you live in the United States, contact the support center listed on the support information card included with the software.

3.11 cpio(C): -l Option

The **-l** flag is intended to link files wherever possible. If the link cannot be done, (e.g. the file already exists) the command will abort.

3.12 sort(C): -u Option

Using the **-u** option can result in random meaningless characters at the end of the sorted file.

3.13 Using Raw Devices

When using **dd**(C) or **tar**(C) with a raw device, specify the block size as a multiple of 1K, which is equivalent to 2 "b." This is recommended because **dd**(C) and **tar**(C) use 512 byte blocks while the XENIX filesystem uses 1024 byte blocks. For example, for **dd** (C) using a 9K (system) block size use:

```
dd if=file of=/dev/rfd0 bs=18b conv=sync
```

You could also use:

```
dd if=file of=/dev/rfd0 bs=9k conv=sync
```

Specify an even blocking factor with **tar**(C):

```
tar cvfb /dev/rfd0 18 file
```

You can also use the default tar settings. (See **tar**(C) in the XENIX *Reference*).

4. System Configuration and Link Kit Notes

4.1 Buffer Overflow

4.1.1 tty Buffer

The XENIX kernel buffers for unread tty input hold up to 256 characters. Overflow of a buffer causes it to be flushed and the 257th character is not echoed. If this 257th character is dropped during the output of an escape sequence, the output of that sequence is other than expected.

For example, if a function key is programmed to send "^[154g" and the function key is held down so that the output of this string on the screen repeats, the 257th string character, "4", is dropped and the string interpreted would instead be "^[15g" that one time.

4.1.2 Function Key Buffer

If **setkey**(C) is used to assign more than 512 characters to function key strings, the character buffer limit in the function key buffer is overflowed. The arrow keys are the first to stop working. To reduce unnecessary character assignments, remove the characters within quotes in the file */usr/lib/keyboard/strings*. Use the **mapstr** utility (see **mapkey**(M)) to make the string changes. Once the buffer has overflowed, you must reboot the system to reestablish the integrity of the system (i.e. invoking **mapstr** will not make the arrow keys work again).

4.2 Using 286 Device Drivers on 386 Machines

New vendor supplied device drivers are *required* for XENIX 386. 86 and 286 device drivers will not link into the 386 kernel. All the device drivers included in the XENIX 286 Operating System are included in the XENIX 386 Operating System. Third party supplied drivers must at least be recompiled to work with the 386 kernel. Contact the IHV/ISV who supplied you with the 286 device driver and direct them to your Support center so that we can help them support their device in the 386 kernel.

4.3 vmstat(C)

The command **vmstat**(C) has been added on 386 systems only. **vmstat** reports system statistics on processes, demand paging, plus cpu and trap activity. The **vmstat** manual page is distributed in the set of updated documentation included with these *Release Notes*. Place it in the correct section of the XENIX *Reference*.

4.4 master(F) Manual Page

The **master**(F) manual page states that the 000200 line in field 3 is "Not used." The manual page should state that 000200 is used for the Halt routine.

4.5 configure(C)

configure has three additional options as described here:

-j *prefix*	prints out the major device number for the driver using *prefix*.

-j NEXTMAJOR	prints out the smallest available major device number.
-r	forces a rewrite of the configuration files whether or not the command changed the configuration.

4.6 uname(S)

No XENIX utilities currently use the node name in the **uname** structure. XENIX utilities use the entry in */etc/systemid*. However, some sites may want to set the node name in the **uname** structure. The node name can be up to 9 characters long. You can set the node name by using the Link Kit to relink a new XENIX kernel. Refer to Chapter 4 of the XENIX *Installation Guide* "Using the Link Kit".

To create the node entry in the **uname** structure, first install the Link Kit with **custom**(C), then:

> \# **cd /usr/sys/conf**
> \# **./configure**
> *select "System Name"*
> *enter name*
> *quit from configure*
> \# **./link_xenix**
> \# **mv xenix /xenix.new**
> *reboot specifying xenix.new*
> \# **uname -a**
> *verify node name appears as desired*
> \# **mv /xenix.new /xenix**
> *reboot as usual*

4.7 NOFILE - Number of Open Files

The XENIX System V Operating System Release 2.2 kernel allows up to 60 open files per process, rather than the previous limit of 20 open files. Note that this increase is in the kernel limit only, and that standard I/O routines other than those in the XENIX System V Development System Release 2.2 have the previous limit. The **open**(S) system call has the new limit of 60 since it depends only on the kernel.

You may have to reconfigure your kernel to increase certain parameters if you use programs that take advantage of this increased limit. For example, you may have to increase NINODES to raise the limit to the number of inodes open on the system at one time. See **configure**(C).

Use the Link Kit to raise any required limits. See Chapter 4 of the *Installation Guide*, "Using The Link Kit."

5. Administering Your System

5.1 badtrk(C)

If you run **badtrk**(C) on a disk that contains existing XENIX filesystems, (such as during a XENIX reinstallation) you may see an extra prompt before you see the **badtrk**(C) main menu. The message you see reads as follows:

> 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 *number* bad tracks. Do you wish to leave it unchanged? <y/n>:

If you elect to leave the limit of the bad track table unchanged, you are not prompted to change the number of reserved replacement tracks during this process. If you wish to change the number of replacement tracks, you cannot preserve existing file systems.

5.2 Intel 387 Co-Processor Problems

Because of manufacturing defects, some Intel 387 math co-processors will not operate correctly under XENIX-386. If your processor does not operate correctly, you have several options. You may remove the 387 chip and replace it with one that is known to work correctly, you may remove it and elect not to replace it, or you may bypass the chip by adding the *ignorefpu* keyword in your boot command as follows:

```
Boot
:xenix ignorefpu
```

To automatically bypass the 387 chip every time you boot your system, add the *ignorefpu* keyword to the */etc/default/boot* file. See **boot**(HW) for more information.

5.3 Using an NEC Businessmate or Powermate 286 in 10Mhz Mode

The default power-up cpu speed of the NEC Businessmate and Powermate computers is 8MHz. To use the 10MHz cpu speed of these machines, add the keyword *nec10mhz* to the boot command as follows:

```
Boot
:xenix nec10mhz
```

To automatically use the 10MHz speed every time you boot your system, add the *nec10mhz* keyword to the */etc/default/boot* file. See **boot**(HW) for more information.

5.4 Using an Intel Inboard in High Speed Mode

The Intel Inboard is a plug-in card for a 286 AT that replaces the 286 cpu with a 386 cpu. To utilize the high-speed, cache mode of the Intel Inboard, add the keyword *inboard* to the boot line:

Boot
: xenix inboard

To automatically use this mode of the inboard every time you boot the system, add the "inboard" keyword to the */etc/default/boot* file. See **boot**(HW) for more information.

5.5 autoboot(M) and System Console

The text describing the SYSTTY=*x* entry for **/etc/default/boot** is incorrect and should read:

SYSTTY=*x* If *x* is zero (0), the system console device is set to the main display adapter. If *x* is one (1), the system console is set to the serial adapter at COM1.

Note that the serial console must be on a standard COM1 card, not a multiport card or one that requires the installation of special drivers.

5.6 mkdev(C)

In the first paragraph of the "Description" section of the **mkdev**(C) manual page, the sentence: "*mkdev* may call *lpinit*(C), *hdinit*, *serinit*, *fdinit*, *fsinit*, *tapeinit*, *ominit*,*or shlinit*" should read: "*mkdev* may call *lpinit*(C), or any script in the directory **/usr/lib/mkdev**."

5.7 termio(M) and stty(C)

There are two new extensions to the *c_cflag* field which is documented on the **termio**(M) manual page:

```
CTSFLOW    0020000
RTSFLOW    0040000
```

CTSFLOW and RTSFLOW enable CTS/RTS flow control. You should also set **-ixon** and **-ixany** so that these two types of flow control do not interfere with each other.

On the **stty**(C) manual page, the following two **stty** settings should have been included in the list of "Common Control Modes":

ctsflow (-ctsflow)
: Enables CTS protocol for a modem line.

rtsflow (-rtsflow)
: Enables RTS signaling for a modem line.

5.8 /etc/passwd Ownership

The ownership of */etc/passwd* file is set to the group ID of the last user to change their password.

5.9 fsck(C) and 386 Core Files

If there are 386 core files present when **fsck** is run, the message "POSSIBLE FILESIZE ERROR" is displayed. This is normal and should be ignored.

5.10 Clockrate Adjustment

If you have an AT&T 6300, you may need to reset your clockrate, since these computers operate at a different frequency than the default setting.

You can check the accuracy of your clock rate using the **date**(C) command over a period of days. If the system clock gains or loses time excessively, refer to the manual page **clockrate**(HW) for information on setting the clock rate.

The correct clock rate for an AT&T 6300 is 1.22878.

5.11 swapctl(C)

The **swapctl** utility listed in Contents and Indexes in the XENIX *User's Reference* is not supported in this release and there is no **swapctl** page.

5.12 /etc/init

/etc/init can only be executed by the kernel as process 1. It cannot be invoked from a shell prompt.

For information on changing **init** states, see **disable**(C), **enable**(C), and **init**(M). Another method of changing **init** states is provided in release 2.2 by **telinit**(C). See **telinit**(C) in the XENIX *Reference* for more information.

inittab(M) is provided for users more familiar with the **telinit** approach to terminal administration, as opposed to the standard XENIX **enable**(C) and **disable**(C) approach. It is intended that a full intergration of these two approaches will be provided in a future version of XENIX.

5.13 /dev/tty[11,12,13,14]

Serial lines */dev/tty11, /dev/tty12, /dev/tty13*, and */dev/tty14* are no longer linked to the serial I/O devices */dev/tty1a, /dev/tty1A, /dev/tty2a*, and */dev/tty2A* respectively. These links were maintained for backwards compatibility with **uucp**(C) and shell scripts. Please note that if you use older versions of **uucp**(C), you may have to alter certain files because of this change.

/dev/tty11 and */dev/tty12* are now **multiscreens**. The next section describes these new multiscreens.

5.14 Additional Multiscreens - tty11, tty12

The maximum number of multiscreens has been increased to twelve. If your system has enough memory, it is automatically configured for up to twelve multiscreens. You can also reconfigure the system with the Link Kit as described in Chapter 4 of the XENIX *Installation Guide*, "Using The Link Kit."

You can use these new multiscreens just like the other ten if your keyboard has the additional function keys F11 and F12, and they are compatible with the standard enhanced keyboard F11 and F12. If you have fewer function keys than the number of multiscreens enabled, or a "non-compatible" keyboard, you can access the other screens by stepping past the last multiscreen with the Ctrl–PrtScr key.

For example, if your keyboard has ten function keys, but you have twelve multiscreens enabled, you can access the eleventh and twelfth screens by first going to the tenth screen, by pressing Alt-F10, then pressing Ctrl-PrtScr to step to the eleventh screen. If you press Ctrl-PrtScr again, you can access the twelfth screen, and if you press it once more, you are back to the first screen.

For general information on multiscreens, see **multiscreen**(M) in the XENIX *Reference*.

5.15 shl(C) Notes

If you have problems with shell layers, **shl**(C), on some terminals, but not on others, or **shl** works on your console, but not on another terminal, you may have to set **istrip** for that terminal line:

stty istrip

XENIX 286 and XENIX 386 are initially configured for one shell layer session at a time. This means that XENIX defaults to allow only one user to access shell layers. You can use **mkdev shl** to change this single session limit. The script will prompt for the number of sessions desired and will relink the kernel. The new limits will become effective after rebooting, as the script instructs.

5.16 dtype(C)

The **dtype**(C) utility reports incorrect filesystem versions (e.g., System 3.x versus System V).

5.17 Creating Boot and Root Floppies

You can now create boot and root floppies with the **mkdev**(C) utility. To do this, use the following command:

mkdev fd

You can create bootable or root file system floppies with 48tpi drives. With 96tpi 5.25 inch and 3.5inch 720KB drives, you can create combined boot/root floppies.

5.18 ULIMIT

The default ULIMIT has been raised for the XENIX System V Operating System Release 2.2. It is now 2,097,152 blocks (1 gigabyte).

5.19 shutdown(C) and Shell Layers

shutdown(C) does not work if **ps** and **who am I** do not report the same tty name, for example from within shell layers (**shl**(C)). Do not run **shutdown**(C) from within shell layers.

5.20 Using backup(C) and restore(C) with floppies

If the default archive medium specified in */etc/default/backup* or */etc/default/restor* is block structured, (i.e. floppy disk) then the volume size in Kbytes must be specified on the command line. Neither utility works correctly without this information. For example, using the default device (below) with the **backup** command, enter the following:

backup k 360

The default device entry for */etc/default/backup* (tape=/dev/*xxx*) and */etc/default/restor* (archive=/dev/xxx) is */dev/rfd0* in your distribution.

5.21 Software Reboot

The XENIX System V Operating System Release 2.2 allows software rebooting, so you do not need to power off to reboot. The following message is displayed after a system shutdown (using **shutdown**(C) or **haltsys**(C)):

** Safe to Power Off **
–or–
** Hit Any Key to Reboot **

Fixed disk heads are "parked" (locked) when this message is displayed.

A new command, **reboot**, causes the system to automatically reboot after shutting down, without waiting for any keyboard input.

Note

Note that not all hardware supports this software reboot. Some machines must be powered off and on again for all the device controllers to be properly initialized.

Note

When the floppy drive door is closed, the computer attempts to boot from a floppy. The floppy drive door must be open to boot off the hard disk.

5.22 Using Altos 386 Binaries with XENIX 386

Altos XENIX 386 binaries are *not* compatible with SCO XENIX 386. This is because Altos did not use the latest compiler from Microsoft, and they do not use the same system call interface that has been adopted to ensure binary compatibility with future versions of XENIX. In most cases the binaries run but there are situations where alignment of structures and system call arguments are not the same between Altos XENIX 386 and all other XENIX 386 systems. Altos plans to correct this problem.

5.23 Large Model restor

A large model version of **restor**(C), **restorL**, is now included for 286 machines as part of the BACKUP package. This allows the use of the **X** key option of the **restor** command, which enables operations on large directory trees without running out of memory. The **restorL** utility is a 286-specific command; 8086 and 80386 machines should use **restor** only.

5.24 Loading adb(CP) Without the Development System

adbL (for the 286) and **adb** (for the 386) are provided as part of the Operating System distribution in order to apply kernel patches as described in "Keyboard Lockup." To load **adb** or **adbL** on your system, insert the N1 floppy into your drive and enter the following commands (substitute **adb** for **adbL** if your machine is a 386):

```
# mount -r /dev/fd096 /mnt
# cp /mnt/bin/adbL /bin
# umount /dev/fd096
```

If your system is an IBM PS/2 (286 or 386), **adb** is distributed on the N4 floppy in tar format. To load **adb** on a PS/2 system, insert the N4 floppy into your drive and enter the following commands:

```
# cd /
# tar x ./bin/adb*
```

Please note that **adb** and **adbL** are not supported for general programming and debugging unless you have purchased the Development System.

6. Using DOS

6.1 DOS Utilities and /tmp files

All DOS utilities leave temporary files in */tmp*, regardless of whether or not the utility executed successfully. These files are removed at the next reboot.

6.2 format(C) and DOS

The **format**(C) command is a low-level formatter for XENIX, and does not format floppies for use under DOS. To format DOS floppies, use the **dosformat** command. See **dos**(C) in the XENIX *Reference*.

Note that neither **format** nor **dosformat** can be used to format the hard disk.

The XENIX System V Operating System requires error-free floppy disks.

6.3 DOS/XENIX Coexistence

The XENIX System V Operating System supports the coexistence of MS™-DOS and XENIX on the same hard disk. Some versions of DOS have restrictions, for example, ITT DOS releases previous to 3.10 cannot share the disk with XENIX or MS-DOS. For these releases, XENIX must occupy the whole disk.

As another example, an ITT MS-DOS+ release 3.20 hard disk partition should not be made larger than 32 Mbytes. Activating a larger partition will corrupt the MS-DOS+ ending cylinder. XENIX **dos**(C) commands may not work when accessing an ITT MS-DOS+ release 3.20 hard disk partition that is 32 Mbytes or larger.

Likewise, some versions of NCR DOS only recognizes 32MB partitions, you see the message "No operating system on fixed disk" when attempting to boot a DOS partition larger than 32MB.

Some versions of DOS (for example, Olivetti DOS) put the boot block in track 0 very close to the end of the masterboot block when installed starting in cylinder 0. If you have a problem with the DOS partition not being accessible after installation, start the DOS partition on cylinder 1 instead of cylinder 0.

Whenever you use DOS and XENIX on the same disk, if in doubt, install DOS first, then install XENIX.

7. Using Communications and Networks

7.1 uucp(C) and cu(C) Notes

This section answers questions you may have about **uucp**(C) and **cu**(C). The installation of the **uucp** programs and utilities is optional. Refer to the chapter "Building a Micnet Network" in the XENIX *Operations Guide* and the chapter on "Building a Communications System" in the XENIX *User's Guide*. Also refer to **custom**(C) to install (or remove) communications files.

7.1.1 Autodialing

In addition to the program on the **dial**(M) manual page, */usr/lib/uucp/dial.c*, the C language program for the distributed */usr/lib/uucp/dial* is included in this release. You can copy and modify *dial.c* or you can use this as an example to write programs to use other kinds of modems. The makefile distributed in

/usr/lib/uucp should be modified to reflect your own dialer program. You must have the XENIX Development System in order to compile C programs. **dial** programs cannot be shell scripts.

If you create a *dial* program for another modem, please send us the source. User-written *dial* programs will be considered for inclusion in future releases.

7.1.2 Communicating with 3.0 XENIX Systems

If you have trouble connecting to a computer running XENIX System 3.0 with **uucp**(C), try making that system "active" and the computer running XENIX System V the "slave".

7.1.3 L-devices and L.sys

We recommend running **uucp**(C) direct connections at 2400 or 4800 baud so that the system's performance is not impaired. These baud rates are specified in the files */usr/lib/uucp/L-devices* and */usr/lib/uucp/L.sys*. This can be done by specifying the *dialer program* instead of the *call unit device features*.

Do not use tabs as field separators in the *L.sys* file.

7.1.4 Using Modems with XENIX

Refer to the XENIX *Operations Guide* chapter "Using Peripheral Devices" for information on modem settings, modem control, aliasing alphabetic characters and dialing with your computer.

7.1.5 Restriction in the USERFILE

USERFILE used in this release must start with the line:

```
uucp, /
```

followed by at least one other line.

Because the **uucp** login can be accessed by many users, some of whom may be undesirable, you should not use "uucp " as the **uucp** account. To ensure proper security, create a different account with **uucico** as its login shell. Many systems use the **uu***machine_name* convention as the user login, where *machine_name* represents the system's machine name.

7.1.6 uuinstall(C)

uuinstall(C) may not allow certain features such as variable baud rate to be configured. Note that the *L.sys* and *L-devices* files may have to be edited with a text editor to use new features.

7.2 Using XENIX-NET

On 286 machines, you cannot use versions of XENIX–NET previous to 1.1.1 with the XENIX System V Operating System Release 2.2. On 386 machines you must have XENIX–NET 1.2.0.

8. Using Printers

8.1 imprint(CT)

References to **imprint**(CT) are incorrect. The **imprint** command is in the (C) commands group.

8.2 parallel(HW)

The **parallel** man page currently indicates that some machines allow the use of two parallel devices, "**/dev/lp0** and either **/dev/lp1** or **/dev/lp2**." This is is only half-correct, as the devices available vary from machine to machine, and may instead allow "either **/dev/lp0** or **/dev/lp1** and **/dev/lp2**." Check the boot message for the devices recognized on your system.

The **parallel** page also includes the stty command:

```
stty eol "^J" < /dev/lp0
```

This should read:

```
stty time 10 < /dev/lp0
```

In addition, when given from a prompt, this command will only work if the port is open. It is recommended that a variation of this command be placed in the interface script used with the parallel device to achieve the same results:

```
stty time 10 0 < &1
```

8.3 lp(C) and lpr(C)

lp(C) and **lpr**(C) will only print files that are publically readable, even when printed by the file's owner. The file's directory and all directories in the path must also be publically readable. The following are two possible workarounds:

pr *filename* | **lpr**

cat *filename* | **lpr**

8.4 lpstat(C): -p Option

The command **lpstat -p** *list* fails to display status of printers.

8.5 IMAGEN printers - ips(M)

When using an IMAGEN printer in parallel printer mode (using **ipbs**, see **ips**(M)) you must specify the quote character as hex 2 (0x02 or control-b) and the EOF character as hex 4 (0x04 or control-d). Control characters must be "taken as is". Refer to the IMAGEN system manuals provided with the printer, for information on specifying these characters in the printer configuration.

8.6 Slow Parallel Printers

If you have a parallel printer that prints abnormally slow, check your configuration according to the procedure described in Appendix B: "Common Installation/Configuration Questions and Answers." If it still prints slowly, there are two options for adapting the driver to suit your printer: "Polling Operation" and "Ignore Deselect Operation."

8.6.1 Polling Operation

If your printer is slow, the parallel port hardware may be missing interrupts. A solution is to alter the way that the hardware and the printer driver communicate. The parallel printer driver can be made to "poll" a parallel port. This way the driver does not rely on interrupts from the parallel port, but this may cause a drain on system resources.

To set up polling for a parallel port/parallel printer, create what is known as a "special device node." Log in as **root** (super-user) and enter one of the following sets of commands. (Note which printer ports are recognized during the boot up message.)

For lp0:

mknod /dev/lp0p c 6 64
chown bin /dev/lp0p
chgrp bin /dev/lp0p
chmod 222 /dev/lp0p

For lp1:

mknod /dev/lp1p c 6 65
chown bin /dev/lp1p
chgrp bin /dev/lp1p
chmod 222 /dev/lp1p

For lp2:

mknod /dev/lp2p c 6 66
chown bin /dev/lp2p
chgrp bin /dev/lp2p
chmod 222 /dev/lp2p

If you are using the print spooler, run **mkdev lp** to inform the spooler of the new parallel poll device. You can choose to add a new printer or re-configure an existing printer. When you are asked to choose a device for the printer, do not use the standard parallel devices that are displayed. Instead, use: "/dev/lp0p", "/dev/lp1p", or "/dev/lp2p".

If this does not solve the problem, try the solution described in the next section.

8.6.2 Ignore Deselect Operation

If your printing speed is unusually slow, about 4 seconds per line, your printer may be deselecting itself after receiving a line of text. An **adb**(CP) patch is provided to adapt your printer driver to this type of printer. Note that there is one patch for the 286 kernel and another for the 386 kernel, and that these patches work only with the 2.2.2 release of XENIX. Instructions on how to load **adbL** (286) or **adb** (386) appear in the section, "Loading **adb** Without the Development System."

This behavior has been seen on some models of Tandy printers, but is not exclusively found on Tandy printers.

Note

The application of these patches can cause other classes of parallel printers to hang.

Before using a patch, you should make a copy of your original kernel. Enter single-user mode and type in the following commands:

```
# cd /
# cp xenix xenix.00
```

This patch can be used with either the polled or standard lp devices.

The pound signs (#) and asterisks (*) are prompts from the system shell and from **adb**; do not type them in.

286 release 2.2.2 patch

```
# adbL -w /xenix
* $x
* patime+9b?xxx
_patime+0x9b:  0x46f6  0x10fc  0xe74
* patime+9b?w 9090
_patime+0x9b:  0x46f6= 0x9090
* patime+9d?w 9090
_patime+0x9d:  0x10fc= 0x9090
* patime+9f?w 9090
_patime+0x9f:  0xe74= 0x9090
* $q
#
```

386 release 2.2.2 patch

```
# adb -w /xenix
* patime+b1?xxx
_patime+0xb1:  0x45f6  0x10f8  0xe74
* patime+b1?w 9090
_patime+0xb1:  0x45f6= 0x9090
* patime+b3?w 9090
_patime+0xb3:  0x10f8= 0x9090
* patime+b5?w 9090
_patime+0xb5:  0xe74= 0x9090
* $q
#
```

9. Using Hard Disks

9.1 Olivetti Computers and Hard Disks

Olivetti computers use the last cylinder of the Hard Disk to write User diagnostics. Therefore, allocating the last cylinder of a drive to be used with an Olivetti computer will result in the corruption of any data stored in that cylinder. Do not allocate the last cylinder of your hard disk if you have an Olivetti computer.

9.2 Adding a 2.1 Hard Disk to 2.2

Follow these steps to add an existing XENIX 286 System V Operating System Release 2.1 hard disk to your XENIX 286 System V Operating System Release 2.2 system:

- Enter single user mode.

- 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 you have your XENIX filesystems 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**.

- 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, create names for all the divisions you want to preserve. You can call them anything you want, such as *u*, *user2*, etc.

  When you have named all of the potential **divvy** divisions, enter "q" to quit from **divvy**, then "i" to install the partition table.

- When you have named all of the likely candidates in all the possible **fdisk** partitions, mount each of the filesystems in turn. For example, to mount a filesystem named */u*, enter:

  mount /dev/u /mnt

  If the **divvy** partition is not valid, the **mount** command should refuse to mount the filesystem. If the filesystem mounts, change directories to it and examine it.

  If it is not the filesystem you want to preserve, unmount it, remove it from */dev*, and try another filesystem.

  Do this until you find the filesystem(s) you want to preserve.

  Use this command for each filesystem you want to preserve:

  mkdev fs */dev/devicename /directory*

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

  mkdev fs /dev/u /u

10. Using Keyboards

10.1 setkey(C) and console(HW)

Assigning more than 512 characters to function keys on a per screen basis fills the allotted string key buffer. The sequences sent by the arrow keys are overwritten, and the arrow keys do not work, most recognizably in **vi**(C). Although the size of the **setkey** string mapping table has been increased to 512 bytes, there is a limit of 30 characters assigned to an individual key. This limit also applies when defining function keys as described on the **console** manual page. However, when using **mapstr**(C), there is no 30 character per key limit, only the total 512 byte limit.

10.2 Keyboard Lock-up

A very small number of systems experience a problem known as "keyboard lockup," where the system does not respond to keyboard input from the console keyboard. This has been investigated thoroughly and should be quite rare. This particular condition only affects keyboards that are attached to the video display adapter, not standard terminals that are attached to serial lines.

Your keyboard may be "locked up" if:

- The system console keyboard cannot be used to enter data or perform any tasks

- You cannot flip Multiscreens and the CAPS LOCK key does not turn the caps lock light on or off

- Other terminals on the system continue to work

- Printers or other devices continue to work

- The system is still running.

Keyboard lockup is similar to other circumstances, so before trying to fix a "locked" keyboard, make sure that:

- You did not accidentally enter Ctrl-S

- The "Keyboard Lock" key is not in the lock position

- The keyboard is still plugged in

- The system itself is still running.

First, make sure you did not enter Ctrl-S accidentally. Press Ctrl-Q several times and check to see if you can enter characters on the screen. Press RETURN a few times, or enter DEL.

Next, check the Keyboard Lock key, if your computer has one. It should be turned to the "unlocked" position. Also, make sure the keyboard is still plugged in to the correct socket.

Make sure the system 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 thirty seconds or so, the system is still running and is using the hard disk. Note that you cannot use other terminals and that the hard disk access light may not flash if you are in single user mode.

If you check all of the suggestions, but you still cannot use your console keyboard, try unplugging the console keyboard then plugging it in again. If this fixes the problem, it is definitely a case of keyboard lockup. If this last step does not fix the problem, you may still have keyboard lockup.

You can prevent keyboard lockup with the following fix. Note that it disables the keyboard lights (LEDs), so you should do this only if you have tried all other approaches:

1. Get the system console working, if it is not. Reboot the system if you have to and bring it up to single-user mode.

2. If you didn't reboot, log in as root on the system console and shut the system down to single user mode with **shutdown**(C):

   # /etc/shutdown su

   See **shutdown**(C) in the XENIX "Reference" for more information.

3. Once the system is in single user mode, back up the kernel:

   # cd /
   # mv xenix xenix.00
   # cp xenix.00 xenix

4. Enter this command (on 386 machines, omit the "L"):

   # adbL -w /xenix
   *** ledspresent/w 0**
   *** $q**

   This patches your kernel with the necessary fix. Again, note that it permanently disables the console keyboard lights (LEDs).

5. Shut down the system:

   # /etc/shutdown 0

6. When you see the "Normal System Shutdown" message, press any key to reboot the system. You have now fixed the keyboard lockup problem.

7. Finally, call your support center and report your problem.

11. Using Video Display Hardware

11.1 Hercules Graphics Card Support

Monochrome graphics are now supported using the Hercules Graphics Card. Most all functions and features of the new support are identical and/or supersets of those currently available with the Monochrome Adapter. The adapter device interface to the Hercules is still */dev/mono* or */dev/monochrome*.

- Control Modes - continue to use the SWAPMONO ioctl to select the Hercules as the output device.

- Display Modes - Two new ioctls have been added to support the possible Hercules adapters graphics modes.

SW_HGC_P0 - Instructs the Hercules to display page 0 at frame buffer address 0xB0000 (length of 32K) in 720x348 graphics mode. This mode is legal whenever the Hercules is installed regardless of the existence of either a CGA, PGA or EGA. Note the the frame buffer is zeroed to clear pre-existing text information.

SW_HGC_P1 - Instructs the Hercules to display page 1 at frame buffer address 0xB8000 (length of 32K) in 720x348 graphics mode. This mode is only legal whenever the Hercules is installed and there is not a CGA, PGA or EGA display adapter installed. Note that the frame buffer is NOT zeroed so as to allow applications the option of multiple text and graphics environments.

- Getting Display Modes - MCA_GET still returns all standard monochrome modes with the addition of the two new Hercules modes previously mentioned.

- Memory Mapping Modes - MAPMONO still maps the frame buffer with the following exceptions. When the Hercules is installed with an additional CGA, PGA or EGA, the standard 32K frame buffer starting at 0xB0000 is mapped, whereas when there is no additional CGA, PGA or EGA, the entire Hercules frame buffer of 64K at frame buffer 0xB0000 is mapped.

- Graphics Adapter Port I/O - MCAIO should still be used for Hercules port I/O. Three new ports are supported including 0x3b9 for configuration, and 0x3b9 / 0x3bb for a light pen.

11.2 screen(HW) - HPA

Any sequence sent to Horizontal Position Absolute (HPA) will position the cursor at column zero.

11.3 screen(HW) - Control Modes

This note is primarily for software developers.

The **screen**(HW) manual page for XENIX System V Operating System Release 2.2 lists the following commands under "Control Modes":

SWAPMONO
SWAPCGA
SWAPPGA
SWAPEGA

In the current release, these *ioctl* commands behave just as if a "change multiscreen" sequence were issued. For compatibility, we recommend that you do not use the commands listed above and that programs should swap to other adapter types by opening the devices associated with the desired adapter.

For example, if a program, started on a multiscreen associated with a monochrome adapter, is to display output on an installed CGA, the program should open */dev/color* and use the mode changing and selector mapping *ioctl*s on the resulting file descriptor. This maintains greater compatibility.

11.4 Display Adapter CMOS Note

CMOS location 0x2d is read at boot time to initialize the display adapter. If that register is non-zero, the adapter may be initialized differently than would normally be expected (i.e. the screen may be filled with random meaningless characters). Some older Compaq 286 machines may exhibit a problem with high resolution monochrome monitors due to this.

11.5 mapchan(M) and mapscrn(M)

Using 7-bit terminals with **mapchan** mapping, where the '[' (left bracket) character is mapped to some other character, can cause escape sequence processing by **termcap** and other applications to fail. If you must map the '[' character, **termcap** entries (and applications programs) can be modified to look for the mapped value instead of '['.

The functionality of **mapscrn** from release 2.1.3 has been restored to XENIX 286 and XENIX 386 to alleviate problems with emulating 7-bit terminals on the video screen.

12. Using Tape Drives

12.1 The Irwin Tape Drive

The following subsection should have appeared at the end of Section 7.3 of the XENIX *Operations Guide*, "Using a Cartridge Tape Drive."

A tape cartridge drive is a mass storage device that uses 1/4 inch tape. XENIX System V Operating System Release 2.2 supports the following Irwin mini-cartridge tape drives:

- Model 110 (10 megabyte)
- Model 125 (20 megabyte)
- Model 145 (40 megabyte)

Memory and Irwin Usage. Irwin use is directly related to the amount of physical memory installed on your computer. The more memory you have, the better the driver performs. This is because of the normal memory fragmentation that occurs in multiuser operating systems.

The Model 110 works with a minimum of 1 megabyte of physical memory. Since some 386 machines have memory which cannot be accessed by XENIX, it is possible to have 1 megabyte installed and still have insufficient memory for the Irwin drive. The amount of memory available to XENIX is displayed at boot time; take note of it and install additional memory if it is less than 1024K.

Note that the tape drive must be configured as drive 2.

We recommend that you always use the Irwin tape drive in single-user mode because there is more memory available and less memory fragmentation. Also, the system is typically less active when running in single user mode. Therefore, Irwin format and I/O operations are faster and there are fewer bad blocks on tapes formatted when running on a quiet system.

When excess memory fragmentation occurs, you see this error message on the console:

<ir>: Not enough memory for mini-cartridge; retrying...

The driver tries to access the tape drive for one minute. During this time you can either wait for processes to exit or kill processes that are using up memory. If memory fragmentation still exists after one minute, the tape command fails and you see,

<ir>: Not enough memory for mini-cartridge; open failed.

Formatting the Irwin Tape Media. You must format blank, unused, Irwin mini-cartridge tapes with the **format**(C) utility before you can use them to store data. Use this command on blank, unformatted tapes:

format /dev/rctmini

The *rctmini* device verifies the format on the tape. You should also use the verification device when making backups tapes.

The Irwin drive cannot reformat 10 or 20-megabyte tapes. These cartridges may be bulk erased and formatted normally, but the drive itself cannot reformat them. 40-megabyte tapes can be reformatted by the drive. Generally, once the format is written, it is good for the life of the media. The format only needs to be re-written if a failure occurs while formatting is in progress.

To reformat a 40-megabyte tape use the **format** command with the **-e** option:

format -e /dev/rctmini

This erases the tape completely, including all previous format information, then reformats the tape. Note that this is a lengthy process (often taking more than 90 minutes for a 40MB tape) and it cannot be interrupted once it has been started.

The **tape**(C) utility does not yet work with the Irwin (*ctmini*) device.

When using an Irwin mini–cartridge tape drive to write large amounts of data, you may see this error message after the last file has been written on the tape:

Tape uncorrectable ECC error

If this occurs after the last file has been written, the message is erroneous and can be ignored. You can check to make sure the last file was successfully written to tape.

If you see this message in the middle of writing, it is probably a valid error message. Try to read the data off of the tape to make sure the error is valid, and if it is, you should reformat the tape with the

format(C) command. Specify the **-e** option. See **format**(C) for information on formatting irwin tapes.

12.2 Tape Support in XENIX 86 Release 2.2

There is no tape support in the XENIX 86 Operating System Release 2.2.

12.3 Spurious Tape Driver Error Messages

Whenever the tape driver finishes writing to a tape, it prints an error message on the system console such as "ct cmd error (1,60)" or "ct cmd error (2,21)". This has been observed on irwin tape drives as well, but with different messages. These messages are spurious and should be ignored.

13. Special Documentation Notes

In order to make the documentation as complete as possible, this section describes errors in the documentation as well as some new features of the XENIX System V Operating System Release 2.2 that are not documented elsewhere.

13.1 Manual Page Permuted Indexes

The following manual pages are mentioned in the Alphabetized list and Permuted index but are not included in this distribution:

catimp(CT)
charmap(CT)
deco(CT)
enco(CT)
dviimp(CT)
itroff(CT)

13.2 Terminals (M)

The following terminals are among those supported under XENIX:

Tandy DT1
Tandy DT100

13.3 Replacement Manual Pages

Replacement manual pages:

diskcp(C)
dos(C)
dtype(C)
mkdev(C)
vmstat(C)
fd(HW)

are included at the back of these *Release Notes*. Please substitute these replacement pages for the pages in your reference books.

14. SVID Conformance Notes

The SCO XENIX System V Operating System meets the single volume System V Interface Definition (SVID) published in 1985 by AT&T (Select Code 307-127). SCO XENIX System V Operating System has been subjected to a stringent set of tests for the purpose of verifying conformance to SVID, and conforms to SVID functionality with the following exceptions:

Function	SVID Specification	SCO XENIX Implementation
shmat	Allows non-zero arguments.	shmaddr argument must be zero. The SVID states that shmat() should allow a non–zero shmaddr argument. The architecture of the 86 family precludes allowing attachment of shared memory at *specific* physical addresses. Because of this, XENIX System V requires that the shmaddr argument be zero to allow the system to choose the first *available* address.
shmop shmctl shmget	Shared memory operations. Return "char *shmaddr"	Because of the architecture of the 86 and 286 family of chips, XENIX System V for the 86/286 utilizes shared memory by using far pointers. The SVID "char *shmaddr" is replaced with "char far *shmaddr." XENIX 386 returns "char *shmaddr."
ptrace	Address specified as (int*).	Address specified as structure. The SVID states that ptrace() calls should fail if the 'addr' argument is not the start address of a word. Because of word alignment conventions in the 86 family of chips this failure will not occur.

These exceptions are found on the **termio** manual pages (**termio**(dev) for SVID and **termio**(M) for XENIX) System V:

<table>
<tr><th>Termio
Characters/values</th><th>SVID
Specification</th><th>SCO XENIX V
Implementation</th></tr>
<tr><td>QUIT</td><td>Ctrl-|</td><td>Ctrl-\</td></tr>
<tr><td>ERASE</td><td>#</td><td>Ctrl-H</td></tr>
<tr><td>KILL</td><td>@</td><td>Ctrl-U</td></tr>
<tr><td>38400 baud</td><td>B38400</td><td>not included</td></tr>
<tr><td>default initial baud rate</td><td>B300</td><td>B9600</td></tr>
</table>

Appendix A
Compatible Hardware

A.1 Compatible Hardware

The XENIX System V Operating System is available for many configurations of Personal Computer hardware. In general, your hardware configuration must have the original settings and boards before you install the XENIX System V Operating System. If you have added any boards, make sure that all switches are set as recommended in the manufacturer's hardware manual for that board.

This Appendix is divided into several sections. "General Compatibility Guidelines" contains general guidelines for hardware compatibility, including charts of compatible peripherals, serial cards, video adapters monitors, hard disks and controllers. System parameters necessary for these devices also appear in this section. These guidelines must be followed to ensure proper system performance.

The remaining sections include exceptions and special notes for each type of machine supported.

Note

> The specific hardware that is listed in these *Notes* should work with the XENIX System V Operating System. However, not every machine and configuration has been tested by us and proven to work. Also, because compatible machine or add-on peripheral manufacturers may change configuration, functionality, or firmware at any time, no guarantee is implied. Please write us with accurate hardware information for possible inclusion on our lists.

1.1 Using this Appendix

To find a listing of compatible hardware for your machine, you must know your machine type. You must know the processor your machine uses and if it follows the Industry Standard format. You should also know if there are enhancements to your system, such as an ESDI interface. To find the available types of machines, check the section titled "Machines" in the "General Compatibility Guidelines" section of this appendix. There is a specific section of compatible hardware in this appendix for each version of SCO XENIX. There is also a section called "General Compatibility Guidelines" that discusses general compatibility issues with SCO XENIX. Read through this section before you install any extra hardware on your system.

1.2 General Compatibility Guidelines

This section describes what hardware can be used with the standard XENIX Operating System Release 2.2 distribution. There are many other devices that you can use, which require additional vendor supplied software, that are available from independent hardware vendors. Call your dealer or software support center and ask for the third-party IHV/ISV catalogue.

If your computer is listed as a supported machine in this Appendix, it should run XENIX without adding any hardware or changing any jumper or switch settings. This appendix is provided as a reference so that you can check the compatibility of any piece of hardware you own or may wish to buy in the future.

1.2.1 Machines

The XENIX System V Operating System Release 2.2 is available for several machine configurations. Find your machine configuration in the right hand column and match it with the corresponding version of SCO XENIX in the left hand column. The XENIX kernel runs in one of three modes:

Processor	Mode
8086	Real mode (unprotected)
80286	Segmented mode (protected)
80386	Demand paged virtual memory mode (protected)

Your version of XENIX is for one of the following machines:

XENIX version	Hardware Configuration
286AT	Industry Standard 286 based personal computer.
386AT	Industry Standard 386 based personal computer.
286PS	IBM Personal System/2 Model 50 or Model 60.
386PS	IBM Personal System/2 Model 80.
286HP	HP Vectra 286 based personal computer.
286 ESDI	Industry Standard 286 based personal computer with an SMS 8620 or 8627 ESDI disk controller.
386 ESDI	Industry Standard 386 based personal computer with an SMS 8620 or 8627 ESDI disk controller.
6300+	AT&T 6300+ 286 based personal computer.
86XT	Industry Standard 8086 based personal computer.

Note

> For machines with bus or processor speeds greater than 6 MHz, check with your peripheral vendor to verify that their hardware will run under your configuration. Malfunctions with slow hardware are especially noticeable in a 386 environment.

Some computers arrive with the hard disk only partially formatted. If you have such a machine, use the DOS hard disk format command or Diagnostics diskette to format the entire disk before installing the XENIX System V Operating System.

Some computers require specific switch settings to run the XENIX System V Operating System. If your computer does not run the XENIX System V Operating System with the settings as shipped, contact your computer hardware representative for the proper settings.

1.2.2 Math Chips

Your personal computer may include the 8087, 80287 or 80387 math co-processor, which is automatically detected and supported by XENIX 86, XENIX 286, and XENIX 386 System V Operating Systems. These co-processors will improve floating point efficiency.

Use math co-processors matching your machine's CPU speed. Follow the manufacturer's recommendations.

At boot time, the XENIX System V Operating System announces the presence of a math coprocessor with the message "math coprocessor present" if an 8087, 80287 or 80387 is detected. Please note that switches on the main system board must be set properly to enable 8087, 80287 or 80387 interrupts or your system must be set up with the manufacturer's setup disk to expect the chip. Insure that the system diagnostics recognize the co-processor presence and check your hardware manual for the proper switch settings.

Please note that on some motherboards, XENIX 386 incorrectly recognizes the presence of an 80387 co-processor even if the chip is not installed. This problem is prevalent on machines that use the Intel motherboard. If your computer incorrectly recognizes the presence of an 80387 chip, make sure that blocks E48 and E49 are not connected with a jumper connection.

XENIX 386 supports both the 287 and 387 math coprocessors. Support for the Weitek 1167 is not yet available.

Some 8087, 80287 and 80387 exceptions have been masked. Refer to the manual page for **8087**(HW).

1.2.3 Memory Cards

In general, most memory cards work with XENIX. If you experience "panic: parity" errors it is often because of low quality memory chips or cards. This problem is especially prevalent with the 32-bit static RAM chips used in 386 machines.

With memory cards, check the switch settings on both the card and motherboard. Refer to the hardware manuals for your computer and for the memory card to find the correct switch settings. The XENIX 86 Operating System supports up to 640K of main memory. The XENIX 286 and XENIX 386 Operating Systems support up to 16MB of main memory.

1.2.4 Multi-Function Cards

The serial ports on many multi-function cards function as expected if COM1 and COM2 are fully compatible with the standard specifications for these serial ports. These specifications are explained in the "Serial I/O Boards" section of these *Notes*.

1.2.5 Serial I/O Boards

This section describes the conditions and results of using various serial I/O boards with the XENIX System V Operating System. Standard single port serial I/O boards function as expected if COM1 and COM2 are fully compatible with the standard specifications for these serial ports.

To configure the Operating System for the serial board you are installing, you must run the **mkdev serial** command. See "Using Peripheral Devices" of the XENIX *Operations Guide* for more information on */usr/lib/mkdev/serial*, the script called by **mkdev serial**.

Each multiport serial I/O board is unique; the XENIX System V Operating System has special driver code for each card listed. Only those with status poll registers can work with the high performance driver scheme chosen, and new boards require additional driver support. The following serial I/O boards are supported by the XENIX System V Operating System for personal computers:

AMI lamb 4 and 8 port
Arnet Controls 2, 4 and 8 port
(clock option not supported)
Arnet Twin port
AST FourPORT/XN
Computone ATvantage-X 8 port board
Control Systems Hostess 4 and 8 port versions
CTC Versanet 4AT (4 port) and 8AT (8 port)
Digiboard 4 and 8 port
IBM standard COM1 and COM2
Kimtron Quartet 4 port*
Olivetti RS232C Multiport Board
Quadram QuadPort™ 1 and 5 port
Stargate Technologies OC4400 (4 port) and
OC8000 (8 port) versions
Tandon Quad Serial Card

* The Kimtron board does not work with all 386 machines.

The following serial I/O boards are supported by the XENIX System V Operating System for the Personal System/2:

IBM 2 port cards (up to 3 per system)
Control Systems MC 8 port cards

Refer to the **serial**(HW) and **mkdev**(C) manual pages for more information on compatible serial I/O cards and on adding and enabling serial lines.

8250a Serial I/O Chip Notes

Some computers or add-on serial I/O cards use the 8250a serial I/O chip. Some revisions of this chip do not handle interrupts properly. DOS does not use interrupts, so the use of this chip with DOS causes no problems. The XENIX System V Operating System does make use of interrupts, as it is a multi-tasking operating system.

The problem with the serial I/O chip shows up when using **uucp**(C) or **cu**(C). Indications that your computer contains a bad revision 8250a chip are that **uucp**(C) may lose characters constantly and generate unkillable **uucico** processes, and that **cu**(C) at high baud rates stops executing and does not exit.

The problem rarely shows up when using the serial port with a terminal. It is associated with high-speed serial input. If you want to use **uucp**(C) or **cu**(C) and your computer has one of these chips, we recommend you replace the 8250a chip with an 8250b serial I/O chip or use a multi-function card containing a serial port and configure it as COM1 or COM2. Disable the built-in serial port or avoid high-speed input on that port.

All COM1 boards should be strapped at interrupt vector 4. All COM2 boards should be strapped at interrupt vector 3. Check your serial card hardware manual or call the hardware manufacturer for the switch settings that implement these addresses.

Note

> SCO uniPATH SNA-3270 uses interrupt vector 3, which can interfere with the use of a serial card on COM2.

Serial Card Addresses and Notes

The following table lists supported serial cards and their addresses. Note that if you install two serial cards on COM1 and COM2 that are incompatible, you may see this message at boot time:

```
WARNING: Serial boards are installed at conflicting addresses
```

The message indicates the boards do not have the correct switch settings. This check does not occur for cards installed on additional COM ports and you may not see this message even if the boards are actually incompatible.

Refer to the following serial address table to resolve conflicts and check the "Notes" following the table for more information.

When using a 286 or 386 IBM Personal System/2, configure additional serial cards at the standard addresses given by the reference diskette.

Physical Port	Board Type	Primary I/O Address	Primary Status Address	Alternate I/O Address	Alternate Status Address
COM1	AMI lamb **	0x540	0x210	none	n/a
	Arnet	0x100	0x142	0x280	0x2C2
	AST*	0x2A0	0x2BF	none	n/a
	CTC†	0x160	n/a	none	n/a
	Digiboard††	0x110	0x151	none	n/a
	Hostess	0x500	0x507	0x680	0x687
	Kimtron	0x120	0x8120	none	n/a
	Olivetti†††	0x2A0	0x2BF	none	n/a
	Quadram***	0x280	0x2D3	none	n/a
	Stargate	0x290	0x2D0	none	n/a
	Tandon***	0x2A0	n/a	none	n/a
COM2	AMI lamb **	0x2C0	0x212	none	n/a
	Arnet	0x180	0x1C2	0x300	0x342
	AST*	0x1A0	0x1BF	none	n/a
	CTC†	0x218	n/a	none	n/a
	Hostess	0x580	0x587	0x700	0x707
	Digiboard††	0x210	0x250	none	n/a
	Kimtron	0x2EO	0x82EO	none	n/a
	Olivetti†††	0x1A0	0x1BF	none	n/a
	Quadram***	0x288	0x2DB	none	n/a
	Stargate	0x190	0x1D0	none	n/a
	Tandon***	0x280	n/a	none	n/a
OTHER1	UNISYS****	0x400	n/a	none	n/a
OTHER2	UNISYS****	0x408	n/a	none	n/a

Serial Card Addresses

* Only enhanced mode is supported. Do not use the AST driver.

** Only continuous mode is supported.

*** Tandon is the only card whose I/O addresses are potentially identical with other supported cards, such as the AST and Quadram serial cards.

Since card addresses must not overlap in the same systems, if you have both a Tandon and a Quadram, the Tandon must be on COM1 and the Quadram must be on COM2.

You can use the following Quadram serial expansion cards in these configurations under the standard XENIX Operating System:

COM1	COM2
5-port	–
1-port	–
–	5-port
–	1-port
5-port	1-port

If you need to use any other configuration of Quadram cards, you must relink the kernel using the Link Kit. You also need the SCO XENIX "Software Development System" to recompile *sioconf.c.* Edit *sioconf.c* as follows:

1. This file contains several groupings of serial card descriptions. Each group is identified by the first number on the description line.

   Find the descriptions for the Quadram cards in each group and move that descriptor line to the front of the group.

2. Remake and boot the new kernel. Refer to "Using the Link Kit" in the *Installation Guide* for information on the Link Kit. Use **config −i.**

**** These serial cards only work on the UNISYS PC/IT.

† Notes for the CTC Versanet serial cards:

1. The correct switch settings for the 8AT and 4AT are:

   As a COM1 (strapped at addr 0x160, using irq4) the 8AT has:

   switches 33, 35, 36, 38, 39 & 40 OFF
   switches 34, 37 ON (i.e.: shunted)

   on the DIPSWITCH selection:

   5, 6 & 8 should be OFF
   all the others should be ON

   As a COM2 (strapped at addr 0x218, using irq3) the 8AT has:

   switches 33, 35-40 OFF
   switch 34 ON (i.e.: shunted)

   on the DIPSWITCH selection:

   1, 2 & 7 should be OFF
   all the others should be ON.

   The 4AT is the same as the 8AT in both the above cases, with the following common exception:

   switches 39 & 40 MUST BE ON (shunted)

2. These boards come in both 8250 and 16450 versions, you must have the 16450 version if you intend to run on 386. The 8250 should run on the 286.

3. The original CTC Versanet boards used different addresses. Please ask your hardware vendor for versanet boards strapping at the above addresses (0x160 and 0x218).

4. The CTC "Maomao-4" serial board is not supported.

†† Notes on the Digiboard serial card:

1. Ports for switches DS2 to DS5 (DS9 for 8 port version) must be strapped starting at the boards base address as given in the table and incrementing by 8 for each port.

   The following example is for COM1 at 110:

DS2	110
DS3	118
DS4	120
DS5	128
DS6	130
DS7	138
DS8	140
DS9	148
DS1	150

2. If COM1 is used then all the ports must be strapped as "EVEN" to interrupt request line 4 (see Digiboard documentation). If COM2 is used then all the ports must be strapped as "ODD" to interrupt request line 3 (same notation).

3. Only rev. 3 and later boards are supported.

††† Notes for the Olivetti RS232C Multiport board:

1. The factory settings will not function properly – you must alter the existing switch positions to reflect those listed in Item 4 below.

2. If you are using a single Olivetti board, you must configure it as COM2.

3. If you are using two Olivetti boards, one must be configured as COM2 and the other configured as COM1, with the COM1 port built into the M380 disabled. To disable the COM1 port built into the

M380, refer to the section on "Setting Up the System" in your Olivetti Installation and Operations Guide.

4. The correct switch settings:

As a COM1 (strapped at addr 0x2A0):

IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7	XA1	XA0	INT	SHR
off	off	on	off	off	off	off	on	on	off

As a COM2 (strapped at addr 0x1A0):

IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7	XA1	XA0	INT	SHR
off	on	off	off	off	off	off	off	on	off

1.2.6 Tape Drive/Controller Combinations

The tape drivers included in this release work with the following drive/controller combinations. Tape drivers are soon to be included with Personal System/2 XENIX. Check with your software support center for PS/2 cartridge tape drivers.

Supported Tape Controllers and Cartridge Drives

Manufacturer	Controller	Drive	Type	Notes
Archive	SC400	Scorpion 5945	A	(4)
Computone	SC400	Scorpion 5945	A	—
TI	SC400	Scorpion 5945	A	—
ITT	PC-36	5000(E)	W	—
COREtape	PC-36	5000(E)	W	—
Olivetti	PC-36	5000(E)	W	(1)
Wangtek	PC-36	5000(E)	W	—
Tecmar	PC-36	QIC-60AT	W	—
Emerald	xnx-50-2012	Cassette	E	(3)
Emerald	xnx-60-2002	Cartridge	E	(3)
Irwin	floppy	145 (40 MB)	NA	(2)
Irwin	floppy	125 (20 MB)	NA	(2)
Irwin	floppy	110 (10 MB)	NA	(2)

Default Settings

Type	DMA	Interrupt	Base Address (hex)
A	3	3	0220H
W	1	5	0338H
E	3	25	0300H

Notes

1. The Olivetti tape controller's factory jumper settings do not match the type W drive default values. To use the Olivetti drive without changing the controller card's jumper settings, use the "mkdev

tape" command to select all four of the following settings:

Type	DMA	Interrupt	Base Address (hex)
W	1	25	0288H

2. The Irwin 40-megabyte drive can use either DC2000 or DC1000 tape cartridges and the 10- and 20-megabyte drives use a DC1000 tape cartridge. A DC1000 tape written on a 125 or 145 model drive can be read by either drive.

3. Emerald drives do not work with 20mHz 386 machines at this time.

4. Archive SC400 full length boards are supported. However, the half-length SC402 board is not currently supported under XENIX.

Refer to "The Irwin Tape Drive" notes in the main body of these *Release Notes* for information on using the Irwin drive. These notes supplement the XENIX *Operations Guide* Chapter 7, "Using Peripheral Devices."

Note that these drives are sometimes sold under other brand names. Only one cartridge tape subsystem is supported per computer. (It is possible to have one cartridge tape drive plus an Irwin drive configured on a system.) QIC24 format is supported on the full size cartridges. Tape support is raw (character) only, no block device. A tape utility, **tape**(C), is provided for rewinding, erasing, retensioning, etc. **tape** does not work with the Irwin drive at this time.

To configure your system for one of the above tape units, run **mkdev tape**. If you do not choose specific values for the DMA channel, interrupt, and base address, the values listed above will be used. Note that most addresses are specified in hexadecimal, indicated with a trailing 'H'. If you do not use the default settings, watch for possible interrupt conflicts with other installed devices. The table below indicates what interrupts may be in use on your system. You should not use interrupts 0, 1, or 6, as these are already in use regardless of additional devices. The **mkdev tape** menu indicates the Irwin units as "Mini-Cartridge," and all other units are indicated as "Cartridge" units. The Irwin units are not configurable.

1.2.7 Typical Device Interrupts

Interrupt	Device
0	Clock
1	Console
2	Networks and others
3	Serial COM2
4	Serial COM1
5	Alternate Parallel Port (lp2)
6	Floppy Disk
7	Main Parallel Port (lp0 or lp1)

1.2.8 Video Adapters and Monitors

Any video adapter/monitor combination which runs with an adapter that is a true equivalent of a standard adapter should run under the XENIX System V Operating System.

The XENIX System V Operating System supports the use of two video adapters and two monitors, one of type mono and the other of type color.

Adapters which come as the default adapter in supported computers (monochrome or color monitor) will work with the XENIX System V Operating System.

We support VGA video adapters running in EGA mode.

We have used the following video adapters and monitors under the XENIX System V Operating System. Although many standard emulating cards work, these are some specific examples.

Video Cards and Monitors	
Card	**Monitor**
IBM Mono Card Hercules mono Persyst mono	standard monochrome WY-530 (monochrome)
IBM CGA Paradise Modular Graphics Card Everex Graphics Edge AST CGA Plantronics Color-Plus	standard RGB (red green blue) or composite color Wyse WY-630 (color)
IBM EGA ATI EGA Wondercard* AST EGA Wyse WY-440 (EGA card) Wyse WY-700 NEC APC-H431	IBM EGA standard RGB color standard monochrome WY-640 (color EGA) WY-700
IBM PGA	IBM PGA
IBM VGA	Model 8503(mono) Model 8512(color) Model 8513(color) Model 8514(color)

* EGA Wondercard Advanced Features Switch (DIP switch 8) must be OFF.

Wyse WY-700 High Resolution Mono Monitor. There is a known problem with the screen display going blank temporarily. This can occur on any Wyse pc286, IBM PC/AT, or AT clone with this monitor installed. This problem occurs with REV D of the WY-700 mother and/or daughter board. See ASSY # on lower left edge of board for REV level. You should replace the board with a more recent revision; request return merchandise authorization (RMA) from Wyse Technology.

AST EGA Card. When switching from graphics mode into high-res text mode, the AST card may exhibit character shifting (e.g. spaces to ampersands, etc.). It is necessary to reboot to regain the screen.

Other Non-Standard EGA Cards. Some EGA cards that are not IBM standard may not operate correctly under SCO XENIX. If your card is non-standard or not in the list of supported cards, please check with the card manufacturer for compatibility. If your non-standard card is not operating correctly, enter the command

stty C80x25

and press **Ctrl-J** to execute the command. Your screen may be blank or displaying random patterns and you may not see the characters you enter. This may correct the problem on your card. If this does not restore your screen display, you must reboot your system.

1.2.9 Add-On Hard Disks

Many hard disks, both standard and nonstandard, can be used by the XENIX System V Operating System as long as the disk controller supports the drive. The disk must interface electrically with the disk controller (usually "ST506"). Controllers supporting other drive interfaces such as RLL or ESDI will work as long as the controller presents an ST506-compatible interface to the bus.

DOS does not support non-standard disks (i.e. those not defined in the ROM Fixed Disk BIOS).

Some hard disks come from the factory only partially formatted (e.g. the Maxtor 1140 140MB). This problem may become evident during installation, when **badtrk**(M) indicates that every sector past a certain cylinder/head location is bad. You should contact the manufacturer to determine whether or not the disk is completely formatted. There are several products available that will format hard disks.

Compatible hard disk controllers are discussed in the next section.

For a standard disk: the motherboard ROM must have an entry for the disk (*type*) determined by the number of heads, cylinders, tracks per cylinder (heads), sectors per track, and other characteristic information.

Follow the manufacturer's instructions to set switches or configuration.

For nonstandard disks: The user may type in information that overrides the ROM disk configuration information during installation.

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 parameters as defined in the "Fixed Disk BIOS Parameter Table" in Section 5 of the IBM Technical Reference (AT).

Some hard disks have more than 1024 cylinders. At this time, XENIX supports a maximum of 1024 cylinders.

1.2.10 Compatible Hard Disk Controllers

Many hard disks will work with the XENIX System V Operating System. Whether or not a disk works depends upon the controller board. Here are two tests the controller must meet:

1. The disk controller is fully compatible with the standard controller for that configuration. For example, here is a table showing the XENIX configurations and the compatible controllers:

Configuration	Controller
86XT	XEBEC
286AT/386AT	WD-WA2 WD1003
PS/2	IBM ESDI IBM ST506
286ESDI/386ESDI	SMS 8620, 8627

2. No special vendor software is needed to make the controller work under DOS.

If a controller meets these tests, it might work, but if it fails these tests, it will not work.

1.2.11 Modems and Autodialing

Any standard RS-232 modem works using **uucp**(C) and **cu**(C). The default autodialer is for the Hayes Smartmodem 1200. An autodial program is also supplied for the Racal Vadic 3451, the 212, and the Hayes Smartmodem 2400. Other autodialing modems can be supported by writing a dialer program, or modifying the existing one in */usr/lib/uucp/dial.*

The "uucp(C) Notes" section of these *Release* Notes and the **dial**(M) manual page in the XENIX *Reference Manual* contain more information on writing other dialer programs.

1.3 XENIX 286AT Operating System

1.3.1 Machines

We have used the following machines with the XENIX 286 Operating System:

ACER 900
ACS ET 286 Plus
American Research Corporation (ARC-AT)
AST Premium 286
AT&T 6310
Basic Time BT/AT
Compaq Portable, Portable II, DeskPro 286, 386* and DeskPro II
Contel/CADO AT/4
Data General DASHER/286
Epson Equity III and Equity III+
Hewlett-Packard Vectra ES/12 PC
IBM PC AT (6 or 8 mHz version)
IBM 286/XT
ITT XTRA XL
MAD 286 AT
Mitsubishi MP 286
NCR PC-8 **
NEC APC IV
Olivetti M28
PC's LIMITED AT™
Sharp PC 7511
UNISYS PC/IT
Tandy 3000
Texas Instruments Business Pro
Texas Instruments System 1100
Tomcat 3200-AT
WYSEpc 286 WY-2200
Victor V286
Zenith Z200 series

* Run in 286 Mode.
** Configure the CMOS database for color monitor, regardless of the type of monitor used.

These machines have been reported to run XENIX-286, but we have not tested them:

Corona ATP
Kaypro 286i

1.3.2 Math Chips

The 80287 math co-processor chip is supported.

1.3.3 Memory Cards

We have used the following memory cards:

AMI SMART PACK 2
AST
JustRAM/AT 8MB Card (Monolithic Systems)
Quadram
Tecmar
Talltree Systems
Silicon Valley Systems
STB Rio Grande
Micron Chessmate

1.3.4 Add-On Hard Disks

As noted earlier, many hard disks can be used as long as the motherboard supports the drive, or the ROM parameters are entered at installation time. ROM entries for the IBM PC AT are listed here. Refer to your computer hardware manual for the disk types supported by other 286AT computers.

IBM PC AT Disk Types Entries on Motherboard ROM

Type	Cyls.	Hds.	Size	
1	306	4	10 MB	
2	615	4	20 MB	
3	615	6	30 MB	
4	940	8	60 MB	
5	940	6	45 MB	
6	615	4	20 MB	no precompensation
7	462	8	30 MB	
8	733	5	30 MB	
9	900	15	110 MB	
10	820	3	20 MB	
11	855	5	35 MB	
12	855	7	50 MB	
13	306	8	20 MB	
14	733	7	40 MB	
15	0	0	Reserved	
16	612	4	20MB	
17	977	5	40MB	
18	977	7	55MB	
19	1024	7	60MB	
20	733	5	30MB	
21	733	7	40MB	
22	733	5	30MB	
23	306	4	10MB	

Note that other compatible machines often have different ROM tables. Check your computer hardware reference manual for the appropriate ROM table entries for your computer.

We have used the following hard disks under the 286AT configuration:

Emerald 50, 70, 140
Maxtor 140
CDC 20, 30, 40, 70
Core

The Bernoulli Box® runs on standard AT compatibles with the device driver from SCO, purchased separately. The XENIX 286

Operating System is available on Bernoulli cartridges.

1.3.5 Compatible Hard Disk Controllers

We have used the following controllers:

Adaptec ACB-237OT RLL
Western Digital WD 1003*
Western Digital WD-WA2*
DTC WD1010 compatible
WD ESDI Controller, as in the Compaq Deskpro 386 - WD-1005

* The Western Digital controller cards use either WD1010, WD2010, or WD2020 controller chips.

The following controller cards have been reported to run with XENIX 286, but we have not tested them:

DTC 5287 (using the PAR83A Controller chip)

1.4 XENIX 386AT Operating System

1.4.1 Machines

We have used the following machines under the XENIX386AT Operating System:

ACER 1100
ALR 6010 FlexCache 386/2
Corvus 331
Compaq 386
Compaq Deskpro 386/20
Hewlett-Packard Vectra RS/16 PC
Hewlett-Packard Vectra RS/20 PC
ITT 386
Mitsubishi PC-386
NEC PowerMate 386
Noble 386 from PC Discount
Olivetti M380
Tandy 4000
Texas Instruments System 1300
Wyse 386
Zenith Z386

The Intel Inboard in an IBM PC/AT or Intel-supported 286AT XENIX compatible is also supported.

The Orchid JET 386 in an IBM PC/AT or Orchid-supported 286AT XENIX compatible has also been reported to run XENIX 386.

These machines have been reported to run XENIX 386, but we have not tested them:

NCR PC916

Note that the personality card supplied by NCR must be configured so that the Video Adapter auto-switch feature is disabled.

The XENIX 386 Operating System will run correctly only on 386 systems using a B1 (or later) stepping of the 386 microprocessor. In addition, the 386 should not have the "32 bit multiply bug" documented in the Operating System Release Notes. Usually, 386 chips marked with a "Σ Σ" are tested to be free of the 32 bit multiply problem. 386 chips marked "Σ" or "16 bit software only" do not function correctly with XENIX 386.

1.4.2 General Guidelines for 386AT Peripherals

In general, all peripherals which are supported by the XENIX 286 Operating System are supported by XENIX 386 Operating System if the peripheral is speed rated for the faster machine.

1.4.3 Video Cards

The DDGA model 25-3047 video card runs correctly in this release. The DDGA model 3045A video card is not supported in this release.

1.4.4 Memory Cards

It is *strongly* recommended that you use 32-bit memory from your machine manufacturer. 16-bit memory is much slower, and may actually degrade overall machine performance. Several manufacturers have not resolved DMA issues relating to 16-bit memory, thus such machines will not recognize 16-bit memory (e.g. Zenith). When in doubt, consult the hardware manufacturer.

REMEMBER: Certain manufacturers reserve the upper 384K of the first megabyte for DOS. On some machines, this "shadow" RAM cannot be accessed by XENIX. You must install additional memory in order to run XENIX.

Note

> Memory parity errors seem to be much more prevalent with 32-bit memory, causing the system to panic. Replacing the defective memory is the only way to solve this problem.

1.4.5 Add-On Hard Disks

In general, 286AT disk configurations work on the 386AT.

1.4.6 Serial Cards

It is also strongly recommended that multiport cards utilize 16450 serial I/O chips instead of the slower 8250 chips. If you see a "double echo" problem, particularly on multiport cards, it is due to slow serial I/O hardware.

1.4.7 Compatible Hard Disk Controllers

We have used the following controllers:

Western Digital WD 1003*
Western Digital WD-WA2*
Western Digital WD 1005
DTC WD1010 compatible
DTC WD2010 compatible

* The Western Digital controller cards use either WD1010, WD2010, or WD2020 controller chips.

1.4.8 Floppy Drives

Machines configured with 3.5 inch media as drive 0 must use a software distribution on 3.5 inch floppies.

1.5 XENIX 286PS Operating System

1.5.1 Machines

We have used the following machines under the XENIX 286PS Operating System:

IBM Personal System/2 Model 50
IBM Personal System/2 Model 60

Note that you should not use the XENIX 286PS Operating System on the IBM Personal System/2 Model 80. Use only the SCO XENIX 386PS Operating System.

1.5.2 Serial Cards

We have used the following Serial cards under the XENIX 286PS Operating System:

IBM Personal System/2 model 3033 dual async card (up to 3 per system)
Control Systems Hostess/MC (8 port card)

1.5.3 Video Cards and Monitors

We have used the following Video cards under the XENIX 286PS Operating System:

IBM Personal System/2 VGA adapter and the 8503, 8512, and 8513 monitors.

1.5.4 Tape Drives

Personal System/2 tape support is being added as Personal System/2 tape devices are just becoming available. Call your software support center for tape driver availability.

We have used the following tape drives with XENIX-286PS:

Mountain Computer FileSafe Tape controller board
Mountain Computer FileSafe Model 7060 Tape drive
Mountain Computer FileSafe Model 7120 Tape drive

1.5.5 Other Hardware

We have used the following hardware under the XENIX 286PS Operating System:

IBM Personal System/2 internal modem model 0349

1.6 XENIX 386PS Operating System

1.6.1 Machines

We have used the following machines under the XENIX 386PS Operating System:

IBM Personal System/2 Model 80

1.6.2 Serial Cards

We have used the following Serial Cards under the XENIX 386PS Operating System:

IBM Personal System/2 model 3033 dual async card (up to 3 per system)
Control Systems Hostess/MC (8 port card)

1.6.3 Video Cards and Monitors

We have used the following Video Cards under the XENIX 386PS Operating System:

IBM Personal System/2 VGA adapter and the 8503, 8512, and 8513 monitors.

1.6.4 Tape Drives

Personal System/2 tape support is being added as Personal System/2 tape devices are just becoming available. Call your software support center for tape driver availability.

We have used the following tape drives with XENIX-386PS:

Mountain Computer Filesafe* Tape controller board
Mountain Computer Filesafe* Model 7060 Tape drive
Mountain Computer Filesafe* Model 7120 Tape drive

1.6.5 Other Hardware

We have used the following hardware under the XENIX 386PS Operating System:

IBM Personal System/2 internal modem model 0349

1.7 XENIX 286HP Operating System

For XENIX 286HP configuration information, consult the documentation included with your machine.

1.7.1 Machines

The minimum machine configuration that can run the XENIX 286HP Operating System is:

Hewlett-Packard Vectra model 45945A

1.7.2 Memory Cards

In addition to standard PC/AT memory, the following HP memory cards work:

Hewlett-Packard 1/2 MB (45973A)
Hewlett-Packard 1 MB (45974A)

1.7.3 Serial I/O boards

In addition to the previously mentioned serial cards, the following HP cards work:

Hewlett-Packard Serial Card (24540A)
Hewlett-Packard Dual Serial/Parallel Card (24541A)

1.7.4 Video Cards and Monitors

In addition to the previously mentioned video cards and monitors, the following HP video cards and monitors work:

Hewlett-Packard Multimode Video Adaptor Card (45981A)
    Hewlett-Packard Monochrome Monitor (35731A)
Hewlett-Packard Color Video Adaptor Card (45984A)
    Hewlett-Packard Color Monitor (35741A)
Hewlett-Packard EGA Video Adaptor Card (45983A)
    Hewlett-Packard EGA Monitor (35743A)

1.7.5 Other HP Hardware

The following terminals have been verified with XENIX-286-HP:

Hewlett-Packard 2392
Hewlett-Packard 2393

Termcap entries have been included to support the HP 2392 and 2393 terminals with the following *termcap* names:

Attribute	name
Standard HP mode	2392
HP mode without standout and underline	2392ne
ANSI mode	2392an (supported by most applications)

Set the TERM environment variable to the appropriate name.

The following printers are supported as standard print devices with XENIX-286-HP:

Hewlett-Packard LaserJet™ Serial/Parallel (2686A)
Hewlett-Packard ThinkJet™ Serial/Parallel (2225D/2225C)
Hewlett-Packard QuietJet™ Serial/Parallel (2227A)

1.7.6 Compatible Hard Disk Controllers

The following table lists the hard disk controllers that have been verified with the XENIX 286HP Operating System:

Manufacturer	Model#	Comments
Hewlett-Packard	45815A	supports 45817M 40MB disk and some third party ST-506 hard disk drives
Hewlett-Packard	45816A	supports 45816M 3 1/2" ruggedized disk
Hewlett-Packard	45895A	Western Digital 2010 compatible hard disk controller

1.7.7 Add-On Hard Disks

The following hard disks have been verified using the HP 45815A controller:

Drive	Description	Jumper Setting	Notes
Priam® 60	60Mb full height	DBC	
Priam® 40	40Mb full height	AEF	
Seagate			
ST-4051	40Mb full height	AEF	
ST-225	20Mb half height	DEF	
Maxtor® XT1085	68Mb full height	AEC	Mounting bracket may have to be slightly bent

The following hard disks have been verified using the HP 45895A controller:

Drive	Description	Disk Type
Seagate	20Mb	2
Seagate	40Mb	17
Priam®	60Mb	12
Maxtor®	68Mb	4
Maxtor®	140Mb	9
CMI	20Mb	2
CMI	30Mb	3
CDC	20Mb	2

The following hard disks types have not been verified but may work with the HP 45895A controller:

Type	Cyls.	Hds.	Size	Precompensation
1	306	4	10 Mb	128
2	615	4	21 Mb	300
3	615	6	32 Mb	300
4	940	8	65 Mb	512
5	940	6	49 Mb	512
6	615	4	21 Mb	none
7	462	8	32 Mb	256
8	733	5	32 Mb	none
9	900	15	117 Mb	none
10	820	3	21 Mb	none
11	855	5	37 Mb	none
12	855	7	52 Mb	none
13	306	8	21 Mb	128
14	733	7	44 Mb	none
15	0	0	Reserved, do not use.	
16	612	4	21 Mb	0
17	977	5	42 Mb	300
18	977	7	59 Mb	none
19	1,024	7	62 Mb	512
20	733	5	32 Mb	300
21	733	7	44 Mb	300
22	733	5	32 Mb	300
23	306	4	10 Mb	0

1.8 XENIX 286/386AT ESDI Operating System

1.8.1 Introduction

This section describes the hardware installation and software configuration of SMS OMTI 8620 and 8627 disk drive controllers under SCO XENIX System V Release 2.2 for 286 and 386 personal computers. There is also a note for system integrators on performing low-level formats on new disks, and some discussion on badtracking and disk space usage.

This product supports one SMS OMTI 8620 or 8627 controller with up to 2 drives attached to it. Either or both drives may have an ESDI or ST506 drive interface. This product provides an OMTI-specific device driver to supplement the Western Digital-specific device driver in standard SCO XENIX System V 286 and 386 Release 2.2.

1.8.2 General Notes

You must know what BIOS is in use on your controller card before you install the disk system. To check your BIOS type, find the BIOS chip on the controller card. Most manufacturer's BIOS chips are found near the bus connectors in space U37. Do not confuse this chip with the OMTI firmware chip which resides near it on the board in space U31. Read and note the model number on the BIOS chip. You need to know this number to accurately set the jumpers on your controller card. Tables are provided later in this section that specify the jumper settings for each different kind of BIOS.

Before XENIX can be installed on a system equipped with an 8620 or 8627 disk controller, you must run your manufacturer's **setup** program and set the computer up for operation without a hard disk. This is because the OMTI firmware supercedes the instructions on the normal CMOS systems for hard disks. After you run the setup utility, the installed disk(s) may need to be formatted with the **sfmt** low-level formatting routine, which resides in the OMTI firmware. Low-level disk formatting is usually performed on bundled systems before delivery. If this formatting has not been done, you must format the disk before installing XENIX.

1.8.3 Standard Installation

XENIX cannot be installed if the disk is in need of low-level formatting or if defective tracks have not been accounted for. Check to see if XENIX ESDI displays the following message when you boot XENIX:

disk[E]drive 0: cyls = ...

The "E" label indicates that you can use the **sfmt** utility to format your disk, if it is not formatted already. (If you do not see this label, do not use **sfmt** to format your disk.) You are redirected to **sfmt** if you attempt to modify disk parameters through **dkinit** at installation time. You will see:

Please use sfmt to modify disk parameters

Reboot, then invoke **sfmt** at the "Boot :" prompt to do low-level formatting, non-standard disk parameter initialization, and initial processing of manufacturer-supplied defect lists. If you do not run **sfmt** at this time, you will have to restart your installation procedure and run it before you can successfully install XENIX. **sfmt** should also be used for non-standard disk support.

The defect-processing algorithm reduces the total number of disk cylinders available for use by XENIX. **dkinit** will display the new total.

If the "disk[E]" banner is not displayed, do not use **sfmt** to format your disk. Instead, please contact your controller's vendor for low-level formatting information if you need to format your disk(s).

Enter "sfmt" at the boot prompt if your disk requires low-level formatting:

```
Boot
: sfmt
```

You see:

This FORMAT routine will DESTROY ALL data on your disk!

Press <RET> to proceed or <ESC> to cancel...

Next you are prompted to enter the number of the drive you are formatting. You see:

Enter drive # (0 or 1):

If there is only one hard disk, enter 0. Next you see:

Use default parameters (Y/N)?

If your disk is non-standard, answer "N". You must now know the parameters for your disk. If you answered "no" you are prompted to enter the total number of cylinders and heads for your disk. You see:

Total CYLS:

Total HEADS:

Next you see:

Write Precomp CYL CONTROL BYTE: 2

sfmt supplies "2" as a default but you can overwrite this answer. Check your manufacturer's specification for the correct value. Next you see:

Press <RET> to proceed or <ESC> to cancel...

If you wish to proceed, press RETURN. Next you see:

Logical partitioning desired (Y/N)?

Enter "N". Next you see:

Any defects (Y/N)?

If you do not need to manually enter any badtracks, press "N". Enter "Y" if you need to add badtracks. You see:

(Press <RET> to end defect list)
CYLINDER:
HEAD:

When you have entered all known defects in the hard disk by cylinder and head numbers, press RETURN at the

CYLINDER:

Prompt. You see:

More entries (Y/N)?

Enter "N". Next you see:

Press <RET> to proceed or <ESC> to cancel...

Now you are prompted to set the interleave factor for your hard disk. Disk speed performance may be severely reduced if your interleave factor is set incorrectly. Check your manufacturer's documentation carefully for the correct interleave factor. You see:

Interleave (1-15):

Note that if your disk supports 35 sectors per track, you should not use an interleave value of 5 or 7. Next you see:

Are you SURE you want to format (Y/N)?

Enter "Y" if you are satisfied that all your answers are correct. You see:

Formatting.....

When **sfmt** is finished formatting the disk, you see:

Formatting complete,
params saved,
Hit any key to reboot

Press any key and you see the XENIX boot prompt again:

Boot
:

Press RETURN to boot XENIX.

dkinit

dkinit is primarily for unusual or non-standard disks. Unless you know your disk is non-standard, assume that it is standard. When **dkinit** is run during XENIX installation, choose option 1 to display the current disk parameters. If these parameters do not correspond to the hard disk you have, you must modify the current parameters. If the "disk [E]" banner was displayed at power-up, please use **sfmt** to modify these parameters. Otherwise, follow the instructions provided in the XENIX *Installation Guide*.

Badtracking

The number of cylinders used for bad track alteration will be subtracted from the total number of cylinders available to the user.

After you have finished scanning for badtracks during the installation process, examine the badtrack table. If the "disk[E]" banner was displayed at power-up and you ran **sfmt**, check to see if any of the flaws in the manufacturer's disk flaw map are displayed in the XENIX badtrack table. If no bad tracks are shown, **sfmt** located all the bad tracks in the flaw map and you do not need to enter them from the list supplied with the disk. If all the bad tracks from the supplied list appear, XENIX **badtrack** located the flaws and you do not need to enter them from the supplied list. If, however, only some of the listed bad tracks are displayed, you must enter any bad tracks that are listed on the supplied bad track list but *not* displayed on the screen. If the "disk[E]" banner was not displayed, you must enter all of the flaws in the manufacturer's supplied list that are not already displayed in the badtrack table. The following table outlines all possible options and the appropriate actions:

Display	Action
No bad tracks displayed	Do not enter any bad tracks
All bad tracks displayed	Do not enter any bad tracks
Some bad tracks displayed	Enter any bad tracks not displayed
You did not run sfmt	Enter all bad tracks on the supplied list

Use XENIX badtracking to process defects that need to be added to the bad track table at a later date. Allocate space for future bad tracks even if none are found during the initial scan. Keep in mind that if you do not allocate enough space now, and your table fills up later, you must reinstall to add additional bad tracks.

1.8.4 Hardware Installation and Initialization

The next few sections are for those system integrators and experienced users who are assembling a computer system from scratch, and who are installing disks themselves.

1.8.5 DMA/Programmed I/O Considerations

The disk controller supports three different I/O modes: DMA, simple programmed I/O and optimized programmed I/O. DMA is Direct Memory Access mode, simple programmed I/O is slow but will work on virtually all machines, and optimized programmed I/O is fast but may not work on all machines. Under XENIX-286, DMA mode is the default. Under XENIX-386, simple programmed I/O is the default mode. After you install XENIX, you may switch between the different modes by patching the kernel in single-user mode using **adb** (**adbL** if your machine is a 286):

```
# adb -w /xenix
* dkio_type /w n
* $q
#
```

The pound signs (#) are prompts from the system shell and the asterisks are prompts from **adb**; do not type them in.

In the patch above, *n* is *1* for the default DMA mode, *2* for simple programmed I/O mode, and *3* for optimized programmed mode.

The default for XENIX-386ESDI is programmed I/O. The default for XENIX-286ESDI is DMA.

Standard personal computers that support I/O channel ready will work with the 286/386 ESDI configuration of XENIX System V Release 2.2 in the optimized programmed I/O mode with no problems. Machines that do not support I/O channel ready will invariably crash, with a high probability of data corruption. If you are unsure as to whether your machine supports I/O channel ready, consult your computer's manufacturer.

1.8.6 Interleave Factors

A 1:1 interleave factor is supported. To change interleave factors, refer to the **sfmt** instructions.

1.8.7 Drive Size Limits

The limitations for attached disks are 1024 cylinders (10 bits of cylinder addressing) and 16 heads, due to the **fdisk** structure shared by all operating systems. You can attach larger disks, but you cannot use more than 1024 cylinders, even with multiple **fdisk** partitions; the excess storage space is unaddressable.

Setting Up a One-Drive System

1. Cabling Requirements: One (1) 34-pin straight-through cable. One (1) 20-pin straight-through cable.

2. On the hard disk: Install drive select jumper to lowest Drive Select (DS0 or DS1).

3. On the controller: Install jumpers W20 to W23 according to the information in the "Drive Jumper Settings" table provided in the "Drive Tables" section on the 286/386AT ESDI Operating System. To find the correct drive table for your hardware, note your board type and BIOS type and find them on the following table. The BIOS types are listed across the top of the table and the board types are listed down the left side. To find the correct table to use, check the listing at the intersection of your BIOS and board types. For example, If your BIOS type is 1002579 and your board type is 8620, you can find your correct board settings in Table A.

Board Types	BIOS Types			
	1002579	1002580	1002661	1002662
8620	Table A	‡	Table C	Table D
8627	‡	Table B	‡	‡

‡ *indicates a drive/BIOS combination that does not exist.*

If your drive is a non-ESDI drive type and it does not correspond to the default ST506 drive type(s), enter disk characteristics as in step 10.

4. Remove the existing hard disk controller and unplug the connecting cables. If a separate floppy drive controller is being used, remove it and connect the 34 pin floppy cable to the J1 connector on the ESDI controller.

5. The number of hard disks in the system must be reported as zero (0) when initializing the system configuration parameters using the **setup** utility provided by your computer's manufacturer.

6. Install the 34-pin drive interface cable to the J2 connector. Install the 20-pin data cable to either the J3 or J4 connector.

7. Install the ESDI controller in any available slot on the motherboard. CAUTION: Power must be off!

8. Use the manufacturers **setup** utility to change your system settings to indicate that there is no hard disk used with your system.

9. If you need to format your disk, type **sfmt** at the *Boot :* prompt. You must know the hard disk parameters before you invoke **sfmt**.

10. Answer all questions as prompted. Note: If your drive is not listed in the BIOS drive table answer "*N*" to "*Use defaults (Y/N)?*" prompt. Then, enter the correct drive characteristics. If your drive is a non-ESDI drive type and it does not correspond to the default ST506 drive type(s), you are prompted to enter disk characteristics.

11. At the completion of this installation procedure, you will be directed to reboot the system. After doing so, continue with normal XENIX installation procedure.

Setting Up a Two-Drive System

1. Cabling Requirements: One (1) 34-pin daisy chain straight-through cable. One (1) 20-pin straight-through cable.

2. On hard disks: set *Drive Select* to DS0 (or DS1) on the first drive, and *Drive Select* to DS1 (or DS2) on the second drive. Install termination resistor on drive at end of daisy chain cable. Remove termination resistor on drive in the middle of daisy chain cable.

3. Follow steps 3-9 above.

User- Configurable Jumpers

W20	W21	W22	W23	Refer to Drive Tables

W17	W18	W19	Winchester I/0 Port Base Address
0*	0*	0*	320H
0	0	1	324H
0	1	0	328H
0	1	1	32CH
1	0	0	1A0H
1	0	1	1A4H
1	1	0	1A8H
1	1	1	1ACH

W16	BIOS EPROM Control
0*	Enable BIOS
1	Disable BIOS

W15	BIOS Base Address
0*	C8000H
1	CA000H

W14	Floppy Disk I/O Port Base Address
0*	03F0H
1	0370H

W10	W11	Bytes/Sector	Sectors/Track (ST506) 8620	8627
0*	0*	512	17	26
0	1	512	18	
1	0	1024	9	
1	1	1056	9	

0=Jumper not installed
1=Jumper installed
*=Jumper as shipped

LUN 0		LUN 1	
W9	**Sector Type**	**W6**	**Sector Type**
0*	Soft sectored	0*	Soft sectored
1	Hard sectored	1	Hard sectored
W8	**Drive Class**	**W5**	**Drive Class**
0*	ST506 Compatible	0*	ST506 Compatible
1	ESDI Compatible	1	ESDI Compatible
W7	**Drive Type**	**W4**	**Drive Type**
0*	Fixed	0*	Fixed
1	Removable	1	Removable

0=Jumper not installed
1=Jumper installed
*=Jumper as shipped
LUN=Logical Unit Number

Drive Jumper Settings

BIOS #1002579, #1002580:

W20-W23: Hard disk drive jumpers

BIOS #1002661, #1002662:

W20	Reserved	
W21	I/O Speed Option:	
	0*	High Performance (not supported by all computers)
	1	Normal
W22, W23	Hard disk drive jumpers	

1.8.8 Drive Tables

Table A
8620 Drive Table
BIOS #1002579, AT3

W20	W21	Drive/Model or Compatibles	Cyl.	Heads	Write Precomp.
LUN 0					
1	1	ESDI Drives			
0	1	Vertex/Priam V170	987	7	–
1	0	Maxtor XT1140	918	15	–
0	0	Miniscribe 3425	612	4	–
W22	**W23**				
LUN 1					
1	1	ESDI Drives			
0	1	Vertex/Priam V170	987	7	–
1	0	Maxtor XT1140	918	15	–
0	0	Miniscribe 3425	612	4	–

Table B
8627 Drive Table
BIOS #1002580, BIOS AT4

W20	W21	Drive/Model or Compatibles	Cyl.	Heads	Write Precomp.
LUN 0					
1	1	ESDI Drives			
0	1	Seagate ST277R	820	6	–
1	0	Seagate ST4144R	1024	9	–
0	0	Seagate ST238	615	4	–
W22	**W23**				
LUN 1					
1	1	ESDI Drives			
0	1	Seagate ST277R	820	6	–
1	0	Seagate ST4144R	1024	9	–
0	0	Seagate ST238	615	4	–

Table C
8620 Drive Table
BIOS #1002661, AT5

W23	Drive Type	Cyl.	Heads	Write Precomp.
LUN 0				
0	Seagate ST225	612	4	–
1	ESDI Drive			
W22				
LUN 1				
0	Seagate ST225	612	4	–
1	ESDI Drive			

Table D
8620 Drive Table
BIOS #1002662, BIOS AT6

W23	Drive Type	Cyl.	Heads	Write Precomp.
LUN 0				
0	Seagate ST238R	615	4	–
1	ESDI Drive			
W22				
LUN 1				
0	Seagate ST238R	615	4	–
1	ESDI Drive			

Drive Characteristics Reference Guide

Drive Characteristics
Reference Guide
MFM Hard Disk Drives

Drive/Model	Cyl.	Heads	Write Precomp.
Maxtor XT1140	918	15	-
CMI 6426	640	4	256
Rodine 352	306	4	-
Microscience HH312	306	4	-
Seagate ST4026	615	4	300
Cynthia 570	987	7	-
Vertex V170	987	7	-
DMA/RICOH (Removable 10MB)	612	2	400
Seagate ST4038	733	5	300
Seagate ST213	612	2	256
Miniscribe 3425	612	4	128
Quantum Q540	512	8	256
Seagate ST4051	977	5	300
CDC 3212	612	2	128
Miniscribe 3012/3212	612	2	128
CMI 6640	640	6	256
Tulin 240	640	6	256
Seagate ST225	615	4	256
CMI 3426	615	4	256
CMI 5412	306	4	128
Lapine 3522	306	4	128
Otari 514	306	4	128
Seagate ST412/ST212	306	4	128
Shugart 712	306	4	128

Note: This is a list of some prominent MFM drives in the industry. It is not a comprehensive list.

Drive Characteristics Reference Guide RLL Vendor Certified Hard Disk Drives

Drive/Model	Cyl.	Heads	Write Precomp.
Atasi 3085	1024	8	–
Miniscribe 8438	612	4	–
Microscience HH330	612	4	–
Microscience HH738	612	4	–
Peripheral Tech 357R	615	6	–
Lapine LT300	616	4	–
Priam V170	987	7	–
Priam V185	1024	7	–
Priam 514	1024	11	–
Priam 519	1024	15	–
Seagate ST251R	820	4	–
Seagate ST277R	820	6	–
Seagate ST4077R	1024	5	–
Seagate ST4144R	1024	9	–
Toshiba MK53FB	830	5	–
Toshiba MK54FB	830	7	–
Seagate ST238R	615	4	–
Miniscribe 3438	615	4	–

Note: This is a list of some prominent RLL drives in the industry. It is not a comprehensive list.

1.9 XENIX 6300+ Operating System

XENIX 6300+ Operating System runs standard AT&T hardware with original switch settings.

1.9.1 Memory Cards

Use AT&T memory cards, since the 6300+ has its own 16 bit bus and is not AT memory card compatible.

1.9.2 Hard Disks

The latest BIOS, version 2.01, contains a table of 8 supported disks and their characteristics:

Type	Size	Example	Cyls	Hds
0	10MB		306 cylinders	4 heads
1	30MB	CDC Wren	697 cylinders	5 heads
2	20MB	CMI CM6426	640 cylinders	4 heads
3	40MB	Tandon	981 cylinders	5 heads
4	40MB	Seagate ST4051	977 cylinders	5 heads
5	80MB	Miniscribe 8086	1024 cylinders	8 heads
6	67MB	Micropolis 1325	1024 cylinders	8 heads
7	20MB	Seagate ST225 or Miniscribe 3425	612 cylinders	4 heads

---

Note

BIOS table 2.01 contains an error, the entry for type 5 should say 67MB instead of 80MB.

Type 7 is the standard 20 MB disk currently in use in the PC 6300+. The above table is the same as the one contained in BIOS versions 1.04 and 1.06. There are some other parameters in the table that are not listed here. These parameters affect performance rather than capacity, so they are not listed.

Hard disk drives C (and, optionally, D) are identified as entries 0-7 in the table above by setting one DIP switch on the motherboard and two jumpers on the Western Digital controller board. On the motherboard, there is a package of eight switches labeled "DSW-1." Switch 3 is the most significant bit for drive D, and switch 4 is the most significant bit for drive C. On the HDU controller board, there is a package of eight jumper positions labeled "SW1."

The next two significant bits are jumper positions 4 and 3 for drive C and 2 and 1 for drive D.

In all cases, "OFF" or "Open" is binary "1," and "ON" or "Closed" is binary "0." So the eight possible settings are:

	Motherboard DSW-1 switch	Disk Controller Board SW1 jumpers	
Setting	3(D)_or_4(C)	4(C)_or_2(D)	3(C)_or_1(D)
0	On	Closed	Closed
1	On	Closed	Open
2	On	Open	Closed
3	On	Open	Open
4	Off	Closed	Closed
5	Off	Closed	Open
6	Off	Open	Closed
7	Off	Open	Open

1.10 XENIX 86XT Operating System

1.10.1 Machines

We have used the following machines under the XENIX 86XT Operating System:

American Research Corporation (ARC-XT)
AT&T 6300 (Olivetti M24 in Europe)
Compaq Portable, DeskPro™ * and Plus™ *
Eagle Turbo
Epson Equity I+
IBM PC XT®
ITT XP
ITT XTRA
Mitsubishi
NCR™ PC-6**
Tandy 1200
UNISYS®
Wavemate Bullet-286 XT
WYSEpc WY-1100
WYSEpc+ WY-1400

* Internal cartridge tape not supported.

** For the NCR machine, configure the switches on the motherboard for a color monitor, regardless of the type of monitor used.

These machines have been reported to run the XENIX 86XT Operating System, but we have not tested them:

Compuserve/Canada Arthur in Canada
Ericsson PC
Leading Edge
Victor VPC 30 and VPC 15

1.10.2 Math Chips

We have used the following math co-processor chips under the XENIX 86XT Operating System:

8087	For 8088 CPUs.
8087-2	For 8086 CPUs.
8087-3	For fast 8086 CPUs.

1.10.3 Memory Cards

We have used the following memory cards under the XENIX 86XT Operating System:

AST 6 Pack+
Tandon
Quadram Quadboard
Sigma Maximizer
Microsoft RAM card
Tecmar Captain or First-Mate
AT&T for the 6300

1.10.4 Accelerator Boards

Accelerator boards are special cards that plug in to an 8088 machine (not an 8086). They replace the 8088 CPU with a 286 CPU and perhaps replace some memory as well. XENIX runs on accelerator boards in 8086 unmapped mode.

The following accelerator board is supported in the XENIX 86XT Operating System standard distribution:

Seattle Telecom and Data (STD) PC286

1.10.5 Add-On Hard Disks

We have used the following hard disks under the *standard disk* configuration.

CMI 15
Plus HardCard 10
Miniscribe 20
Seagate 20
CDC 30
Rodime 30
Mountain 20, 30
Syquest SQ306 standard
CDC Wren
Seagate ST-225
Miniscribe 3425

The following disks have been reported to work under the standard disk configuration, but we have not tested them:

Peachtree 10, 20, 30
Maynard Apollo 30
Franklin Telecom 10, 20, 33
Rodime 10
Tandon 10
Alpha Omega 10, 20, 30
Mountain Hard File

The following tables list the controllers most commonly used with 8086/8088 machines, and the type of hard disks they can support:

XEBEC Controller
(standard IBM PC/XT compatible)

Type	Cyls	Hds	Size	Example
0	306	2	5 MB	
1	375	8	24 MB	
2	306	6	15 MB	
3	306	4	10 MB	standard

DTC Controller
(standard on AT+T 6300 - fully supported)

Type	Cyls	Hds	Size	Example
0	306	2	5 MB	
1	375	8	24 MB	
2	306	6	15 MB	
3	306	4	10 MB	
4	306	2	5 MB	Syquest SQ306 standard
5	697	5	30 MB	CDC Wren
6			20 MB	Seagate ST-225
7			20 MB	Miniscribe 3425
Drive types 8 - 15 reserved				

1.10.6 Compatible Hard Disk Controllers

The following table lists the hard disk controllers that we have used under the XENIX 86XT Operating System configuration:

XEBEC 1210A
DTC 5150
WD 1002/WX-2
Adaptec 2002
I^2 interface
ADES
CDC

Appendix B

Common Installation/Configuration Questions and Answers

B.1 Common Questions and Answers

This section is our support department's combined effort to present answers to the most common installation/configuration questions.

1. "EEE"s on Screen When Booting from N1 Floppy

2. Blank or Garbled Display When Booting, Or Display Won't Scroll

3. Error Message: "**Panic: memory failure -- parity error**"

4. Adding a Serial Card and Terminal

5. Serial Printer is Not Printing

6. Setting Terminal Type Automatically

7. 286 Drivers Won't Work with 386

8. Trouble Configuring Memory on 386 Machine, or System Runs Slow After Adding Memory

9. Parallel Printer is Printing Slowly or Not at All

1. **PROBLEM:** When booting from the N1 floppy, "EEEEEE"s print on the console and the system refuses to respond.

   REMEDY: Try booting a DOS floppy, or other bootable diskette. If the machine can boot a diskette, then your N1 floppy is corrupted. You are entitled to a free replacement floppy.

2. **PROBLEM:** When booting up, the console screen blanks, the cursor is gone, or the display is garbled.

   REMEDY: You may have an incompatible monitor card. Check "Video Adapters and Monitors" under "General Compatibility Guidelines" in Appendix A of these Notes for a list of supported video cards and monitors. If yours is not listed but compatible with one on the list, it should work. If not, check your card's hardware manual to see if there are ways to configure the switch settings so the card is in an IBM-compatible emulation mode and that it is addressing the kind of monitor attached. If changing the switch settings fails, then your monitor card is incompatible, and should be replaced with a compatible card.

3. **PROBLEM:** I see the error message:

   Panic: memory failure -- parity error

   REMEDY: Some part of your hardware is sending a "non-maskable memory error interrupt" (a signal sent by the hardware that halts the operating system). Run your system's hardware diagnostics tests if available. Physically re-seat your memory cards and chips, and check for bent pins, etc. If these measures fail to correct the problem, or you don't feel comfortable in checking your hardware yourself, seek assistance from professional hardware experts. One frequent cause is memory chips that are slower than factory-recommended chips.

4. **PROBLEM:** I am trying to add a terminal to a serial port, but do not get a login.

   REMEDY: Follow these procedures to locate the cause of the problem.

   A. If you are using a 4 or 8-port serial expansion board, check to see if your board is recognized at bootup by checking the bootup message. If not, the some of switches on your card are not set properly. The most frequent cause of boards not being recognized at boot time is incorrect switch settings for the strap address. Check your board's hardware documentation for the proper switch settings and the "Compatible Hardware" section of these *Notes* for the correct addresses. COM1 should be configured as interrupt 4 and COM2 as interrupt 3. When the card is correctly configured and recognized in the boot up message, run "/etc/mkdev serial" which creates device files for your extra serial ports.

   B. Note that the bootup message only verifies the addresses of the ports. If your interrupt is incorrectly configured on the board, it still will not work. Both the addresses and the interrupt line must be set correctly on the board. You must also make sure that the addresses and interrupt line chosen do not conflict with any other devices (e.g. other serial boards, tape drives, etc.).

   C. Make sure you are in multi-user mode, your terminal is plugged in and turned on and set for 9600 baud, 8 data bits, 1 stop bit, no parity. (If your terminal doesn't send these, then see the section titled "Changing the gettydefs File" in Chapter 7 of the *Operations Guide* for ways to configure your terminal.) The terminal should be connected with a cable so that *Transmit Data* on the serial port is connected to *Read Data* on the terminal, and

Transmit Data on the terminal is connected to *Read Data* on the serial port (see the **serial**(HW) manual page in the XENIX *Reference Manual* or your terminal manual for more information.)

D. 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 login. If not, disable the port by typing:

disable /dev/*ttyname* <RETURN>

where *ttyname* is the device special filename of the port in question. (For more information on the naming convention for serial ports, see **serial**(HW) in the *Run Time Environment*.) Make sure you're using the non-modem control device.

E. Check the */etc/ttys* file so the entry for this serial port looks like this:

0mttyname

If it does not look like the example above, then edit */etc/ttys* with a text editor to correct it.

F. From the console, as root, see if you can redirect output to your terminal by typing:

date > /dev/ttyname <RETURN>

If you do not see the date printed on your 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, a hardware problem is possible, so verify the following:

- Your cable is configured correctly (see B. above). Also, try using only pins 2, 3 and 7.

- Re-check your terminal setup configuration (see #2 above).

- Re-check your switches on your serial port. If you are using a multi-port card, try other lines on that card.

- Try attaching the terminal to a standard serial port, COM1 or COM2, to see if the terminal and cabling is correct.

G. Once you get the date printed on your terminal, then enable the port and look for the 'login:' prompt.

enable /dev/*ttyname* <RETURN>

If you do not see the 'login' prompt, then verify that **getty** is actually running on that port and that the software is configured properly by typing:

ps -lft *ttyname* <RETURN>

The output should look similar to the following where "login" or "getty" is listed in the "CMD" column.

```
F S UID PID  PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
1 S 214 9210 1    0 30  20 344  22 6994  2a  0:06 getty
```

H. If you have typed the **disable** and **enable** commands many times, it is possible that a new getty cannot be spawned on that port. If so, then shutdown the system, re-boot, enter multi-user mode, and try again.

5. **PROBLEM:** I can't get a printout on my serial printer.

REMEDY: Make sure your serial printer is configured properly under XENIX:

A. Be sure your printer is capable of supporting XON/XOFF or DTR protocols and is configured for those protocols. Verify that it is turned on and the cable is properly attached to the computer and printer. For more information on XON or XOFF protocols, consult your printer hardware documentation.

B. If the port is enabled, disable it by typing:

disable /dev/*ttyname* <RETURN>

where *ttyname* is the device special filename of the port in question. (For more information on the naming convention for serial ports, see **serial** (HW) in the *Run Time Environment*.) Make sure you're using the non-modem control device.

C. From the console, as root, see if you can redirect output to your printer by typing:

date > /dev/*ttyname* <RETURN>

If you do not see the date printed on your printer 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, hardware is suspect so verify the following:

- Your cable is configured correctly. If using XON/XOFF protocol, see Question 5 step B above. Also, try using only pins 2, 3 and 7.

- Re-check your printer configuration by verifying its switches in your printer hardware manual. (See item A above.)

- Re-check your switches on your serial port. If you are using a multi-port 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 is correct.

D. When you get the date printed on your printer, you should run **mkdev lp** to configure the spooler to support this printer. (See Chapter 7 of the XENIX *Operations Guide* for more information on **lpinit**, which is called by **mkdev lp**.) Then be sure to enter the correct **stty**(C) options in the spooler's interface script (*/usr/spool/lp/interface/printername*) to control the baud rate, XON/XOFF protocol, and other options as necessary. The baud rate and XON/XOFF protocol should match the baud rate set on your printer.

6. **PROBLEM**: How can I get my terminal type set automatically when I login?

   REMEDY: XENIX can be easily configured to automatically set the terminal type upon login by following these procedures:

   A. Login on the terminal in question and determine which *ttyname* you're using by typing:

      tty <RETURN>

   B. Login 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

C. Then the user's start up file must be edited with the appropriate **tset**(C) command line to automatically set the terminal type. In each C-shell user's *.login* file, add the following line:

```
tset -s -Q > /tmp/tset$$; source /tmp/tset$$; /bin/rm /tmp/tset$$
```

Be sure the remove the default **setenv**(C) command line involving TERM and TERMCAP from the *.login* file.

In each Bourne Shell user's *.profile*, add the following line:

```
eval 'tset -s'
```

Be sure to remove the existing **tset** command line from the *.profile* file.

D. Have each user log out, then log in again to test the new terminal type change. After they log in, have them verify the new termtype by typing:

env <RETURN>

7. **PROBLEM:** My third-party peripheral device (tape, serial card, etc.) that used to work with XENIX 286 OS using their driver, no longer works with XENIX 386 OS. In some cases, my system hangs when trying to reboot after installing their driver.

   REMEDY: New vendor-supplied device drivers are required for XENIX 386. 286 device drivers will not link into the 386 kernel. Contact the vendor who supplied you with the 286 device driver, and ask them for a new 386 device driver.

8. **PROBLEM:** I'm having having trouble configuring more memory on my 386 machine, or since adding more memory, my 386 machine runs more slowly.

   REMEDY: With a Compaq 386 machine you must fully populate the memory board that comes with the machine with column static RAM chips, before you can add extended memory. This is generally true with other 386 machines also.

On a Compaq 386, extended memory must be configured to start at the 4 MB address.

Most extended memory cards use 16-bit memory chips with a speed of about 150 ns. This memory is about 8 times as slow as static RAM, and so performance may be noticeably slower. We recommend using only 32-bit memory from your computer's manufacturer. If you must use 16-bit memory, at least use memory with fast (90ns) chips.

9. **PROBLEM**: I'm trying to print on a parallel printer but am getting no response or very slow printing.

   REMEDY: Verify the following items, as it is important that your parallel ports be configured properly to work under XENIX:

   A. The printer must be IBM compatible and should use a standard Centronics interface cable in order for it to work at all.

   B. The IBM AT and XT and compatibles, and 6300+ only support up to 2 parallel ports. Deconfigure additional ports.

   C. The parallel port on the monochrome card should be configured for interrupt vector 7, and is recognized as **lp1** when booting up.

   D. The main parallel port should be configured for interrupt vector 7 and is recognized as **lp0**. So you must use either the main or the monochrome's port – not both – to avoid a hardware conflict which would cause slow printing.

E. The alternate/second parallel port should be configured for interrupt vector 5, and recognized as **lp2**. Make sure no other hardware is using these interrupts. (See your hardware manual for information on configuring your parallel ports.)

F. Check to make sure your parallel card is recognized by XENIX by rebooting your system. Following the copyright information on the screen, you should see:

   Parallel port lp*n* present

   where lp*n* is the device special filename of the port the printer is attached to (**lp0**, **lp1** or lp2) as described above. If you do not see this message, check the switches and jumpers on your parallel card to make sure they are correct. Also try setting the card for a different configuration, if possible.

G. Be sure your printer is turned on and on-line, and the cable is properly attached to the computer and printer.

H. From the console, logged in as root, see if you can redirect output to your printer by typing:

   date > /dev/lp*n* <RETURN>

   If you do not see the date printed on your printer, there is most likely some type of hardware malfunction, so verify the following:

   - Your cable is securely connected and all wires are good. Using the cable on a known good system, or printing under DOS are good ways to test this.

   - Re-check your printer configuration by verifying its switches in your printer hardware manual.

- Re-check the switches on your parallel card. It must be recognized at bootup, as described above in Step F.

When you get the date printed on your printer, you should run **mkdev lp** to configure the spooler to support this printer. (See "Using Peripheral Devices" of the XENIX *Operations Guide* for more information on **lpinit**, the script called by **mkdev lp**.)

I. If your parallel ports are configured properly, as described above and you still get slow printing, your parallel port may not be capable of generating interrupts. See "Slow Parallel Printers" under Software Notes in these *Release Notes* for information on correcting this problem.

Name

diskcp, diskcmp – Copies or compares floppy disks.

Syntax

diskcp [-f] [-d] [-s] [-48ds9] [-96ds9] [-96ds15] [-135ds9] [-135ds18]
diskcmp [-d] [-s] [-48ds9] [-96ds9] [-96ds15] [-135ds9] [-135ds18]

Description

These commands provide easy copying of a source floppy disk. *dd*(C) is used to make an image of the source floppy (the one you wish to copy). On machines with one floppy drive *diskcp* temporarily transfers the image to the hard disk until a blank "target" floppy is inserted into the floppy drive. On machines with two floppy drives *dd* immediately places the image of the source floppy directly on the target floppy.

The options are:

-f Format the target floppy disk before the image is copied.

-d The computer has dual floppy drives. *diskcp* copies the image directly onto the target floppy.

-s Uses *sum*(C) to compare the contents of the source and target floppies; gives an error message if the two do not match.

-48ds9
This setting is for low density 48tpi floppies. It is the default setting.

-96ds9
This setting is for high density 96tpi floppies.

-96ds15
This setting is for quad density 96tpi floppies.

-135ds9
This setting is for high density 135tpi 3.5 inch floppies.

-135ds18
This setting is for quad density 135tpi 3.5 inch floppies.

diskcmp functions similarly to *diskcp*. It compares the contents of one floppy disk with the contents of a second floppy disk using the *cmp* utility.

Examples

To make a copy of a floppy, place the source floppy in the drive and type:

```
diskcp
```

When *diskcp* is finished copying to the hard disk, it prompts you to insert the target floppy in the drive. If you specify the **-f** flag when you invoke *diskcp* , the program formats the target floppy. When the copy is finished, *diskcp* prompts if you would like to make another copy of the same source disk. If you enter 'n', it prompts if you would like to copy another source disk.

Specify the **-d** flag on the command line if you have two floppy drives:

```
diskcp -d
```

Notes

If *diskcp* encounters a write error while copying the source image to the target disk, it formats the disk and tries to write the source image again. This happens most often when an unformatted floppy is used and the **-f** flag is not specified.

Files

/usr/bin/diskcp
/usr/bin/diskcmp
/tmp/disk$$

See Also

cmp(C), dd(C), sum(C)

Name

dos, doscat, doscp, dosdir, dosformat, dosls, dosrm, dosrmdir – Access DOS files.

Syntax

doscat [-r | -m] file ...

doscp [-r | -m] file1 file2

doscp [-r | -m] file ... directory

dosdir directory ...

dosformat [-fqv] drive

dosls directory ...

dosmkdir directory ...

dosrm file ...

dosrmdir directory ...

Description

The *dos* commands provide access to the files and directories on MS-DOS disks and on a DOS partition of the hard disk. The commands perform the following actions:

doscat Copies one or more DOS files to the standard output. If **-r** is given, the files are copied without newline conversions. If **-m** is given, the files are copied with newline conversions (see "Conversions" below).

doscp Copies files between a DOS disk and a XENIX filesystem. If *file1* and *file2* are given, *file1* is copied to *file2*. If a *directory* is given, one or more *file*s are copied to that directory. If **-r** is given, the files are copied without newline conversions. If **-m** is given, the files are copied with newline conversions (see "Conversions" below).

dosdir Lists DOS files in the standard DOS style directory format.

dosformat Creates a DOS 2.0 formatted diskette. The drive may be specified in either DOS drive convention, using the default file **/etc/default/msdos**, or using the XENIX special file name. The **−f** option suppresses the interactive feature. The **−q** (quiet) option is used to suppress information normally displayed during *dosformat* . The **−q** option does not suppress the interactive feature. The **−v** option prompts the user for a volume label after the diskette has been formatted. The maximum size of the volume label is 11 characters.

dosls Lists DOS directories and files in a XENIX style (see *ls*(C)).

dosrm Removes files from a DOS disk.

dosmkdir Creates a directory on a DOS disk.

dosrmdir Deletes directories from a DOS disk.

The *file* and *directory* arguments for DOS files and directories have the form:

device:name

where *device* is a XENIX pathname for the special device file containing the DOS disk, and *name* is a pathname to a file or directory on the DOS disk. The two components are separated by a colon (:). For example, the argument:

/dev/fd0:/src/file.asm

specifies the DOS file, **file.asm**, in the directory, **/src**, on the disk in the device file **/dev/fd0**. Note that slashes (and not backslashes) are used as filename separators for DOS pathnames. Arguments without a *device:* are assumed to be XENIX files.

For convenience, the user configurable default file, **/etc/default/msdos**, can define DOS drive names to be used in place of the special device file pathnames. It may contain the following lines:

A=/dev/fd0
C=/dev/hd0d
D=/dev/hd1d

The drive letter "A" may be used in place of special device file pathname **/dev/fd0** when referencing DOS files (see "Examples" below). The drive letter "C" or "D" refer to the DOS partition on the first or second hard disk.

The commands operate on the following kinds of disks:

DOS partitions on a hard disk
5 1/4 inch DOS
3 1/2 inch DOS
8, 9, 15, or 18 sectors per track
40 tracks per side
1 or 2 sides
DOS versions 1.0, 2.0 or 3.0

Conversions

All DOS text files use a carriage-return/linefeed combination, CR-LF, to indicate a newline. XENIX uses a single newline LF character. When the *doscat* and *doscp* commands transfer DOS text files to XENIX, they automatically strip the CR. When text files are transferred to DOS, the commands insert a CR before each LF character.

Under some circumstances the automatic newline conversions do not occur. The **−m** option may be used to insure the newline conversion. The **−r** option can be used to override the automatic conversion and force the command to perform a true byte copy regardless of file type.

Examples

```
doscat /dev/fd0:/docs/memo.txt
doscat /tmp/f1 /tmp/f2 /dev/fd0:/src/file.asm

dosdir /dev/fd0:/src
dosdir A:/src A:/dev

doscp /tmp/myfile.txt /dev/fd0:/docs/memo.txt
doscp /tmp/f1 /tmp/f2 /dev/fd0:/mydir

dosformat A:
dosformat /dev/fd0

dosls /dev/fd0:/src
dosls B:

dosmkdir /dev/fd0:/usr/docs

dosrm /dev/fd0:/docs/memo.txt
dosrm A:/docs/memo1.txt

dosrmdir /dev/fd0:/usr/docs
```

Files

/etc/default/msdos	Default information
/dev/fd*	Floppy disk devices
/dev/hd*	Hard disk devices

See Also

assign(C), dtype(C)

Notes

It is not possible to refer to DOS directories with wild card specifications. The programs mentioned above cooperate among themselves so no two programs will access the same DOS disk. Only one process will access a given DOS disk at any time, while other processes wait. If a process has to wait too long, it displays the error message, "can't seize a device," and exits with an exit code of 1.

The following hard disk devices:

/dev/hd0d
/dev/rhd0d
/dev/hd1d
/dev/rhd1d

are similar to **/dev/hd0a** in that the disk driver determines which partition is the DOS partition and uses that as *hd?d*. This means that software using the DOS partition does not need to know which partition is DOS (the disk driver determines that).

The XENIX Development System supports the creation of DOS executable files, using *cc* (CP). Refer to the XENIX *C User's Guide* and *C Library Guide* for more information on using XENIX to create programs suitable for DOS systems.

Name

dtype – Determines disk type.

Syntax

dtype [**−s**] device ...

Description

dtype determines type of disk, prints pertinent information on the standard output unless the silent (**−s**) option is selected, and exits with a corresponding code (see below). When more than one argument is given, the exit code corresponds to the last argument.

Disk Type	Exit Code	Message (optional)
Misc.	60	error (specified)
	61	empty or unrecognized data
Storage	70	dump format, volume n
	71	tar format[, extent e of n]
	72	cpio format
	73	cpio character (−c) format
MS-DOS	80	DOS 1.x, 8 sec/track, single sided
	81	DOS 1.x, 8 sec/track, dual sided
	90	DOS 2.x, 8 sec/track, single sided
	91	DOS 2.x, 8 sec/track, dual sided
	92	DOS 2.x, 9 sec/track, single sided
	93	DOS 2.x, 9 sec/track, dual sided
	94	DOS 2.x, fixed disk
	110	DOS 3.x, 9 sec/track, dual sided
XENIX	120	XENIX 2.x filesystem [needs fsck]
	130	XENIX 3.x or later filesystem [needs fsck]

Notes

word-swapped refers to byte ordering of long words in relation to the host system.

XENIX file systems and dump and cpio binary formats may not be recognized if created on a foreign system. This is due to such system differences as byte and word swapping and structure alignment.

This utility only works reliably for floppy diskettes.

Name

fd - floppy devices

Description

The **fd** devices implement the XENIX interface with floppy disk drives. Typically, the *tar*(C), *cpio*(C) or *dd*(C) commands are used to read or write floppy disks. For instance,

```
tar tvf /dev/fd0
```

tabulates the contents of the floppy disk in drive 0 (zero).

The block special **fd** devices are also block-buffered. The floppy driver can read or write 512 bytes at a time using raw i/o. Note that block transfers are always a multiple of the 1K disk block size.

The floppy devices are named **/dev/fd0** and **/dev/fd1** (see Notes, below, for more information about device naming procedure).

The corresponding character special (raw) devices, **/dev/rfd0** and **/dev/rfd1,** afford direct, unbuffered transmission between the floppy and the user's read or write transfer address in the user's program.

For information about formatting, see *format*(C).

The minor device number determines what kind of physical device is attached to each device file (see Notes).

Files

/dev/fd0	/dev/rfd048ds8	/dev/rfd096ds15	/dev/rfd0135ds9
/dev/fd1	/dev/rfd148ds8	/dev/rfd196ds15	/dev/rfd1135ds9
/dev/rfd0	/dev/rfd048ds9	/dev/rfd096ds8	/dev/rfd0135ds18
/dev/rfd1	/dev/rfd148ds9	/dev/rfd196ds8	/dev/rfd1135ds18
		/dev/rfd048ss8	
		/dev/rfd148ss9	

Notes

When accessing the character special floppy devices, the user's buffer must begin on a word boundary. The count in a *read*(S), *write*(S), or *lseek*(S) call to a character special floppy device must be a multiple of 512 bytes.

Device names determine the particular drive and media configuration. The device names have the form: fd048ds9 Where: fd0 = drive number (0, 1, 2 or 3) 48 = number of disk tracks per inch (48 or 96) ds = single or double sided floppy (ss or ds) 9 = number of sectors on the floppy (8 or 9)

For instance, /dev/fd048ss9 indicates a 48 track per inch, single sided, 9 sector floppy disk device in drive 0.

The minor device numbers for floppy drives depend on the drive and media configuration. The most common are:

	48tpi				96tpi		135tpi	
	ds/8	ds/9	ss/8	ss/9	ds/15	ds/8	ds/9	ds/18
Drive	Minor Device Number							
0	12	4	8	0	52	44	36	60
1	13	5	9	1	53	45	37	61
2	14	6	10	2	54	46	38	62
3*								

* reserved for special, non-floppy devices connected to the floppy controller as unit #3.

The scheme for creating minor device numbers is as follows. When interpreted as a binary number, each bit of the minor device number represents some aspect of the device/media configuration.

For example, the minor device number for **/dev/fd048ss8** is "8." Interpreted as a binary number, 8 is:

00001000

This is how each bit, or binary digit, is significant:

48tpi - 0 96tpi - 1 135tpi - 1	Sectors per Track		ss - 0 ds - 1	Drive	
32	16	8	4	2	1
0	0	1	0	0	0

Only the last six digits of the number are used in minor device identification. The first significant digit is the third from the left. In this example, the third digit from the left is zero, thus the device is 48tpi. The next two digits mean:

Bits		Sectors per Track
16	8	
0	0	9
0	1	8
1	0	15
1	1	18

The fourth digit tells whether the floppy is single sided (ss - 0) or double sided (ds - 1). The last two signify the drive number:

Bits		Drive Number
2	1	
0	0	0
0	1	1
1	0	2
1	1	3*

* reserved for special, non-floppy devices connected to the floppy controller as unit #3.

Using this information, you can construct any minor device numbers you need.

It is not advisable to format a low density (48tpi) diskette on a high density (96tpi or 135tpi) floppy drive. Low density diskettes written on a high density drive should be read on high density drives. They may or may not be readable on a low density drive.

Use error-free floppy disks for best results on reading and writing.

Name

mkdev - Calls scripts to add peripheral devices.

Syntax

/etc/mkdev lp
/etc/mkdev hd
/etc/mkdev serial
/etc/mkdev fs [*device file*]
/etc/mkdev fd
/etc/mkdev tape
/etc/mkdev om
/etc/mkdev shl

Description

mkdev calls the scripts to create the requested type of device file(s). *mkdev* may call *lpinit*(C), *hdinit*, *serinit*, *fdinit*, *fsinit*, *tapeinit*, *ominit*, or *shlinit*. If no arguments are listed, *mkdev* prints a usage message.

/etc/mkdev lp creates device files for use with line printers. (See *lpinit*(C).)

/etc/mkdev hd creates device files for use with a peripheral hard disk. The device files for an internal hard disk already exist. *hdinit* invokes the following utilities: *dparam*(C), *badtrk*(M), *fdisk*(C), and *divvy*(C).

/etc/mkdev serial creates device files for use with serial cards. The device files for the first and second ports already exist. Additional device files must be created for the ports added when expansion cards are added to the system. The **/etc/ttys** and **/etc/ttytype** files are updated.

/etc/mkdev fs performs the system maintenance tasks required to add a new filesystem to the system once the device is created (*mknod*(C)) and the filesystem is made (*mkfs*(C)). It creates the */file* and */file/***lost+found** directories, reserves slots in the **lost+found** directory, and modifies **/etc/checklist**, **/etc/default/filesys** and **/etc/default** to check (*fsck*(C)) and mount (*mount*(C),*mnt*(C),*rc*(C)) the filesystem as appropriate. It is usually used in conjunction with *mkdev hd* when adding a second hard disk to the system or with *mkdev fd* when creating a mountable filesystem on a floppy, but can be used on any additional filesystem (for example, on a large internal hard disk).

/etc/mkdev fd creates bootable and root file system floppy disks. The three basic options are: boot and root on a single disk (96 or 135 tpi only), boot and root pair (48 tpi) or filesystem only. Use with *mkdev fs* when creating a filesystem-only floppy.

/etc/mkdev tape configures the tape driver in preparation for linking a new kernel that includes tape support. It adds a standard quarter-inch cartridge tape driver and/or a mini-cartridge tape driver.

The current driver configurations can be displayed, and changed if necessary. A zero in any of the fields means the driver automatically detects the type of tape device installed and uses the built-in values for that device. If the autoconfiguration values are not correct for your drive, refer to your hardware manual for the correct values, configure the driver and relink the new kernel. *mkdev tape* can also be used to remove a tape driver from the existing kernel.

/etc/mkdev shl initializes necessary devices and configures kernel parameters associated with the number of shell layers sessions available on the system.

Once the driver is configured, you are prompted for re-linking the kernel. The appropriate devices in **/dev** are created.

The various *init* scripts prompt for the information necessary to create the devices.

Files

/usr/lib/mkdev fs
/usr/lib/mkdev/hd
/usr/lib/mkdev/lpinit
/usr/lib/mkdev/serial
/usr/lib/mkdev/tape
/usr/lib/mkdev/fd
/usr/lib/mkdev/shl

See Also

badtrk(M), divvy(C), dparam(C), fd(HW), fdisk(C), filesys(F), format(C), hd(HW), lp(HW), lpinit(C), mkfs(C), mknod(C), mount(C), serial(HW), tape(HW), and the "Using Peripheral Devices" chapter in the XENIX *Operations Guide*.

Notes

ominit is not supplied with the standard XENIX distribution.

Name

vmstat – Report paging and system statistics.

Syntax

vmstat [**−fs**] [**−n** namelist] [**−l** lines] [interval [count]]

Description

vmstat reports some statistics kept by the system on processes, demand paging, and cpu and trap activity. Three types of reports are available:

(default)
A summary of the number of processes in various states, paging activity, system activity, and cpu cycle consumption.

−f Number of *fork*(S)'s done.

−s
A verbose listing of paging and trap activity.

If no *interval* or *count* is specified, the totals since system bootup are displayed.

If an *interval* is given, the number of events that have occurred in the last *interval* seconds is shown. If no *count* is specified, this display is repeated forever every *interval* seconds. Otherwise, when a *count* is also specified, the information is displayed *count* times.

Other flags that may be specified include:

−c *corefile*
Uses the file *corefile* in place of **/dev/kmem.**

−n *namelist*
Use file *namelist* as an alternate symbol table instead of **/xenix.**

−l *lines*
For the default display, repeat the header every *lines* reports (default is **20**).

The fields in the default report are:

procs
The number of processes which are:

r In the run queue.

b Blocked waiting for resources.

w Swapped out.

These values always reflect the current situation, even if the totals since boot are being displayed.

paging
Reports on the performance of the demand paging system. Unless the totals since boot are being displayed, this information is averaged over the proceeding *interval* seconds:

si Number of processes swapped in.

so Number of processes swapped out.

ch Page cache hits.

cm
Page cache misses.

ffr Filesystem page reads.

swr
Swap area page reads.

sww
Swap area page writes.

rec
Number of pages reclaimed from the free list.

shf
Number of pages shared as copy-on-write after *fork*.

shc
Number of pages shared due to cache hits.

cpy
Number of shared pages copied.

pf Number of page faults.

system
Reports on the general system activity. Unless the totals since boot are being shown, these figures are averaged over the last *interval* seconds:

in Number of (non-clock) device interrupts.

sy Number of system calls.

cs Number of context switches.

cpu
Percentage of cpu cycles spent in various operating modes:

us User.

su System.

id Idle.

This information may not be displayed on some systems.

The **−f** and **−s** reports are a series of lines of the form:
number description
which means that *number* of the items described by *description* happened (either since boot or in the last *interval* seconds, as appropriate). These reports should be self-explanatory.

Files

/xenix
Default namelist.

/dev/kmem
Default source of statistics.

See Also

fork(S), ps(C), pstat(C)

2-29-88

018-030-031

SCO XENIX® System V

Operating System

Release Notes
Roadmap
Installation Guide
Introduction to XENIX
Operations Guide

The Santa Cruz Operation, Inc.

This document was typeset with an IMAGEN® 8/300 Laser Printer.

Document Number: XG-12-18-85-2.1/1.0

XENIX System V Operating System Release Notes

Release Notes
Release 2.1.0
XENIX® -86 System V for personal computers
XENIX-286 System V for personal computers
Operating System
January 27, 1986

1. Preface

This document contains information about features and programs of XENIX-86 and XENIX-286 System V Release 2.1.0 Operating System for personal computers. *Please read through this entire document before installing the XENIX operating system.*

We are always pleased to hear of user's experience with our product, and recommendations of how it can be made even more useful. All written suggestions are given serious consideration.

2. IMPORTANT NOTES ABOUT INSTALLATION

Please refer to the *Installation Guide* in the binder marked "XENIX Operating System *Run Time Environment*" to install the XENIX Operating System. If you have purchased all three systems, (XENIX Operating System, Development System and Text Processing System) you can use the XENIX *Installation Guide* to install them all at once. If you wish to install the Development System and/or the Text Processing System at a later time, refer to the *Release Notes* at the beginning of the appropriate guide (XENIX *Programmer's Guide "System Development Tools"* or XENIX *Text Processing Guide*). See also the manual page for **custom**(C). **custom** allows you to install all or portions of the XENIX System. Read the *Release Notes* and *Installation Guide* in their entirety and make sure you completely understand the installation process before installing the product.

It is not advisable to abort the installation process (for example, by using the DEL or Ctrl-\ keys). If you need to stop, for example, because you enter incorrect information, start the process again from the beginning rather than trying to proceed from the stopping point.

The three systems are delivered on these (48tpi) floppies:

XENIX Operating System	-volumes N1-N5 (versions 86XT; 6300+)
	-volumes N1-N4 (version 286AT)
	-volumes B1-B3
	-volumes X1-X7
XENIX Development System (Optional)	-volumes D1-D10
XENIX Text Processing System	-volumes T1-T4

Note

Users upgrading their XENIX 3.0 system to XENIX System V should refer to Appendix B "Upgrading to XENIX System V," later in these *Release Notes*. Users installing XENIX for the first time can ignore the Upgrade appendix.

If you use **custom**(C) to install the single file */xenix* (the XENIX kernel) you must serialize the kernel by hand. This is done with the command:

/etc/brand *serialnumber activationkey* /xenix

If you do not do this, the kernel will boot with a very limited number of processes.

2.1 Choosing Swap Area

While installing XENIX System V you are prompted to allocate the amount of swap area you want. If you do not specify the number of blocks of swap area you want, you are automatically given a default amount. This default is minimal. Due to the difficulty in increasing your swap area, it is recommended that you ask for the upper limit amount shown by the **divvy**(C) program, if you are planning to use large applications (like databases or spreadsheet programs), use the software development set extensively, or make many programs "sticky" for performance reasons.

3. Compatible Software

The following software application packages are available from The Santa Cruz Operation for use with XENIX.

SCO FoxBASE™	Relational Database Management System
SCO Professional™	Electronic Spreadsheet
Informix®	Relational Database Management System
LEVEL II COBOL™	Mainframe Level Implementation of the business programming language
Lyrix™	Word Processing System
Multiplan®	Electronic Worksheet
uniPATH™ SNA-3270	SNA Mainframe Communications Package

Call your local dealer or The Santa Cruz Operation for more information and for availability of the following software application packages:

Microsoft Networks for XENIX	PC network compatible LAN
UX-Basic®	BASIC Interpretive Compiler

Your local dealer may also have the following software which users have been able to run on XENIX System V for personal computers:

Microsoft Pascal 3.3 for XENIX-286
Microsoft FORTRAN 3.3 for XENIX-286
RM Fortran V1.1 for XENIX-286
RM Cobol V2.1a for XENIX-86 or XENIX-286

4. Features of the 2.1.0 Release

This section relates important features of the XENIX operating system.

4.1 Autoboot

When your computer is turned on, it can go through all boot stages automatically. Refer to the **autoboot**(M) manual page for information on configuring your system to **autoboot**.

4.2 Binary Compatibility

This section briefly discusses the types of binaries that are known to run on XENIX for personal computers. See the manual page **machine**(M) for a more complete listing of binary compatibility.

4.2.1 XENIX 2.3 Binary Compatibility

The system can execute Microsoft XENIX version 2.3 8086, split i/d (instruction/data), x.out format binaries (for example, Altos 586 or Intel 86/310/330 binaries). Use of Altos specific system calls is not supported.

It is recommended that DOS be installed on the hard disk before installing XENIX. Installing DOS after XENIX, in some circumstances, changes the **fdisk** table making the XENIX partition unbootable. (DOS **fdisk** reports disk size in cylinders, XENIX **fdisk** reports disk size in tracks.) If this happens, you must recreate the **fdisk** table. Boot XENIX from a Bootable XENIX floppy or use the Installation BOOT floppy. Run XENIX **fdisk**(C), delete the XENIX partition, then create the XENIX partition exactly as it was before DOS was installed. If you use the Installation BOOT floppy, delete from the installation procedure after you have run **fdisk**, and reboot the computer. The XENIX partition is once again bootable.

If you want to install XENIX and DOS on two hard disks refer to the chapter "Using XENIX and DOS on the Same Hard Disk" in the XENIX *Installation Guide* and the section titled "Adding a Second Hard Disk" in the chapter "Using Peripheral Devices" in the XENIX *Operations Guide*. Also, refer to the XENIX *Reference Manual* manual pages for **fdisk**(C), **mount**(C), and **mkfs**(C).

The following hard disk devices are new in XENIX System V:

```
/dev/hd0d
/dev/rhd0d
/dev/hd1d
/dev/rhd1d
```

These devices are similar to */dev/hd0a* in that the disk driver determines which partition is the DOS partition and uses that as *hd?d*. This means that software using the DOS partition does not need to know which partition is DOS (the disk driver determines that).

The file */etc/default/msdos* is an easily configurable file that aliases default device names used by the **dos**(C) commands. For example, it now contains the lines:

```
C=/dev/hd0d
D=/dev/hd1d
```

Users using the **dos**(C) utilities can specify "C:" or "D:" on the command line, referring to the DOS partition on the first or second hard disk. For a complete description on using */etc/default/msdos* see the manual page **dos**(C) in the XENIX *Reference Manual*.

The XENIX Development System supports the creation of DOS executable files, using **cc**(CP). Refer to the XENIX *C User's Guide* and *C Library Guide* for information on using XENIX to create programs suitable for DOS systems.

4.5 8087 and 80287 Support

Your personal computer may include the 8087 or 80287 math co-processor, which is supported by XENIX-86 and XENIX-286. Please note that switches on the main system board must be set properly to enable 8087 or 80287 interrupts. Check your hardware manual for the proper switch settings.

At boot time, XENIX will display "math coprocessor present" if an 8087 or 80287 is detected.

Some 8087 and 80287 exceptions have been masked. Refer to the manual page for **8087**(M).

4.6 Link Kit

A Link Kit is provided with the XENIX Operating System. The Link Kit enables you to add device drivers to your system. Additional device drivers are necessary to run non-supported peripheral devices. Refer to Chapter 10 of the XENIX *Operations Guide*, "Installing Device Drivers", for information about using the Link Kit.

In addition, the XENIX *Programmer's Guide* contains two chapters that discuss "Writing Devices Drivers," and "Sample Device Drivers." Also, the Link Kit files themselves include useful instructions and examples (including a driver for real-time clocks). Be sure to read the text files (especially the *README* file) in the directory */usr/sys/conf* if you intend to use the Link Kit.

You may request up to 202 buffers when using the Link Kit with XENIX-286. XENIX-86 has no limit to the number of buffers you can request, but it is possible to create more buffers than available memory. If this happens XENIX will not boot.

In order to save disk space, do not install the Link Kit files unless you intend to use them. The kernel contains the data space necessary for installing device drivers in this 2.1.0 release. **custom**(C) should be used to install (or remove) the Link Kit files.

Available kernel data space (needed to link in large or multiple device drivers) has been increased in the 2.1.0 release. Device drivers which previously did not fit into XENIX might now.

4.7 Memory Limitations

Some utilities may not run on systems with less than 512 Kilobytes(K) of main memory. **vsh**(C) (the Microsoft visual shell) and **vi**(C) (the Berkeley full-screen editor) are examples of these. When you invoke a utility that needs more memory than available, the message "Killed" displays on your screen. If your system has 384K of main memory we recommend using XENIX with only one screen enabled. Multi-user mode will function, but performance suffers, with 384K memory.

vi(C) and **vsh**(C) are supported for machines with 512K of main memory or greater. Use of any medium model programs may severely impair system performance with less than 512K of main memory.

We recommend using the XENIX Development System only on machines with 512K of main memory or greater.

4.8 mkdev(C)

The XENIX Operating System now allows you to add additional filesystems more easily by using the **fs** option with **mkdev**(C). See the manual page for **mkdev**(C) included with these *Release Notes*, Chapter 2 "Installation Procedure" in the XENIX *Installation Guide*, and Chapter 7 "Using Peripheral Devices" in the XENIX *Operations Guide*.

4.9 Multiscreen™ and Color Support

The console under XENIX can act as a console and up to 9 alternate terminals, each of which can support different activities (see the manual pages for **console**(M) and **multiscreen**(M) in the XENIX *Reference Manual* and the chapter "Using Peripheral Devices" in the XENIX *Operations Guide*).

Note

The number of screens available depends on the amount of memory in your computer, and is displayed when the system boots. The actual number of screens **enabled** is user configurable. See **enable**(C) to use the available screens.

When error messages from the XENIX kernel appear, they display on the console screen. When this happens the console screen displays instead of the current screen.

If you have a color monitor you can use the XENIX **setcolor**(C) utility to select the foreground, background, and graphics screen colors from a palette of 16 colors (see the manual page for **setcolor**(C) in the XENIX *Reference Manual*).

4.10 System V sh shell

Included in this release is the "System V sh command programming language", found in */bin/shV* and invoked as **shV**. There are no known problems with this shell, but it is not supported. Let us know if you encounter problems with **shV**. It has the same functionality as **sh**(C) found in this and previous releases, with the following additions:

"command" function
MAILCHECK parameter
MAILPATH parameter
SHACCT parameter
SHELL parameter
input/output redirection for some commands
hash −r special command and flag
pwd special command
return special command

See the **shV**(C) manual page included with these *Release Notes*.

4.11 Software Reboot

XENIX System V allows software rebooting. You need not power off to reboot. The following message is displayed after a system shutdown:

** Safe to Power Off **
-or-
** Hit Any Key to Reboot **

Fixed disk heads are "parked" (locked) when this message is displayed.

Note

When the floppy drive door is closed, the computer attempts to boot from a floppy. The floppy drive door must be open to boot off the hard disk.

4.12 SVID Conformance Notes

XENIX-286 meets the System V Interface Definition (SVID) published in 1985 by AT&T (Select Code 307-127). XENIX System V has been subjected to a stringent set of tests for the purpose of verifying conformance to SVID. SCO XENIX System V conforms to SVID functionality with the following exceptions:

Function	SVID Specification	SCO XENIX Implementation
shmat	Allows non-zero arguments.	shmaddr argument must be zero. The SVID states that shmat() should allow a non-zero shmaddr argument. The segmented architecture of the 86 family precludes allowing attachment of shared memory at *specific* physical addresses. Because of this, SCO XENIX requires that the shmaddr argument be zero to allow the system to choose the first *available* address.
shmop shmctl shmget	Shared memory operations. Return "char *shmaddr"	Because of the architecture of the 86 family of chips, SCO XENIX chose to implement shared memory by using far pointers. The SVID "char *shmaddr" is replaced with "char far *shmaddr." This is for the same reason as listed for *shmat* above.

Function	SVID Specification	SCO XENIX Implementation
ptrace	Address specified as (int*)	Address specified as structure shown in section 4.6 of the *Development System Release Notes*. The SVID states that ptrace() calls should fail if the 'addr' argument is not the start address of a word. Because of word alignment conventions in the 86 family of chips this failure will not occur.

These exceptions are found on the **termio** manual pages (**termio**(dev) for SVID and **termio**(M) for XENIX):

Termio Characters/values	SVID Specification	SCO XENIX V Implementation
QUIT	Ctrl-|	Ctrl-\
ERASE	#	Ctrl-H
KILL	@	Ctrl-U
c_cc value	7 SWTCH	not included
38400 baud	B38400	not included
Block layer output - hardware control	LOBLK	not included
default initial baud rate	B300	B9600
ISIG causes input characters to be checked against	INTR, SWTCH and QUIT	INTR and QUIT

The *c_cc* value of SWTCH, the block layer output value of LOBLK, and ISIG checking against SWTCH are all related to the implementation of job control. XENIX System V Release 2.1.0 does not support job control.

4.13 Ulimit

The maximum file size, as specified by **ulimit**(S), is easily adjusted upward. The **login**(M) utility checks the file */etc/default/login* for the default **ulimit** size. Refer to the manual pages for **default**(M), **login**(M), and **ulimit**(S).

5. Hardware Notes

This section contains notes relating to hardware issues.

In general, your hardware configuration must have the original settings and boards before you boot XENIX. If you have added any boards, make sure that all switches are set as recommended in the manufacturer's hardware manual for that board. These guidelines must be followed to ensure proper system performance.

XENIX must boot from the physical drive 0 (the internal hard disk).

5.1 Clockrate Adjustment

The clocks of some computers operate at a different frequency than the default setting. You can check the accuracy of your clock rate using **date**(C) command over a period of days. If the system clock gains or loses time excessively, refer to the manual page **clockrate**(C) for information on setting the clock rate.

The default clock rate for generic 8086 and 80286 machines is 1.19318. The correct clock rate for an AT&T 6300 or Olivetti M24 is 1.22878.

5.2 IMAGEN printers – ips(C)

When using an IMAGEN printer in parallel printer mode (using **ipbs**, see **ips**(C)) you must specify the quote character as ASCII 2 and the EOF character as ASCII 4. Control characters must be "taken as is". Refer to the IMAGEN system manuals provided with the printer, for information on specifying these characters in the printer configuration.

5.3 Serial I/O Chips

Some computers or add on serial I/O cards use the 8250a serial I/O chip. This chip does not handle interrupts properly. DOS does not use interrupts, so the use of this chip with DOS causes no problems. XENIX does make use of interrupts, as it is a multi-tasking operating system. The problem with the serial I/O chip shows up when using **uucp**(C) or **cu**(C). Indications that your computer contains an 8250a chip are that **uucp**(C) may lose characters constantly and generate unkillable **uucico** processes and that **cu**(C) at high baud rates hangs and will not exit.

The problem rarely shows up when using the serial port with a terminal. It is associated with high-speed serial input. If you want to use **uucp**(C) or **cu**(C) and your computer has one of these chips, we recommend you replace the 8250a chip with an 8250b serial I/O chip or use a multi-function card containing a serial port and configure it as COM1 or COM2. Configure the built-in serial port as COM3 or COM4, and avoid using the port with high-speed input.

5.4 Serial Lines

There are 4 and 8 port serial port expansion cards available. The ports on these cards are accessible only when the jumpered I/O address is properly set. The addresses vary for different manufacturers of boards, and for the port. The following table lists (hexadecimal) addresses for the Hostess, Arnet, AST and Sperry expansion cards. The Hostess and Arnet cards are available in 4 and 8 port versions. The AST and Sperry cards are available in 4 port version only. The Sperry card works with the Sperry PC/IT only.

Serial Card Addresses					
Physical Port	Board Type	Primary I/O Address	Primary Status Address	Alternate I/O Address	Alternate Status Address
COM1	Hostess	0x500	n/a	0x680	n/a
	Arnet	0x100	0x140	0x280	0x2C0
	AST	0x2A0	n/a	none	n/a
	AMI lamb*	none	n/a	none	n/a
COM2	Hostess	0x580	n/a	0x700	n/a
	Arnet	0x180	0x1C0	0x300	0x340
	AST	0x1A8	n/a	none	n/a
	AMI lamb*	0x2C0	0x210	none	n/a
OTHER1	Sperry**	0x400	n/a	none	n/a
OTHER2	Sperry**	0x408	n/a	none	n/a

* Only continuous mode is supported.

** These serial cards only work on the Sperry PC/IT.

Check your serial card hardware manual or call the hardware manufacturer for the switch settings that implement these addresses.

Refer to Appendix B "Compatible Hardware" of these *Release Notes* and the serial(M) manual page for more information on compatible serial I/O cards and serial lines.

6. Software Notes

This section explains some software issues of note.

6.1 Include Files and Utilities

Machine dependent Development System include files and utilities are included on the 2.1.0 Operating System N*n* Volumes. If you do not install the Link Kit and the Development System set, you will not be prompted to extract any files off some of the NFIn volumes.

.2 crypt(C)

The **crypt**(C) command has been removed from XENIX System V. If you want the **crypt**(C) utility and associated **crypt**(S) libraries, and you live in the United States, contact the support center listed on the support information card included with the software.

.3 csh(C) - sh(C) Conflicts

The C-shell, **csh**(C), has many built-in commands with the same name and functionality, but different syntax, than those used by the **sh**(C) shell. The **sh**(C) commands are those documented in the XENIX *Reference*. An example of two commands which exhibit this behavior are **echo** and **nice**. Refer to the manual page **csh**(C) for information on built-in C-shell commands.

An */etc/cshrc* file is included with this release. You can customize the C-shell environment for all **csh**(C) users by using this file (similar to */etc/profile* for **sh**(C) users).

6.4 fdisk(C)

The **fdisk**(C) utility does not allocate the last track on the hard disk unless the "Use Entire Disk for XENIX" option is used. The "Create XENIX Partition" option always leaves the last track unassigned.

For example, if a disk has 2442 tracks, **fdisk** reports these as tracks 0-2441. It will assign (using the "Create XENIX Partition" option) tracks 1-2440. Track 0 is reserved. Track 2441 is only assigned with the "Use Entire Disk for XENIX" option.

6.5 format(C)

The **format**(C) utility does not format floppies for use under DOS. Also, XENIX requires error free floppies.

6.6 lp(C)

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 **boots**. This is done on the first open only so that printers with large text buffers will not be 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. Make a new device file that sends an *init* every time it sends a file:

   ```
   #  /etc/mknod /dev/lp0i c 6 128
   #  sync
   ```

   The commands for the other parallel ports are:

   ```
   #  /etc/mknod /dev/lp1i c 6 129
   #  /etc/mknod /dev/lp2i c 6 130
   ```

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 doesn't have a large internal text buffer) you can use the */dev/lp0i* device all the time:

```
# rm /dev/lp
# ln /dev/lp0i /dev/lp
```

The **lp**(C) command now sends an **init** every time a file is sent to the printer.

Some printers do not have newline/carriage return mapping. If your printer needs to have newlines mapped to newline/carriage returns, specify the *crnlmap* filter when you set up the printer using the **lpinit**(C) command. When you are prompted for the type of printer (dumb printer, Imagen laser printer (parallel interface), Imagen laser printer (serial interface), or other), enter "other". You are then prompted for the pathname of the interface program. The printer filter file is found in */usr/spool/lp/model/crnlmap*.

6.7 mvdir

You must be super-user (root) to use this command.

6.8 pg(C)

This release now includes the **pg**(C) utility.

6.9 runbig(C)

The utility **runbig**(C) is not included with this release. It will be included in future releases.

6.10 uname(C)

No XENIX utilities currently use the node name in the **uname** structure. XENIX utilities use the entry in */etc/systemid*. However, some sites may want to set the node name in the **uname** structure.

There are two ways to set the node name in the **uname** structure but you must have the XENIX Development System to make the change either way. You can use **adb**(CP) to patch the XENIX kernel, or you can use the link kit to recompile and install a new kernel.

Refer to the *Development System Release Notes* for instructions on using **adb** to set the node name.

You can use the link kit to create a new kernel, where the only change is the node name. Install the link kit (use **custom**(C)). Use a text editor to change the:

    node=

entry in the file */usr/sys/conf/c.c*. Follow instructions in Chapter 10 of the XENIX *Operations Guide* "Installing Device Drivers" to create a new */xenix* and install it.

6.11 Using Raw Devices

When using **dd**(C) or **tar**(C) with a raw device, specify the block size as a multiple of 1K, which is equivalent to 2 "b." This is recommended because **dd** (C) and **tar**(C) use 512 byte blocks. The XENIX filesystem uses 1024 byte blocks. For example, for **dd** (C) using a 9K (system) block size use:

    dd if=file of=/dev/rfd0 bs=18b

Specify an even blocking factor with **tar**(C):

    tar cvfb /dev/rfd0 18 file

or use the default tar settings (see **tar**(C)).

6.12 uucp(C) Notes

This section answers questions you may have about **uucp**(C) and **cu**(C). The installation of the **uucp** programs and utilities is optional. Refer to the chapter "Building a Micnet Network" in the XENIX *Operations Guide* and the chapter on "Building a Communications System" in the XENIX *User's Guide*. Also refer to **custom**(C) to install (or remove) communications files.

6.12.1 Modem Usage under XENIX

Refer to the XENIX *Operations Guide* chapter "Using Peripheral Devices" for information on modem settings, modem control, aliasing alphabetic characters and dialing with your computer.

6.12.2 Autodialing

The manual page **dial**(M) includes an example C program for using a Hayes modem with **uucp**(C). This program, unlike the distributed *dial* program, does not make entries in the *LOGFILE*.

In addition to the program on the **dial**(M) manual page, the C language program, */usr/lib/uucp/dial.c*, for the distributed */usr/lib/uucp/dial* is included in this release. You can copy and modify *dial.c* or you can use these as examples to write programs to use other kinds of modems. The makefile distributed in */usr/lib/uucp* should be modified to reflect your own dialer program. You must have the XENIX Development System in order to compile C programs.

If you create a *dial* program for another modem, send us the source. User generated *dial* programs will be considered for inclusion in future releases.

The **uucp** dialup timeout has been increased with this release. This increase is sufficient to allow **uucp** to dial long distance numbers on pulse dial telephone lines.

6.12.3 L-devices and L.sys

We recommend running **uucp**(C) direct connections at 2400 or 4800 baud so that the system's performance is not impaired. These baud rates are specified in the files */usr/lib/uucp/L-devices* and */usr/lib/uucp/L.sys*.

Do not use tabs as field separators in the *L.sys* file.

6.12.4 Communications with 3.0 XENIX Systems

If you have trouble connecting to a XENIX System 3.0 with **uucp**(C), try making that system "active" and the SCO XENIX System V the "slave".

6.13 vi and vedit

The **vedit** utility is supported in this release. It is invoked by typing:

vedit *filename*

It has essentially the same functionality as **vi**. The difference is in the option settings. **vedit** uses *novice* mode which prevents job control character (Ctrl-Z) mapping, changes the default setting, turns off the **magic** option, sets the option **report=1** and calls the **showmode** option.

The **showmode** option informs the *vedit* user, in a message in the lower right hand corner of the screen, which mode is being used. For instance after the **ESC−i** command is used, the message reads "INSERT MODE".

Note that you can not set the **novice** option from within *vi* or *ex*. If you want to use the **novice** option you must use the *vedit* utility. (It is possible to set the **nonovice** option from within *vedit*.)

6.14 vsh (C)

The use of special characters (*, [,], and ?) is explained in the chapter "Basic Concepts" in the *Introduction to XENIX*. The visual shell, **vsh**(C), does not currently support the use of these special characters when specifying file names.

The super-user (root) must initialize the visual shell help file before other users can have access to it. To do so, follow these steps:

1. Log in as super-user (root).

2. Invoke the visual shell by entering "vsh" on the command line.

3. When the **vsh** menu comes up, enter "h" (for help).

4. The help file appears, with another menu. Enter "r" (for resume).

5. The main **vsh** menu appears again. You can now quit from the v-shell. Enter "q" (for quit), then "y" (to confirm the quit).

The **vsh** help file is now initialized so that all users have access to online help.

7. Documentation Notes

This section lists errors, changes, and general notes about the documentation.

7.1 Blank Pages in the User's Reference

There are several blank pages in the *User's Reference* "(M)" section. They can be removed from the following locations:

boot(M)	1 blank page
console(M)	3 blank pages
fd(M)	1 blank page
keyboard(M)	4 blank pages
machine(M)	1 blank page

7.2 Hardware Related Manual Pages

The following hardware dependent manual pages are located in the (M-HW) section found at the end of the *Run Time Environment* volume:

boot(M)
cmos(M)
console(M)
8086(M)
fd(M)
hd(M)
keyboard(M)
lp(M)
machine(M)
parallel(M)
serial(M)

7.3 Replacement Manual Pages

The following manual pages have been revised or added. Please replace the manual pages found in the *User's Reference* with the ones included here. The replacement pages are dated "January 27, 1986".

mkdev(C)
shV(C) - new, insert between shutdown(C) and sleep(C)

7.4 console(M)

The following escape sequences have functions which are not documented on the manual page **console**(M). These features are experimental extensions to the ANSI specification and are subject to change or removal from other XENIX releases. They are present and functional in the XENIX System V 2.1.0 release:

ESC[=*C*A	where *C* is 0-15. Sets border color on color adaptors. Color codes are same as for ESC[2;C1;C2m.
ESC[=*F*;*T*B	Sets bell frequency and duration. *F* is the value loaded into the timer-counter, and *T* is the bell duration in 1/10ths of a second (initial settings are 1500 and 2/10ths of a second).
ESC[=*Ss*;*Se*C	Sets cursor to start at scan line *Ss*, and end at scan line *Se*. Default underline cursor is 6,7 for color adaptor and 10,2 for monochrome.

Another experimental function is the ioctl call for setting the screen color from inside a program:

attr = ioct(l, GIO_ATTR, 0)

The attribute is returned in ioctl(),'s return code. The foreground color is in the least significant nibble.

7.5 C-shell

There are four **csh**(C) **aliases** distributed with the XENIX System V **csh**. These are *pushd*, *popd*, *swapd*, and *flipd*. These **aliases** maintain a directory stack. They function as follows:

pushd *dir* Pushes the current directory onto the top of the directory stack, changes to the directory *dir*.

popd — Changes to the directory at the top of the stack, then removes (pops) the top directory from the stack, and announces the current directory.

swapd — Swaps the top two directories on the stack. The directory on the top becomes the second to the top, and the second to the top directory becomes the top directory.

flipd — Flips between two directories, the current directory and the top directory on the stack. If you are currently in *dir1*, and *dir2* is on the top of the stack, when *flipd* is invoked, you change to *dir2* and *dir1* is replaced as the top directory on the stack. When *flipd* is again invoked, you change to *dir1* and *dir2* is again the top directory on the stack.

7.6 master(F)

Page 1 of the **master**(F) manual page says:

Field 2: interrupt vector size (decimal, in bytes)

It should read:

Field 2: number of interrupt vectors

Seven lines down it says:

000040 clock ticks

It should read:

000040 clock poll handler

Eight more lines down it says:

000100 init handler

It should read:

000100 no qswitch on interrupt

Page 2 of the **master**(F) manual page says:

Fields 11-14: maximum of four interrupt vector addresses.

It should read:

Fields 11-14: maximum of four interrupt vector numbers.

7.7 su(C)

In order to enable the logging of super-user **su**(C) attempts, the file */etc/default/su* must contain the line:

SULOG=/usr/admin/sulog

8. Restoring Lost Files

The **custom**(C) utility is used to list, install, or remove individual files, sets within the XENIX System packages, or entire packages of the XENIX System. Refer to **custom**(C) in the XENIX *Reference Manual* for information on using **custom**.

4.2.2 XENIX 3.0 Binary Compatibility

The system can execute Microsoft XENIX version 3.0 8086, split i/d (instruction/data), x.out format binaries (for example, Altos 586 or Intel 86/310/330 binaries). Use of Altos specific system support calls is not supported.

4.2.3 XENIX System V Binary Compatibility

The binaries produced by the XENIX C compiler will run on SCO XENIX-286, XENIX-186, and XENIX-86 Systems, as well as Microsoft 286 XENIX 3.0 or System V and IBM XENIX 3.0 or System V.

Machines with SCO XENIX-286 installed can run Microsoft or IBM 286 XENIX 3.0 or System V binaries.

4.3 Configurable Console Keyboard and Screen

XENIX System V Release 2.1.0 supports console keyboard and console screen reconfiguration. You can reconfigure the keyboard for a different keytop layout, or program some keys, including function keys, to send strings of characters.

You can map characters so that, on output, they display as international or graphics characters. For example, you can map "e" to display as "ê." Refer to the **console**(M), **keyboard**(M), **mapkey**(M), and **setkey**(M) manual pages in the XENIX *Reference Manual* for information on configuring the console keyboard.

/usr/lib/keyboard and */usr/lib/console* contain files for British English (UK) and German keyboard layouts and screen mapping.

The XENIX kernel read buffers hold up to 256 characters. Overflow of a buffer causes it to be flushed and the 257th character is not echoed. If this 257th character is dropped during the output of an escape sequence, the output of that sequence is other than expected.

For example, if a function key is programmed to send "^[154g" and the function key is held down so that the output of this string on the screen repeats, the 257th string character, "4," is dropped and the string interpreted would instead be "^[15g" that one time.

Refer to section 7.1 "**console** (M)" for more information on the console.

4.4 DOS support

XENIX supports the coexistence of MS™-DOS and XENIX on the same hard disk. Consult Chapter 3 of the XENIX *Installation Guide* and the manual entries for **fdisk**(C), **divvy**(C), and **dos**(C). The **fdisk**(C) and **divvy**(C) utilities create and change multiple disk partitions, allowing separate XENIX and DOS partitions. **fdisk**(C) has similar functionality to the DOS utility of the same name. The utilities mentioned in **dos**(C) allow access to DOS files on the hard disk and floppy diskettes.

Should you decide to install XENIX and DOS on the same hard disk, be sure to reserve a sufficiently large partition for XENIX.

Note

The minimal XENIX System configuration you can install requires approximately 2000 1K blocks, or 2M bytes. The entire XENIX System (XENIX Operating System, XENIX Developmanet System and XENIX Text Processing System) requires approximately 12,000 1K blocks, or 12M bytes. Refer to the manual page **custom**(C) for information on installing or removing portions of the XENIX System.

---

9. Packages In This Set

These are the packages in the Operating System:

Operating System Packages

ALL	Entire operating system set
INST	XENIX installation utilities
PERM	XENIX contents and permissions lists
RTS	XENIX run time system
BASE	Basic extended utility set
BACKUP	System backup and recovery tools
SYSADM	System administration tools
FILE	File manipulation tools
LINK	The link kit
LPR	Multiple line printer spooler
MAIL	Electronic mail and local area networking
CSH	The C-shell
DOS	DOS utilities
VSH	The visual shell
EX	The **ex** and **vi** editors
UUCP	**uucp** and **cu** communications utilities
*GAMES	Fun and games

Note

* Games are supplied, as is, for your pleasure and enjoyment. They are NOT supported. On line documentation is supplied for some of the games, in the directory */usr/games/doc*.

A. Compatible Hardware

This appendix describes hardware that can be used with XENIX System V.

The hardware listed in this section has been used with XENIX. However, because compatible machine or add-on peripherals manufacturers may change configuration or functionality of firmware at any time, no guarantee of functionality is implied. Please write us with accurate hardware information for possible inclusion on our lists.

Some of the hardware listed requires "vendor supplied drivers." Ask your hardware vendor for the appropriate device driver for SCO XENIX V. Be forewarned that device drivers intended for versions 2.0.3 or earlier may not function with this 2.1.0 release. You must link device drivers, supplied by the hardware vendors, into the XENIX kernel in order to use some of these devices. For information on linking device drivers, see the section "Link Kit" in these *Release Notes* and Chapter 10 of the XENIX *Operations Guide* "Installing Device Drivers."

A.1 XENIX−86 System V

A.1.1 Machines

These machines can run XENIX-86 System V:

AT&T 6300 (Olivetti M24 in Europe)
Compuserve/Canada Arthur in Canada
Compaq Deskpro* and Plus*
Eagle Turbo
Ericsson PC
IBM PC XT
ITT XP
ITT XTRA
Leading Edge
Mitsubishi
NCR PC-6**
Tandy 1200
Sperry
WYSE PC

* Internal cartridge tape not supported.
** For the NCR machine, configure the switches on the motherboard for a color monitor, regardless of the type of monitor used.

Note

ITT DOS cannot share the disk with XENIX or MS-DOS. Therefore, XENIX, or a combination of XENIX and MS-DOS must occupy the whole disk on the ITT machines.

Also note that the ITT XP (286 cpu) runs in 8086 mode.

Some computers arrive with the hard disk only partially formatted. If you have such a machine, use the DOS hard disk format command or Diagnostics diskette to format the entire disk before installing the XENIX Operating System.

Some computers require specific switch settings to run XENIX. If your computer does not run XENIX with the settings as shipped, contact your computer hardware representative for the proper settings.

Some machines may require a clock rate adjustment. Refer to section 5.1 of these *Release Notes*.

A.1.2 Math Chips

You can use these math co-processor chips under XENIX with the following Central Processing Units (CPUs):

8087	For 8088 CPUs.
8087-2	For 8086 CPUs.
8087-3	For fast 8086 CPUs.

A.1.3 Memory Cards

With memory cards, check the switch settings on both the card and mother board. Refer to the hardware manuals for your computer and for the memory card to find the correct switch settings. XENIX-86 supports up to 640K of main memory.

The following memory cards have been used successfully with XENIX-86:

AST 6 pack+
Quadram quadboard
Sigma Maximizer
Microsoft RAM card
Tecmar Captain or First-Mate
AT&T for the 6300

AT&T 6300 note: It is recommended that only AT&T memory cards be used with the 6300. Other cards may considerably slow the machine.

Note

In general, most memory cards work with XENIX-86. If you experience "panic parity" errors it is often because of low quality memory chips or cards.

The serial ports on many multi-function cards will function as expected if COM1 and COM2 are fully compatible with the IBM specifications for these serial ports. See section 5.4 "Serial Lines" and section A.1.5 "Serial I/O Boards" in these *Release Notes*.

A.1.4 Accelerator Boards

Accelerator boards are special cards that plug in to an 8088 machine (not an 8086). They replace the 8088 CPU with a 286 CPU and perhaps replace some memory as well. XENIX System V runs on accelerator boards set in 8086 unmapped mode.

The following accelerator board is supported in the standard distribution:

Seattle Telecom and Data (STD) PC286

A.1.5 Serial I/O Boards

Note that you should not use cards that use the 8250a serial I/O chip. See the Hardware Notes section in these *Release Notes* on "Serial I/O Chips" for more information on the 8250a chip.

These serial I/O boards are supported in the standard distribution:

IBM standard COM1 and COM2
Control Systems Hostess 4 and 8 port versions
Arnet Controls 4 and 8 port (8250b version)
(clock option not supported)
AST FourPORT/XN

A.1.6 Tape Units

The following tape devices are supported using a device driver supplied by the hardware vendor:

Emerald Systems series 2002	-	streaming cartridge
Computone ATvantage-SX Tape backup	-	streaming cartridge
Overland Data TC50X	-	nine track (reel to reel)

A.1.7 Video Cards and Monitors

The following video cards and monitors have been used successfully with XENIX-86. (They are to be used in character mode only, graphics are not currenty supported.)

Video Cards and Monitors	
Card	Monitor
IBM Mono Card	standard monochrome
AST CGA Everex Edge IBM CGA	standard RGB (red green blue) or composite color
IBM EGA	standard monochrome standard RGB color IBM EGA

Cards which come as the default card in supported computers (monochrome or color monitor) work fine with XENIX.

Although many standard emulating cards work with XENIX, these are the ones that we know about.

A.1.8 Networking Cards

This networking card is supported using a device driver supplied by the hardware vendor:

Excelan TCP/IP ethernet card

A.1.9 Add On Hard Disks

Many hard disks can be used by XENIX-86 as long as the disk controller supports the drive. Compatible hard disk controllers are discussed in the next section.

The following tables list the controllers most commonly used with 8086/8088 machines, and the type of hard disks they can support:

XEBEC Controller
(standard IBM PC XT compatible)

Type	Cyls	Hds	Size	Example
0	306	2	5 MB	
1	375	8	24 MB	
2	306	6	15 MB	
3	306	4	10 MB	standard

DTC Controller
(standard on AT+T 6300 - fully supported)

Type	Cyls	Hds	Size	Example
0	306	2	5 MB	
1	375	8	24 MB	
2	306	6	15 MB	
3	306	4	10 MB	
4	306	2	5 MB	Syquest SQ306 standard
5	644	5	30 MB	CDC Wren
6			20 MB	Seagate ST-225
7			20 MB	Miniscribe 3425
Drive types 8 - 15 reserved				

The following hard disks have been tested successfully with XENIX-86:

Disk	Controller
CMI 15	XEBEC
Miniscribe 20	DTC
Seagate 20	DTC
CDC 30	DTC
Rhodime 30	I^2
Mountain Hard File	built-in controller
Mountain 20, 30	DTC

The following disks are reported to work with XENIX-86, but are untested:

Peachtree 10, 20, 30
Maynard Apollo 30
Franklin Telecom 10, 20, 33
Rhodime 10
Tandon 10
Alpha Omega 10, 20, 30

To determine whether you can use a hard disk with XENIX, the disk must first interface electrically with the disk controller (usually "ST506"). The disk controller ROM must have an entry for the disk (*type*) determined by the number of heads, and the number of cylinders on the disk. The number of cylinders on your disk must be greater than or equal to the number of cylinders listed for that *type*. If these conditions are met, you can use the hard disk, but XENIX can only access up to the number of cylinders in the list.

For example, a "Brand X 40" hard disk has 8 heads and 700 cylinders. It is a type 1 disk on a XEBEC or DTC controller. You can use the Brand X disk, but XENIX can access only 375 cylinders (24 megabytes) of the disk. The only way to access the full disk is to replace the ROM on the disk controller with one that recognizes your disk configuration.

A.1.10 Compatible Hard Disk Controllers

Many hard disks will work with XENIX. Whether or not a disk works depends upon the controller board. Here are two tests the controller must meet:

1. The disk controller board is XEBEC compatible; and,

2. There is no special vendor software needed to make the disk work under DOS.

If a controller meets these tests, it might work, but if it fails these tests, it will not work.

The following table lists the hard disk controllers known to work with XENIX-86:

Manufacturer	Model#	Comments
XEBEC	1210A	PC XT std. Miniscribe 10mb as well as CMI 15
DTC	5150	6300 std. Miniscribe 10, 20, Seagate 20, CDC 30, Mountain 30
WD	1002/WX-2	a special version of WX-2 is supplied with some CDC 30 disks.
Adaptec	2002	
I^2 interface		Rhodime 30
ADES		
CDC		Comes with the CDC 30 add on hard disk.

A.2 XENIX−286 System V

This section discusses some of the hardware known to run with XENIX-286.

A.2.1 Machines

These machines can run XENIX-286 System V:

Compaq Portable* and DeskPro* 286
Contel/CADO AT/4
Corona ATP
Epson Equity III
IBM PC AT (20 or 30 meg version)
Kaypro 286i
Mitsubishi-286
NCR PC-8 **
Sperry PC/IT
Tandy 3000
Tomcat 3200-AT
Victor V286

* Internal cartridge tape not supported.
** For the NCR machine, configure the CMOS database for color monitor, regardless of the type of monitor used.

Although the Texas Instruments Business Pro (color monitor only) will run SCO XENIX, TI XENIX is directly available from TI (and supports more peripherals on TI machines).

Note

For machines with bus or processor speeds greater than 6 mHz, check with your peripheral vendor to verify that their hardware will run under your configuration.

A.2.2 Math Chips

You can use the following math co-processor chip with XENIX-286:

80287	For 80286 CPUs

A.2.3 Memory Cards

With memory cards, check the switch settings on both the card and mother board. Refer to the hardware manuals for your computer and for the memory card to find the correct switch settings. XENIX-286 supports up to 16 megabytes of main memory, however, hardware modifications might be necessary to add this much memory.

The following memory cards have been used successfully with XENIX-286:

AMI SMART PACK 2
AST
Quadram
Tecmar
Talltree Systems
Silicon Valley Systems

Note

In general, most memory cards work with XENIX-286. If you experience "panic parity" errors it is often because of low quality memory chips or cards.

A.2.4 Serial I/O boards

Note that you should not use cards which have the 8250a serial I/O chip. See the section in these *Release Notes* on "Serial I/O" for more information on the 8250a chip.

The following serial I/O boards are supported in the standard distribution:

Arnet Controls 4 and 8 port (8250b version)
(clock option not supported)
AST FourPORT/XN
Control Systems Hostess 4 and 8 port versions
IBM standard COM1 and COM2 cards
Sperry* 4 port card

* This serial I/O board only works with the Sperry PC/IT.

This serial I/O board is supported using a device driver supplied by the hardware vendor:

Computone ATvantage-X smart serial expansion

A.2.5 Tape Units

The following tape devices are supported using a device driver supplied by the hardware vendor.

Computone ATvantage-SX tape backup	-	streaming cartridge
Emerald Systems series 2002	-	streaming cartridge
Overland Data TC50X	-	nine track (reel to reel)

A.2.6 Video Cards and Monitors

The following video cards and monitors have been used successfully with XENIX-286. (They are to be used in character mode only, graphics are not currenty supported.)

Video Cards and Monitors	
Card	Monitor
IBM Mono Card	standard monochrome
AST CGA Everex Edge IBM CGA	standard RGB (red green blue) or composite color
IBM EGA	standard monochrome standard RGB color IBM EGA

Cards which come as the default card in supported computers (monochrome or color monitors) work fine with XENIX. The TI Business Pro, however, must have a color card and a color monitor.

Although many standard emulating cards work with XENIX, these are the ones that we know about.

A.2.7 Networking Cards

This networking card is supported using a device driver supplied by the hardware vendor:

Excelan TCP/IP ethernet card

A.2.8 Add On Hard Disks

Many hard disks can be used by XENIX-286 as long as the motherboard ROM supports the drive. ROM entries for the IBM PC AT are listed here. Refer to your computer hardware manual for the disk types supported by other computers.

PC AT Disk Types Entries on Motherboard ROM

Type	Cyls.	Hds.	Size	
1	306	4	10 MB	
2	615	4	20 MB	
3	615	6	30 MB	
4	940	8	60 MB	
5	940	6	45 MB	
6	615	4	20 MB	no precompensation
7	462	8	30 MB	
8	733	5	30 MB	
9	900	15	110 MB	
10	820	3	20 MB	
11	855	5	35 MB	
12	855	7	50 MB	
13	306	8	20 MB	
14	733	7	40 MB	
15	0	0	Reserved, do not use.	

Note that other compatible XENIX-286 machines may have different ROM tables. Check your computer hardware reference manual for the appropriate ROM table entries for your computer.

The following hard disks have been tested with XENIX-286:

CDC 20	-	with PC AT only
Emerald 50, 70, 140	-	with PC AT only
Maxtor 140	-	with PC AT as Type 9
CPC 30	-	with Sperry PC/IT only

To determine whether you can use a hard disk with XENIX, the disk must first interface electrically with the disk controller (usually "ST506"). The motherboard ROM must have an entry for the disk (*type*) determined by the number of heads, and the number of cylinders on the disk. The number of cylinders on your disk must be greater than or equal to the number of cylinders listed for that *type*. If these conditions are met, you can use the hard disk, but XENIX can only access up to the number of cylinders in the list.

For example, a "Brand X 140" hard disk has 15 heads and 1000 cylinders. It is a type 9 disk on the AT. You can use the Brand X disk, but XENIX can access only 900 cylinders (110 megabytes) of the disk. The only way to access the full disk is to replace the ROM on the motherboard (or other hardware modification) with one that recognizes your disk configuration.

A.3 Modems and Autodialing

Any standard RS-232 modem will work with XENIX-86 and XENIX-286 for personal computers using **uucp**(C) and **cu**(C). The default autodialer is for the Hayes Smartmodem 1200. An autodial program is also supplied for the Racal Vadic 3451. Other autodialing modems can be supported by writing a dialer program. See the manual page **dial**(M) in the XENIX *Reference Manual* and section 6.12.2 of these *Release Notes* for information on writing other dialer programs.

B. Upgrading to XENIX System V

The following procedure shows you how to upgrade XENIX-86 3.0 and XENIX-286 3.0 to XENIX System V. This procedure does not apply if you are installing XENIX System V for the first time

Part of the process is to completely reinstall XENIX. With a complete reinstallation you are assured of having all the latest software, your disk is less fragmented when you are finished, and, if you want, you can easily change the size of the XENIX partition or the size of the XENIX swap zone at this time.

This procedure does not affect any other operating systems (or partitions), such as MS-DOS, that currently share the disk with XENIX 3.0, unless you choose to change partition sizes. Refer to Chapter 3 of the XENIX *Installation Guide* "Using DOS and XENIX on the Same Disk" if you want to change the size of your XENIX partition.

Before proceeding, be sure to read all of these *Release Notes*, including new manual pages. Note changes in the software since the last time you installed XENIX.

Follow these steps to upgrade your XENIX system:

1. Shut the system down to Single User (System Maintenance) Mode.

2. Make a full backup of your system.

3. Save particular files from your current system.

4. Install XENIX System V.

5. Relink the kernel, if necessary.

6. Reinstall any applications software packages.

7. Merge the saved files with the reinstalled system.

Note that these are the basic steps and the exact procedure may vary from site to site. Examples are given wherever possible, but you should know how to use commands such as **tar**(C) (or **cpio**(C)).

Remember that an initial pound sign (#) in the examples is the root prompt and is a reminder that you should execute the listed command only when logged in as root. You do not enter the pound sign as part of the command.

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. Check with the supplier of the other software for further details.

When upgrading from SCO XENIX-86 System 3.0 to SCO XENIX-86 System V, you must save applications data files in ASCII or symbolic form, not as binaries. This is because the word order was changed from System 3.0 to System V in XENIX-86.

Binaries from SCO, IBM, or Microsoft XENIX-286 System 3.0 are compatible with SCO XENIX-286 System V binaries. However, saving data files in ASCII or symbolic form is still a good idea.

When you restore files after reinstalling XENIX, use the applications to convert your data back to binary form. The procedure for saving data files is discussed in a following section "Data Files and **ar**(CP) Libraries." Restoring data files is discussed in a following section "Merging Saved Files."

B.1 Shut Down the System

You should be the only person logged in to your computer when you perform the upgrade. 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.

Refer to **shutdown**(C) in the XENIX *Reference* for information on bringing the system down to Single User (or System Maintenance) Mode.

B.2 Backup the System

Before you do anything else, make a full backup of the system. Use the **tar**(C) or **cpio**(C) commands. Refer to **tar**(C) and **cpio**(C) in the XENIX *User's Reference.* You cannot use any other filesystem backup utilities.

You should make a backup immediately before you upgrade 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.

B.3 Save Special Files

You can use any backup media, including floppy and tape drives, to save most of your files. Note that if you use a special driver with your backup device, you cannot use that device immediately after installing XENIX System V, since the new kernel is not yet linked with the required device driver. Therefore, for precautionary reasons, we recommend that you backup your files on floppies before you install XENIX System V.

Before you can use any special devices, you must link the new XENIX kernel. If you have source for the device drivers, save the source code on the default backup device, the floppy drive, then recompile the driver on the new system. Also, if you need special, customized libraries, or if any libraries are furnished with your device drivers, save these libraries with the device driver source.

This is important, since you can only read information from the default device, once you reinstall XENIX, until you link the kernel.

B.3.1 Choosing Which Files To Save

You should save any files that are customized or are in some way particular to your system. These files include:

- Applications data files and ar(CP) libraries.
- Any standard system files that are modified for your site.
- Any locally created shell scripts or programs.
- All user directories, subdirectories and files.

At this time, make sure you have the original distributions of any applications packages you use. You must reinstall those software packages after you upgrade the XENIX system.

Save any files you use with your applications, such as databases and control files, in ASCII or symbolic form. For example, if you use Lyrix, you should save all the files in the */usr/lib/wp* directory. Note that Lyrix data files are already in ASCII form, and you can save them directly onto your backup media.

You should check the following standard XENIX system files and directories for local changes made to the XENIX 3.0 standard distribution. Save those files which have local modifications.

XENIX 3.0 Files

File or Directory:	Examine for:
/.profile	root *sh* startup script.
/etc/profile	System wide *sh* startup script.
/etc/cshrc	System wide *csh* startup script.
‡/etc/default/*	For local changes.
‡/etc/group	Group database.
‡/etc/passwd	User database.
†/etc/rc	For any local additions.
‡/etc/systemid	The name of your system.
†/etc/ttys	Terminal line configuration data.
/etc/ttytype	Terminal line to terminal type mapping data.
/etc/termcap	For any local entries.
/lib	For locally developed language processors.
/usr/bin/*	For local additions.
/usr/dict/words	For local additions to **words**.
/usr/include/*	For local additions.
/usr/lib/mail/*	For local **mail** routing information.
/usr/lib/crontab	**cron** daemon database.
/usr/lib/font/*	For locally developed font libraries.
/usr/lib/lint/*	For locally developed **lint** libraries.
/usr/lib/tabset/*	For locally developed tab setting files.
/usr/lib/term/*	For locally developed **nroff** driving tables.
/usr/lib/tmac/*	For locally developed **nroff/troff** macros.
/usr/lib/uucp/*	For local **uucp** configuration files and dialers.
‡/usr/news	For local news.
/usr/spool/*	For current **mail**, **uucp**, **at**, and other files.
‡/usr/[janed,johnd]	User home directories.

It is very important that the files marked with a dagger (†) be examined carefully before reinstalling on XENIX System V. Check the files for local additions or changes, but do not replace the System V versions with your XENIX 3.0 versions. There are changes to content, and sometimes format, of these files with System V. Using the 3.0 versions of these files can cause problems. This is the case for most of this list. Those directories marked with a double dagger (‡) can safely be copied from your 3.0 system. The last item in the above list refers to all individual user directories on the *root* file system. User directories may be in another location or */usr* may be mountable on your system.

B.3.2 Save the Actual Files

When using floppies, make sure they are error free and formatted. XENIX requires high quality media.

The **tar** command is a convenient way to save files. You can also use the **cpio** command. For example, to save the standard system files located in */etc*, enter:

```
# cd /
# tar cv ./etc/profile ./etc/group ./etc/rc ./etc/ttys ./etc/ttytype ./etc/termcap
```

Remember to use the "relative pathname" format with **tar**. For example, specify *./etc/profile*, not */etc/profile*, where the dot (.) indicates the pathname is relative to your current directory.

Make sure you label any floppies or tapes you use with their exact contents, the command used to create the backup, the date and time, and if they are part of a multi-volume set. If you use more than one volume (floppy or tape), use the **k** option with **tar**.

These commands save some user directories as well as other directories in */usr*:

```
# cd /
# tar cv ./usr/johnd ./usr/janed ./usr/spool ./usr/news
```

These commands save all the Lyrix custom system files in */usr/lib/wp*:

```
# cd /
# tar cv ./usr/lib/wp
```

For more information on **tar**, see the XENIX *User's Reference*.

B.3.3 Save Data Files and ar(CP) Libraries

If you are using XENIX-86, you must follow these guidelines for saving data files and **ar** libraries. These steps are not necessary under XENIX-286, however they are a good precaution.

Data Files

Applications, such as Multiplan, Informix and Level II COBOL use data files, which you must save in ASCII or symbolic form.

With Multiplan, use the **TRANSFER OPTION** command to save data files in symbolic (SYLK) format. Then transfer the SYLK files to your backup media.

See the "Command Directory" in the Multiplan *User's Guide* for more information on **TRANSFER OPTION**. See also Appendix 4 "The SYLK (Symbolic Link) File Format" in the Multiplan *User's Guide*.

You can save Informix data files using the Informix **UNLOAD** command. Also, with **UNLOAD** you can transfer a file directly to your backup media. For information on **UNLOAD**, see your Informix manual under "**dbstatus.**"

All Level II COBOL programs you create are completely portable, so you can save them using **tar** or **cpio** without any special handling. However, C-ISAM® files created under XENIX-86 3.0 cannot be transferred to XENIX-86 System V. You must recreate any such files on the new system.

Extracting and merging your data files is described in section B.7, under "Merging Data Files" and "ar(CP) Libraries", in these *Release Notes*.

ar(CP) Libraries

To save **ar** archive files, you:

- Extract the files that make up the archive from the archive file.

- Save the extracted files on your backup media.

For example:

```
# cd /usr/lib
# ar xv libisam.a
```

When the extract command is finished, you save the resulting files on your backup media:

```
# tar cv `ar t libisam.a`
```

Recreating archives on the new system is discussed section B.7, under "Merging Data Files" and "**ar**(CP) Libraries", in these *Release Notes*.

B.4 Install XENIX System V

After you have backed up the system and saved all special files that you need for your system, install XENIX System V. Follow the instructions in the XENIX *Installation Guide* in the binder marked "XENIX Operating System *Run Time Environment*" to install the XENIX Operating System. furnished with System V.

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.

Also, you must recompile any device drivers under SCO XENIX System V. If you are going to recompile any device drivers, be sure to install the Development System. If you do not have source code for the device drivers you use, you should obtain new binaries which run under SCO XENIX System V from the manufacturer or vendor of the device.

When you are finished installing, make sure you are in Single User, or System Maintenance Mode. Then return to the next section to merge the special files you saved with your new operating system.

B.5 Relink the Kernel

If you have special device drivers, you must:

- Recompile the software under XENIX System V, or obtain a replacement.
- Link the driver into the XENIX kernel with the link kit.

Any device drivers compiled under SCO XENIX 3.0 must be recompiled under System V. You may have to obtain an SCO XENIX System V version of any drivers from the driver or device manufacturer if you do not have source code.

Make sure you installed the link kit software. If you have not done so, see **custom**(C) for information on extracting packages from XENIX System V distributions.

B.6 Reinstall Applications

All SCO XENIX 3.0 applications run on SCO XENIX System V. Therefore, you should use the original distribution floppies to reinstall any applications.

See the appropriate release notes and installation manuals for information on installing any applications programs. Install any such programs as if this is the first time.

Do not copy or save the applications from the old system, unless the original distributions are lost or damaged, and you cannot replace them. It is safer to reinstall applications software packages.

B.7 Merge Saved Files

Once XENIX and any applications are reinstalled, you can merge in new files. This section describes the procedure for merging standard system files, data files, and **ar** libraries.

Note

> Remember that all commands given in this section are only examples. The exact procedure depends upon how and in what order you saved your files and directories.

For standard system files, except for */usr/spool* and */usr/[janed,johnd]* files, prepare a temporary directory and extract the files from a backup volume into this directory:

```
# mkdir /usr/convert
# cd /usr/convert
# tar x
```

Move the extracted files to their correct locations , for example if you have extracted files which belong in */etc*:

```
# mv ./etc/* /etc
```

Repeat the **tar** extraction for each volume you made.

The spooling directory and the users' home directories are very simple to extract directly into the correct directories. For example, with the media containing */usr/[johnd, janed]* and */usr/spool* mounted in the default device, enter:

```
# cd /usr
# tar xv
```

Special Lyrix files can be extracted in place from your backup media. For example, with the backup volume containing */usr/lib/wp* mounted in the default device:

```
# cd /usr/lib/wp
# tar xv
```

Other files, however, must be merged into the distributed versions by hand, for example, with the following files already extracted into the *convert* directory:

```
# cd /usr/convert/etc; cp passwd group ttytype /etc
# cd /usr/convert/usr/lib
# cp crontab /usr/spool/cron/crontabs/root
```

Be careful when merging files, especially complex data files like */etc/rc*, */etc/termcap*, or */etc/ttys*. Use the **diff**(C) program to note the differences between newly installed files, and those you save from your previous system. Add the differences, if they are necessary, to the new System V files. Do not directly overwrite the System V versions with your 3.0 version files.

After all the files have been examined and local site information is merged into the System V files, remember to clean up:

```
# cd / ; rm -fr /usr/convert
```

Merge Data Files

For Multiplan data files that are saved in symbolic form, extract them into the appropriate directories from the backup media. Then use the Multiplan command **TRANSFER OPTIONS** to translate the file from symbolic form.

With Informix, extract the data files into the appropriate directories. Use the **LOAD** command, described in the "**dbstatus**" section of the Informix manual.

ar(CP) Libraries

To merge archive and library files, create a scratch directory, or work in */tmp*. Mount the backup volume containing the appropriate libraries. Remember, you saved them individually.

Extract the files, for example:

```
# cd /tmp
# tar xv
```

Rearchive the extracted files, for instance with our *libisam.a* example:

```
# ar ruv libisam.a *.o
```

When the archive is created, run the **ranlib** program:

```
# ranlib libisam.a
```

Finally, move the new archive back to the appropriate directory and remove any unneeded files:

```
# mv libisam.a /usr/lib
# rm *.o
```

Your upgrade is now complete.

Name

mkdev - Calls scripts to create devices

Syntax

/etc/mkdev lp
/etc/mkdev hd
/etc/mkdev serial
/etc/mkdev fs [*device file*]

Description

mkdev calls the scripts to create the requested type of device file(s). *mkdev* calls either *lpinit*(C), *hdinit*, *serinit*, or *fsinit*. If no arguments are listed, *mkdev* prints a usage message.

/etc/mkdev lp creates device files for use with line printers.

/etc/mkdev hd creates device files for use with peripheral hard disks. The device files for an internal hard disk already exist.

/etc/mkdev serial creates device files for use with serial cards. The device files for the first and second ports already exist. Additional device files must be created for the ports added when expansion cards are added to the system.

/etc/mkdev fs performs the system maintenance tasks required to add a new filesystem to the system once the device is created (*mknod*(C)) and the filesystem is made (*mkfs*(C)). It creates the **/file** and **/file/lost&found** directories, reserves slots in the **lost&found** directory, and modifies **/etc/checklist** and **/etc/rc** to check (*fsck*(C)) and mount (*mount*(C)) the filesystem as appropriate. It is usually used in conjunction with *mkdev hd* when adding a second hard disk to the system, but can be used on any additional filesystem (for example, on a large internal hard disk or a floppy.)

The various *init* scripts prompt for the information necessary to create the devices.

Files

/etc/hdinit
/etc/lpinit
/etc/serinit
/etc/fsinit

 MKDEV (C)

See Also

lpinit(C), hd(M), lp(M), serial(M)

Name

brkctl – Allocates data in a far segment.

Syntax

char far *brkctl(*command*, *increment*, *ptr*)
int *command*;
long *increment*;
char far *ptr;

Description

The *brkctl* system call allocates and deallocated memory in additional data segments in small and middle model programs. In order for the C compiler to make use of the far pointer return values in small and middle model programs, *brkctl* must be declared to return a far pointer. To enable the 'far' keyword for small model C programs, the compile **−Me** option must be used. Middle model C programs require the **−Mme** option.

The *brkctl* system call should be used only for dynamically allocating additional segments in small and middle model programs. All other uses should be avoided in favor of *sbrk*(S), *malloc*(S), and other standard UNIX system services.

In the syntax example (found in the preceding section):

command is either **BR_ARGSEG**, **BR_NEWSEG**, or **BR_IMPSEG**.

increment is a signed long increment.

'char far *ptr' is only used when *command* is **BR_ARGSEG**.

The second argument (*increment*) may not be greater than 64K bytes. The *increment* may not be more negative than the total amount of memory in all far segments plus the amount allocated in the near segment after process creation.

If the *command* is **BR_IMPSEG**, and a negative *increment* causes one or more segments to be freed, the 'segment in question' (see the *Return Values* section) is the last remaining segment that was not freed. **BR_IMPSEG** implies the use of the last data segment. Unless the process is small or middle model and currently has only one data segment, a positive *increment* that would overflow the last data segment causes a new segment to be allocated.

If the *command* is **BR_ARGSEG**, the *increment* may not be more negative than the size of the segment. The third argument (*ptr*), is assumed to be a far pointer in all models; the offset portion is never used.

If the *command* is **BR_NEWSEG**, the *increment* may not be negative at all. Any memory allocated is guaranteed to be at the base of a new segment.

Return Value

Brkctl() almost always returns a far pointer to the base of the affected region, (char far *)-1 on error.

When the *increment* is greater than 0, the return value is a pointer to the base of the newly allocated memory.

When the *increment* is less than or equal to 0, the return value is a pointer to the first illegal byte in the segment in question (usually the base of the deallocated memory). If that segment is full (exactly 64K bytes), the return value will be a pointer to the base of the next segment (which may or may not exist).

Command	Increment	Ptr	Action
BR_ARGSEG	0	<valid far ptr>	report on segment
BR_ARGSEG	other	<valid far ptr>	increment specified segment
BR_NEWSEG	0	-	allocate new segment, size = 0
BR_NEWSEG	other	-	allocate new segment, size = increment
BR_IMPSEG	0	-	report on last segment; may free up empty segment(s).
BR_IMPSEG	other	-	increment last segment; on large model (or small and middle model with mutiple data segments) may allocate new segment.

See Also

cc(CP), ld(CP), machine(M), malloc(S), sbrk(S)

Notes

Use of the type of low level control possible when using *brkctl* is not advised. The functionality of *brkctl* may change in future releases. It is best to use the *sbrk*, *brk*, and *malloc* calls rather than *brkctl*.

Brkctl is currently available only on protected mode XENIX.

In all models, the 'near' data segment must be the first data segment.

Brkctl calls with **BR_IMPSEG** and a negative *increment* that would affect a shared data segment are refused.

Name

shV – Invokes the shell command interpreter.

Syntax

shV [**–ceiknrstuvx**] [args]

Description

The shell is the standard command programming language that executes commands read from a terminal or a file. See *Invocation* below for the meaning of arguments to the shell.

Commands

A *simple–command* is a sequence of nonblank *words* separated by *blanks* (a *blank* is a tab or a space). The first word specifies the name of the command to be executed. Except as specified below, the remaining words are passed as arguments to the invoked command. The command name is passed as argument 0 (see *exec*(S)). The *value* of a simple–command is its exit status if it terminates normally, or (octal) 1000+*status* if it terminates abnormally (i.e., if the failure produces a core file). See *signal*(S) for a list of status values.

A *pipeline* is a sequence of one or more *commands* separated by a vertical bar (|). (The caret (^), also has the same effect.) The standard output of each command but the last is connected by a *pipe*(S) to the standard input of the next command. Each command is run as a separate process; the shell waits for the last command to terminate.

A *list* is a sequence of one or more pipelines separated by ;, **&**, **&&**, or ||, and optionally terminated by ; or **&**. Of these four symbols, ; and **&** have equal precedence, which is lower than that of **&&** and ||. The symbols **&&** and || also have equal precedence. A semicolon (;) causes sequential execution of the preceding pipeline; an ampersand (&) causes asynchronous execution of the preceding pipeline (i.e., the shell does *not* wait for that pipeline to finish). The symbol **&&** (||) causes the *list* following it to be executed only if the preceding pipeline returns a zero (nonzero) exit status. An arbitrary number of newlines may appear in a *list*, instead of semicolons, to delimit commands.

A *command* is either a simple-command or one of the following commands. Unless otherwise stated, the value returned by a command is that of the last simple-command executed in the command:

for *name* [**in** *word* . . .] **do** *list* **done**
Each time a **for** command is executed, *name* is set to the next *word* taken from the **in** *word* list. If **in** *word* is omitted, then the **for** command executes the **do** *list* once for each positional parameter that is set (see *Parameter Substitution* below). Execution ends when there are no more words in the list.

case *word* **in** [*pattern* [| *pattern*] . . .) *list* **;;**] . . . **esac**
A **case** command executes the *list* associated with the first *pattern* that matches *word*. The form of the patterns is the same as that used for filename generation (see *Filename Generation* below).

if *list* **then** *list* [**elif** *list* **then** *list*] . . . [**else** *list*] **fi**
The *list* following **if** is executed and, if it returns a zero exit status, the *list* following the first **then** is executed. Otherwise, the *list* following **elif** is executed and, if its value is zero, the *list* following the next **then** is executed. Failing that, the **else** *list* is executed. If no **else** *list* or **then** *list* is executed, then the **if** command returns a zero exit status.

while *list* **do** *list* **done**
A **while** command repeatedly executes the **while** *list* and, if the exit status of the last command in the list is zero, executes the **do** *list*; otherwise the loop terminates. If no commands in the **do** *list* are executed, then the **while** command returns a zero exit status; **until** may be used in place of **while** to negate the loop termination test.

(*list*)
Executes *list* in a subshell.

{*list*;}
list is simply executed.

name () {*list*;}
Define a function which is referenced by *name*. The body of functions is the *list* of commands between { and }. Execution of functions is described below (see *Execution*.)

The following words are recognized only as the first word of a command and when not quoted:

if then else elif fi case esac for while until do done { }

Comments

A word beginning with # causes that word and all the following characters up to a newline to be ignored.

Command Substitution

The standard output from a command enclosed in a pair of grave accents (``) may be used as part or all of a word; trailing newlines are removed.

Parameter Substitution

The character $ is used to introduce substitutable *parameters*. Positional parameters may be assigned values by set. Variables may be set by writing:

name=*value* [*name*=*value*] . . .

Pattern-matching is not performed on *value*.

${*parameter*}
A *parameter* is a sequence of letters, digits, or underscores (a *name*), a digit, or any of the characters *, @, #, ?, −, $, and !. The value, if any, of the parameter is substituted. The braces are required only when *parameter* is followed by a letter, digit, or underscore that is not to be interpreted as part of its name. A *name* must begin with a letter or underscore. If *parameter* is a digit then it is a positional parameter. If *parameter* is * or @, then all the positional parameters, starting with **$1**, are substituted (separated by spaces). Parameter **$0** is set from argument zero when the shell is invoked.

${*parameter*:−*word*}
If *parameter* is set and is not a null argument, substitute its value; otherwise substitute *word*.

${*parameter*:=*word*}
If *parameter* is not set or is null, then set it to *word*; the value of the parameter is then substituted. Positional parameters may not be assigned to in this way.

${*parameter*:?*word*}
If *parameter* is set and is not a null argument, substitute its value; otherwise, print *word* and exit from the shell. If *word* is omitted, the message "parameter null or not set" is printed.

${*parameter*:+*word*}
If *parameter* is set and is not a null argument, substitute *word*; otherwise substitute nothing. In the above, *word* is not evaluated unless it is to be used as the substituted string, so that in the following example, **pwd** is executed only if **d** is not set or is null:
echo ${d:-`pwd`}

If the colon (:) is omitted from the above expressions, then the shell only checks whether *parameter* is set.

The following parameters are automatically set by the shell:

\# The number of positional parameters in decimal

\- Flags supplied to the shell on invocation or by the **set** command

? The decimal value returned by the last synchronously executed command

$ The process number of this shell

! The process number of the last background command invoked

The following parameters are used by the shell:

HOME
The default argument (home directory) for the *cd* command

PATH
The search path for commands (see *Execution* below)

MAIL
If this variable is set to the name of a mail file, then the shell informs the user of the arrival of mail in the specified file

MAILCHECK
This parameter specifies how often (in seconds) the shell will check for the arrival of mail in the files specified by the **MAILPATH** or **MAIL** parameters. The default value is 600 seconds (10 minutes). If set to 0, the shell will check before each prompt.

MAILPATH
A colon (:) separated list of file names. If this parameter is set, the shell informs the user of the arrival of mail in any of the specified files. Each file name can be followed by % and a message that will be printed when the modification time changes. The default message is *you have mail.*

PS1
Primary prompt string, by default "$ "

PS2
Secondary prompt string, by default "> "

IFS
Internal field separators, normally **space**, **tab**, and **newline**

SHACCT
If this parameter is set to the name of a file writable by the user. the shell will write an accounting record in the file for each shell procedure executed. Accounting routines such as *acctcom*(C) and *accton*(C) can be used to anaylze the data collected.

SHELL
When the shell is invoked, it scans the environment (see *Environment* below) for this name. If it is found and there is an 'r' in the file name part of its value, the shell becomes a restricted shell.

The shell gives default values to **PATH**, **PS1**, **PS2**, and **IFS**, while **HOME** and **MAIL** are not set at all by the shell (although **HOME** *is* set by *login* (M)).

Blank Interpretation

After parameter and command substitution, the results of substitution are scanned for internal field separator characters (those found in **IFS**) and split into distinct arguments where such characters are found. Explicit null arguments ("" or '') are retained. Implicit null arguments (those resulting from *parameters* that have no values) are removed.

Filename Generation

Following substitution, each command *word* is scanned for the characters *, ?, and [. If one of these characters appears, the word is regarded as a *pattern*. The word is replaced with alphabetically sorted filenames that match the pattern. If no filename is found that matches the pattern, the word is left unchanged. The character . at the start of a filename or immediately following a /, as well as the character / itself, must be matched explicitly. These characters and their matching patterns are:

* Matches any string, including the null string.

? Matches any single character.

[. . .]
Matches any one of the enclosed characters. A pair of characters separated by - matches any character lexically between the pair, inclusive. If the first character following the opening bracket ([) is an exclamation mark (!), then any character not enclosed is matched.

Quoting

The following characters have a special meaning to the shell and cause termination of a word unless quoted:

; & ( ) | ^ < > newline space tab

A character may be *quoted* (i.e., made to stand for itself) by preceding it with a \. The pair **\newline** is ignored. All characters enclosed between a pair of single quotation marks (´´), except a single quotation mark, are quoted. Inside double quotation marks (""), parameter and command substitution occurs and \ quotes the characters \, `, ", and $. **"$*"** is equivalent to **"$1 $2 . . ."**, whereas **"$@"** is equivalent to **"$1" "$2"** . . .

Prompting

When used interactively, the shell prompts with the value of **PS1** before reading a command. If at any time a newline is typed and further input is needed to complete a command, the secondary prompt (i.e., the value of **PS2**) is issued.

Spelling Checker

When using *cd*(C) the shell checks spelling. For example, if you change to a different directory using *cd* and misspell the directory name, the shell repsonds with an alternative spelling of an existing directory. Enter "y" and press RETURN to change to the offered directory, or retype the command line if the offered spelling is incorrect. In this example the **shV**(C) response is **bold**faced.

$ cd /usr/spol/uucp
cd /usr/spool/uucp?y
ok

Input/Output

Before a command is executed, its input and output may be redirected using a special notation interpreted by the shell. The following may appear anywhere in a simple-command or may precede or follow a *command*. They are *not* passed on to the invoked command; substitution occurs before *word* or *digit* is used:

<word	Use file *word* as standard input (file descriptor 0).
>word	Use file *word* as standard output (file descriptor 1). If the file does not exist, it is created; otherwise, it is truncated to zero length.
>>word	Use file *word* as standard output. If the file exists, output is appended to it (by first seeking the end-of-file); otherwise, the file is created.
<<[-]word	The shell input is read up to a line that is the same as *word*, or to an end-of-file. The resulting document becomes the standard input. If any character of *word* is quoted, no interpretation is placed upon the characters of the document; otherwise, parameter and command substitution occurs, (unescaped) **\newline** is ignored, and \ must be used to quote the characters \, **$**, `, and the first character of *word*. If **-** is appended to <<, all leading tabs are stripped from *word* and from the document.
<&digit	The standard input is duplicated from file descriptor *digit* (see *dup*(S)). Similarly for the standard output using >.
<&-	The standard input is closed. Similarly for the standard output using >.

If one of the above is preceded by a digit, the file descriptor created is that specified by the digit (instead of the default 0 or 1). For example:

... 2>&1

creates file descriptor 2 that is a duplicate of file descriptor 1.

If a command is followed by **&** , the default standard input for the command is the empty file **/dev/null**. Otherwise, the environment for the execution of a command contains the file descriptors of the invoking shell as modified by input/output specifications.

Environment

The *environment* (see *environ*(M)) is a list of name-value pairs that is passed to an executed program in the same way as a normal argument list. The shell interacts with the environment in several ways. On invocation, the shell scans the environment and creates a parameter for each name found, giving it the corresponding value. Executed commands inherit the same environment. If the user modifies the values of these parameters or creates new ones, none of these affect the environment unless the **export** command is used to bind the shell's parameter to the environment. The environment seen by any executed command is composed of any unmodified name–value pairs originally inherited by the shell, minus any pairs removed by **unset**, plus any modifications or additions, all of which must be noted in **export** commands.

The environment for any *simple–command* may be augmented by prefixing it with one or more assignments to parameters. Thus:

TERM=450 cmd args

and

(export TERM; TERM=450; cmd args)

are equivalent (as far as the above execution of *cmd* is concerned).

If the **–k** flag is set, *all* keyword arguments are placed in the environment, even if they occur after the command name.

Signals

The INTERRUPT and QUIT signals for an invoked command are ignored if the command is followed by &; otherwise signals have the values inherited by the shell from its parent, with the exception of signal 11. See the **trap** command below.

Execution

Each time a command is executed, the above substitutions are carried out. Except for the *Special Commands* listed below, a new process is created and an attempt is made to execute the command via *exec*(S).

The shell parameter **PATH** defines the search path for the directory containing the command. Alternative directory names are separated by a colon (:). The default path is **:/bin:/usr/bin** (specifying the current directory, **/bin**, and **/usr/bin**, in that order). Note that the current directory is specified by a null pathname, which can appear immediately after the equal sign or between the colon delimiters anywhere else in the path list. If the command name contains a **/**, then the search path is not used. Otherwise, each directory in the path is searched for an executable file. If the file has execute permission but is not an **a.out** file, it is assumed to be a file containing shell commands. A subshell (i.e., a separate process) is spawned to read it. A parenthesized command is also executed in a subshell.

Shell procedures are often used by users running the **csh** . However, if the first character of the procedure is a # (comment character), the **csh** assumes the procedure is a **csh** script, and invokes */bin/csh* to execute it. Always start **sh** procedures with some other character if **csh** users are to run the procedure at any time. This invokes the standard shell */bin/sh* .

The location in the search path where a command was found is remembered by the shell (to help avoid unnecessary *execs* later). If the command was found in a relative directory, its location must be re-determined whenever the current directory changes. The shell forgets all remembered locations whenever the **PATH** variable is changed or the **hash** **-r** command is executed (see below).

Special Commands

Input/output redirection is now permitted for these commands:

: No effect; the command does nothing. A zero exit code is returned.

. *file*
: Reads and executes commands from *file* and returns. The search path specified by **PATH** is used to find the directory containing *file*.

break [*n*]
: Exits from the enclosing **for** or **while** loop, if any. If *n* is specified, it breaks *n* levels.

continue [*n*]
: Resumes the next iteration of the enclosing **for** or **while** loop. If *n* is specified, it resumes at the *n*-th enclosing loop.

cd [*arg*]
: Changes the current directory to *arg*. The shell parameter **HOME** is the default *arg*. The shell parameter **CDPATH** defines the search path for the directory containing *arg*. Alternative directory names are separated by a colon (:). The default path is <null> (specifying the current directory). Note that the current directory is specified by a null path name, which can appear immediately after the equal sign or between the colon delimiters anywhere else in the path list. If *arg* begins with a /, the search path is not used. Otherwise, each directory in the path is searched for *arg*.

If the shell is reading its commands from a terminal, and the specified directory does not exist (or some component cannot be searched), spelling correction is applied to each component of *directory*, in a search for the "correct" name. The shell then asks whether or not to try and change directory to the corrected directory name; an answer of *n* means "no", and anything else is taken as "yes".

eval [*arg* ...]
The arguments are read as input to the shell and the resulting command(s) executed.

exec [*arg* ...]
The command specified by the arguments is executed in place of this shell without creating a new process. Input/output arguments may appear and, if no other arguments are given, cause the shell input/output to be modified.

exit [*n*]
Causes a shell to exit with the exit status specified by *n*. If *n* is omitted, the exit status is that of the last command executed. An end-of-file will also cause the shell to exit.

export [*name* ...]
The given *names* are marked for automatic export to the *environment* of subsequently executed commands. If no arguments are given, a list of all names that are exported in this shell is printed.

hash [**−r**] [*name* ...]
For each *name*, the location in the search path of the command specified by *name* is determined and remembered by the shell. The **−r** option causes the shell to forget all remembered locations. If no arguments are given, information about remembered commands is presented. *Hits* is the number of times a command has been invoked by the shell process. *Cost* is a measure of the work required to locate a command in the search path. There are certain situations which require that the stored location of a command be recalculated. Commands for which this will be done are indicated by an asterisk (*) adjacent to the *hits* information. *Cost* will be incremented when the recalculation is done.

newgrp [*arg* ...]
Equivalent to **exec newgrp** *arg* ...

pwd
Print the current working directory. See *pwd*(C) for usage and description.

read [*name* . . .]
One line is read from the standard input and the first word is assigned to the first *name*, the second word to the second *name*, etc., with leftover words assigned to the last *name*. The return code is 0 unless an end-of-file is encountered.

readonly [*name* . . .]
The given *names* are marked *readonly* and the values of the these *names* may not be changed by subsequent assignment. If no arguments are given, a list of all *readonly* names is printed.

return [*n*]
Causes a function to exit with the return value specified by *n*. If *n* is omitted, the return status is that of the last command executed.

set [**−eknuvx** [*arg* . . .]]

−e
If the shell is noninteractive, exits immediately if a command exits with a nonzero exit status.

−f Disables file name generation.

−h
Locates and remembers fuction commands as function are defined (function commands are normally located when the function is executed).

−k
Places all keyword arguments in the environment for a command, not just those that precede the command name.

−n
Reads commands but does not execute them.

−u
Treats unset variables as an error when substituting.

−v
Prints shell input lines as they are read.

−x
Prints commands and their arguments as they are executed. Although this flag is passed to subshells, it does not enable tracing in those subshells.

−−
Does not change any of the flags; useful in setting $1 to −.

Using **+** rather than **−** causes these flags to be turned off. These flags can also be used upon invocation of the shell. The current set of flags may be found in **$−**. The remaining arguments are positional parameters and are assigned, in order, to **$1**, **$2**, ... If no arguments are given, the values of all names are printed.

shift
The positional parameters from **$2** ... are renamed **$1** ...

test
Evaluates conditional expressions. See *test*(C) for usage and description.

times
Prints the accumulated user and system times for processes run from the shell.

trap [*arg*] [*n*] ...
arg is a command to be read and executed when the shell receives signal(s) *n*. (Note that *arg* is scanned once when the trap is set and once when the trap is taken.) Trap commands are executed in order of signal number. The highest signal number allowed is 16. Any attempt to set a trap on a signal that was ignored on entry to the current shell is ineffective. An attempt to trap on signal 11 (memory fault) produces an error. If *arg* is absent, all trap(s) *n* are reset to their original values. If *arg* is the null string, this signal is ignored by the shell and by the commands it invokes. If *n* is 0, the command *arg* is executed on exit from the shell. The **trap** command with no arguments prints a list of commands associated with each signal number.

type [*name* ...]
For each *name*, indicate how it would be interpreted if used as a command name.

ulimit [[−f] *n*]
imposes a size limit of *n* blocks on files.

−f imposes a size limit of *n* blocks on files written by child processes (files of any size may be read). Any user may decrease the file size limit, but only the super−user (root) can increase the limit. With no argument, the current limit is printed.

If no option is given and a number is specified, **−f** is assumed.

unset [*name* ...]
For each *name*, remove the corresponding variable or function. Teh variables **PATH**, **PS1**, **PS2**, **MAILCHECK** and **IFS** cannot be unset.

umask [*ooo*]
The user file-creation mask is set to the octal number *ooo* where *o* is an octal digit (see *umask*(C)). If *ooo* is omitted, the current value of the mask is printed.

wait [*n*]
Waits for the specified process to terminate, and reports the termination status. If *n* is not given, all currently active child processes are waited for. The return code from this command is always 0.

Invocation

If the shell is invoked through *exec*(S) and the first character of argument 0 is **−**, commands are initially read from **/etc/profile** and then from **$HOME/.profile**, if such files exist. Thereafter, commands are read as described below, which is also the case when the shell is invoked as **/bin/sh**. The flags below are interpreted by the shell on invocation only; note that unless the **−c** or **−s** flag is specified, the first argument is assumed to be the name of a file containing commands, and the remaining arguments are passed as positional parameters to that command file:

−c *string* If the **−c** flag is present, commands are read from *string*.

−s If the **−s** flag is present or if no arguments remain, commands are read from the standard input. Any remaining arguments specify the positional parameters. Shell output is written to file descriptor 2.

−t If the **−t** flag is present, a single command is read and executed, and the shell exits. This flag is intended for use by C programs only and is not useful interactively.

−i If the **−i** flag is present or if the shell input and output are attached to a terminal, this shell is *interactive*. In this case, TERMINATE is ignored (so that **kill 0** does not kill an interactive shell) and INTERRUPT is caught and ignored (so that **wait** is interruptible). In all cases, QUIT is ignored by the shell.

−r If the **−r** flag is present, the shell is a restricted shell (see *rsh*(C)).

The remaining flags and arguments are described under the **set** command above.

Exit Status

Errors detected by the shell, such as syntax errors, cause the shell to return a nonzero exit status. If the shell is being used noninteractively, execution of the shell file is abandoned. Otherwise, the shell returns the exit status of the last command executed. See the **exit** command above.

Files

/etc/profile
$HOME/.profile
/tmp/sh*
/dev/null

See Also

cd(C), env(C), login(M), newgrp(C), rsh(C), test(C), umask(C), dup(S), exec(S), fork(S), pipe(S), signal(S), umask(S), wait(S), a.out(F), profile(M), environ(M)

Notes

The command **readonly** (without arguments) produces the same output as the command **export**.

If << is used to provide standard input to an .synchronous process invoked by **&**, the shell gets mixed up about naming the input document; a garbage file **/tmp/sh*** is created and the shell complains about not being able to find that file by another name.

If a command is executed, and a command with the same name is installed in a directory in the search path before the directory where the original command was found, the shell will continue to *exec* the original command. Use the **hash** command to correct this situation.

If you move the current directory or one above it, **pwd** may not give the correct response. Use the **cd** command with a full path name to correct this situation.

N

ROADMAP

S

A Guide to XENIX Documentation

THE • SANTA • CRUZ • OPERATION

This document was typeset with an IMAGEN® 8/300 Laser Printer.

Document Number: G-8-26-85-2.0/1.0

CONTENTS

INTRODUCTION

The documentation for a large and comprehensive operating system can sometimes take on a complexity of its own. The XENIX documentation is a complete set of over 3000 pages spanning information on hundreds of utilities, procedures, and routines. This can be a valuable resource, and the *Roadmap* is guide to using it as effectively as possible.

The *Roadmap* is a collection of introductory and reference material that helps you locate documentation on a command, concept, or other area of XENIX. Its purpose is both to orient the user by "mapping" the organization of the guides, and to act as a shortcut to finding information.

The first section orients the user by describing the kinds of things each guide covers. Other sections point the user to the guide appropriate to what is being sought. The user can search for information by command, or by the type of information being sought (*getting started, file editing, status information*, and so on).

USING THIS GUIDE

ORGANIZATION

The *Roadmap* has five parts:

Questions and Answers

This is a group of questions commonly asked about XENIX and its documentation.

Contents Summary

This is a "map" of what each XENIX guide covers. Major chapters are listed and contents are summarized for quick reference.

Cross Reference

This section lists the kinds of everyday tasks you can do with XENIX and in which guide or guides they can be found. This is useful when you want to do something specific, but aren't sure where to look.

Command Page Index

This section is an index to XENIX command or *man* pages. You look up the command name to find the page and its location.

Command/Topic Index

This section is also an index to commands. You look up a topic such as *Pattern Matching* and find the *man* page.

CONVENTIONS

Several conventions have been used in this guide. For example, the names of guides are abbreviated in text:

Release Notes	RN
Installation Guide	IG
Roadmap	RM
Introduction to XENIX	IX
User's Guide	UG
Operations Guide	OG
Text Processing Guide	TP
User's Reference	UR

When a section of a guide is noted, it follows the abbreviation for the name of the guide. For example, the *User's Reference*, which describes many XENIX commands, is divided into three sections, "C," "M," and "F."

The "C" section of the *User's Reference* is expressed as:

UR(C)

Commands are always shown with the section in which they are found. For example, the *vi* command is expressed as:

UR vi(C)

Note

The *Roadmap* is a guide to the *user* documentation listed above. However, the Command Page and Command/Topic Indexes also index the *Programmer's Reference*:

Programmer's Reference	PR (CP and S sections)

As the *Roadmap* is a guide to the basic XENIX user documentation, the above programming guides are not otherwise indexed here.

The *Roadmap* is intended as both an orientation to the XENIX documentation, and as a reference guide in itself.

YOU CAN USE DIFFERENT SECTIONS TO FIND OUT . . .	. . . BY LOOKING IN
How the guides are organized and what each covers	Contents Summary (p. 5)
Where to look to learn how to do a specific task	Cross Reference (p. 31) Command/Topic Index (p. 38)
Where to look to find a certain command	Command Page Index (p. 56)

Q A

Here are some questions commonly asked about the XENIX documentation, with responses.

Q: When should I use the command documentation (**man** pages) as opposed to the other documentation?

A: *The* **man** *pages are intended for reference and document commands only. Once you learn the format of* **man** *pages, they are a quick way to learn how a command works.*

The rest of the documentation explains not only commands, but includes more detailed information and examples.

Q: How do I know which group of **man** pages documents the commands I'm looking for?

A: *In an effort to make the large amount of XENIX command documentation more accessable,* **man** *pages have been split across three guides: The User's Reference contains user-oriented commands, The Programmer's Reference contains commands for programming, and The Text Processing Guide documents text processing commands.*

Q: I looked for documentation on **spell** in the *User's Reference* but couldn't find any. Where do I find out how to use this facility?

A: **spell** *is a text processing tool, so it is documented in the* **man** *page section (CT) of the Text Processing Guide.*

Q: What printers are supported for use with **nroff**?

A: *A list of* **nroff** compatible printers is included on the **nroff** man page in the CT section of the *Text Processing Guide*. See also TP term(CT), and UR term(F).

Q: Where can I find information on the **sed** and **awk** programs?

A: **sed** and **awk** are facilities that allow large scale, noninteractive of very large or multiple files. They are described in Appendix A of the *Text Processing Guide.*

CONTENTS SUMMARY

This section is a quick reference to major topics in the XENIX guides.

The primary chapters of each guide are listed in an easily readable format.

Name

Release Notes - Documents special features of a given XENIX release

Synopsis

- Notes on installation
- Compatibility of XENIX with various kinds of software and hardware
- Features of the specific release
- Notes on hardware and software
- Notes on the documentation
- Listings of the files contained in this release

Description

The Release Notes contain many useful notes, suggestions, and general information about the release of XENIX for your computer. This includes lists of the applications software that are compatible with this release of XENIX, as well as supported hardware such as computers, hard disks, and memory cards.

The Release Notes also include information about features that are new to the release of XENIX, and things that have changed from previous releases.

Finally, documentation errors are listed, plus a complete listing of the XENIX files included on your distribution media.

See Also

Installation Guide, Introduction to XENIX

Name

Installation Guide – Installing XENIX on your computer

Synopsis

- Step by step guide to installing XENIX
- Adding peripherals such as modems, hard disks, and printers
- Using both the XENIX and DOS operating systems

Description

The Installation Guide explains how to install XENIX on your computer. The procedure is largely automated so that all you have to do is insert the media (tape or disk) containing XENIX and answer the questions that appear onscreen. The procedure takes you through installation, and creation of the super–user (administration) account and first user accounts.

Partitioning hard disk to provide more storage space for files is also discussed. This includes setting up filesystems on the disk, and arranging it so that you can use the DOS operating system in addition to XENIX.

See Also

Operations Guide, UR(C): fdisk, badtrk, fsck, su, mkuser

INSTALLATION GUIDE

INSTALLATION PROCEDURE

- First user account
- System distribution
- Installation procedure
- Super-user password

USING DOS ON THE SAME DISK

- Partitioning with fdisk
- Installation on a DOS system
- DOS accessing utilities

Name

Introduction to XENIX - Introduces basic facilities and concepts of XENIX

Synopsis

- Demonstration of a sample XENIX session
- Introduction of basic XENIX concepts
- Performing typical tasks with XENIX

Description

The Introduction to XENIX begins by taking you through a sample XENIX session. This includes "logging in" (typing your name, password, and terminal type), how to correct simple typing mistakes, and how to give XENIX commands.

Basic concepts are then introduced, including: files and filesystems, commands, and how to direct the results of a command to a file.

Finally typical tasks for which XENIX is used are explained, including: manipulating files and directories, controlling processes, and getting status information.

All of this information is included elsewhere in greater depth, but this guide touches on many important concepts, and should be read first to get a quick idea of what you can do with XENIX.

See Also

Release Notes, Installation Guide, User's Guide, Operations Guide

INTRO TO XENIX

BASIC CONCEPTS

- Files and file systems
- Naming conventions
- Commands
- Input & output

DEMONSTRATION

- Logging in
- Logging out
- Typing commands
- Mistakes in typing
- Stopping a program

TASKS

- System access
- Process control
- Calculating
- Reminder service
- Terminal setup
- Processing information
- File security
- Status information
- Moving in the file system
- Command line editing
- Communicating with other users
- Lineprinting
- System clock and calendar

Name

User's Guide - Documents basic XENIX facilities

Synopsis

- Using the *vi* and *ed* text editors for file editing
- Using mail to send and receive messages
- Working in different command environments called *shells*
- Using *uucp* to communicate with remote systems

Description

The User's Guide explains how to use several key XENIX facilities, such as text editing and mail. You can begin using them immediately and they will become powerful tools as you learn more about them.

XENIX provides two programs for text editing: *vi* and *ed*. *vi* is an editor that displays a screenful of text and shows changes as they happen, hence the name *vi*, which stands for *visual*.

The *ed* editor is called a *line editor* because only the line you are editing is displayed. Though this may seem an inconvenience, you can do many things with *ed* that are not possible in *vi*.

Using *mail* you can send, receive, forward, and reply to messages. XENIX also includes a program called *uucp* that allows transfer of information to and from other computers.

XENIX uses a set of very flexible command interpreters called *shells*. The shell is a program that takes a user command (such as *mail* or *vi*) and passes it to the computer for action. XENIX offers several shells that are tailored to various kinds of users. All of the shells include powerful command languages that allow unusual control over how and when the commands are actually issued.

See Also

vi(UR-C), ed(UR-C), mail(UR-C), sh(UR-C), csh(UR-C), uucp(UR-C)

VI: A TEXT EDITOR

- Demonstration
- Solving problems
- Editing tasks
- Environment setup
- Commands summary

MAIL

- Advanced features
- Basic concepts
- Quick reference
- Demonstration
- Commands

THE SHELL

- Basic concepts
- Shell variables
- The shell state
- Invocation
- A command's environment
- Procedure examples
- Grammar
- Shell programming
- Passing arguments
- Redirection
- Special commands

BUILDING A COMMUNICATIONS SYSTEM

- Installation
- Maintenance
- Log inquiry
- System to system file copy
- System to system execution
- Security
- Command execution
- uucp spool directory cleanup

THE VISUAL SHELL

- Getting started
- Visual shell screen
- Reference

THE C-SHELL

- C-shell history
- Shell variables
- Invocation
- Expressions
- Redirection
- Aliases
- Special characters
- Built-in commands
- Command scripts

ED

- Demonstration
- Tasks
- Basic concepts
- Speeding up editing
- Summary of commands
- Context and regular expressions
- Editing scripts
- Cutting and pasting

Name

Operations Guide – Provides information for the system administrator

Synopsis

- Starting and stopping the system
- Using, maintaining, and backing up filesystems
- Adding new user accounts, passwords, and permissions
- Adding peripheral devices (modems, printers, and terminals)
- Solving system problems

Description

The Operations Guide contains various information for the system administrator, the person who keeps the system running smoothly. This may be you, or your computer may have a designated administrator who performs tasks such as adding user accounts or extra terminals.

This guide discusses how files are maintained. Topics include: file security and how to control access through permission levels, regulating file storage space, and how to back up and copy files. Special filesystems used by XENIX are also described.

Peripheral devices, such as modems, printers, and hard disks are also discussed here. Modems are used to let your computer communicate with remote systems over telephone lines. Information on how to set up a modem is also included.

Typical problems encountered by administrators are listed and procedures and solutions are presented. These include: freeing jammed printers, stopping runaway processes, restoring free storage space, and replacing a forgotten password.

See Also

UR(C): su, passwd, pwadmin, mkfs, mount, chmod, chown, df, du, tar, sysadmin, stty, micnet

INTRODUCTION

- The system manager
- The keyboard
- The Super-user account

STARTING AND STOPPING THE SYSTEM

- Starting the system
- Stopping the system
- Logging in as super-user

MAINTAINING FILE SYSTEMS

- Maintaining free space
- File system integrity

PREPARING FOR USERS

- Adding a user account
- Forcing a new password
- Changing a user's login group
- Changing a user's password
- Creating a group
- Changing a user ID
- Removing a user account

BACKING UP FILE SYSTEMS

- Strategies for backups
- Using the sysadmin program
- Using the tar command

USING FILE SYSTEMS

- File systems
- Permissions
- Managing file ownership
- System security
- Using accounting features

USING PERIPHERAL DEVICES

- Adding a terminal
- Removing a terminal
- Setting the terminal type
- Setting terminal lines
- Changing serial lines
- Setting serial line operation
- Adding a lineprinter

XENIX DIRECTORIES

• root	• /lib	• /bin
• /dev	• /mnt	• /etc
• /tmp	• Log files	• /usr

SOLVING SYSTEM PROBLEMS

- Restoring a nonechoing terminal
- Stopping a runaway process
- Solving lineprinter problems
- Removing hidden files
- Restoring free space
- Restoring lost system files
- Replacing a forgotten password
- Restoring an inoperable system
- Recovering from a system crash
- Changing initialization

BUILDING A MICNET NETWORK

- Planning a network
- Building a network
- Starting the network
- Testing a micnet network
- Using a uucp system

SPECIAL DEVICE FILES

- File system requirements
- Special filenames
- Terminal and network requirements
- Gap and block numbers
- Block sizes

Name

User's Reference - Reference to user level XENIX commands

Synopsis

- Lists commonly used XENIX commands in a standard XENIX format

Description

The User's Reference describes the XENIX commands most often used by the typical user. The format is standard for all commands, so once you learn it, reference is quick and efficient.

Each command (or *man*) page lists the name of the command, how it is used (its "syntax"), and an explanation of the syntax and usage. A section that directs you to other areas of the documentation is also standard. Finally, there is a section called "Notes" that gives special pointers about using the command.

See Also

Programmer's Reference

Name

Text Processing Guide - Introduces XENIX text processing tools

Synopsis

- Using text editing tools: *grep, diff, comm, sort, wc, cut, paste*
- Using text analysis tools: *spell, style, diction*
- Using text processing "shortcuts:"*macros*
- Formatting and typesetting text: *nroff/troff*
- Formatting tables: *tbl*
- Formatting mathematics: *eqn*
- Using large-scale editing tools: *awk, sed*

Description

The Text Processing Guide introduces a powerful set of text manipulation, formatting, and analysis tools. Using some or all of these facilities in combination, you can create surprisingly professional and visually pleasing documents.

This guide summarizes facilities, introduces general formatting concepts, provides sample projects, and explains how to organize your documents so they will be prepared for formatting later.

Orientation is towards real world applications, with many common document elements described, such as: font characteristics, line and character spacing, paragraphs, headings, lists, diagrams, and tables.

A highlight of this guide is a description of a unique set of text analysis tools. Using these, you can analyze grammatical content of a document, including how hard it is to read for the average person. One facility, called *diction* even checks for redundant or cluttered language.

See Also

vi(UR-C), ed(UR-C)

TEXT PROCESSING GUIDE

WRITING AND EDITING TOOLS

- Commands for text processing
- Using style and diction
- Using spell
- Writing tools

OVERVIEW

- Basic concepts
- A sample project
- Summary
- Formatting documents
- Managing writing projects

MM REFERENCE

- Invoking the macros
- Paragraphs and headings
- Displays
- References
- Memorandum and released paper styles
- Miscellaneous features
- Table of contents
- Lists
- Errors
- Formatting concepts
- Page headers and footers
- Footnotes
- Reserved names
- Summary of macros, strings, and number registers

USING THE MM MACROS

- Getting started with mm
- Using nroff/troff commands
- Basic formatting macros
- Checking mm input with mmcheck

USING NROFF/TROFF

- Inserting commands
- Tabs
- Point sizes and line spacing
- Fonts and special characters
- Strings
- Indents and line lengths
- Drawing lines and characters
- Titles, pages and numbering
- Number registers and arithmetic
- Diversions
- Macros with arguments
- Conditionals
- Macros
- Environments

NROFF/TROFF REFERENCE

- Basic formatting requests
- Processing control facilities
- Output and error messages
- Character translations, overstrike, and local motions
- Summary of escape sequences and number registers

FORMATTING TABLES

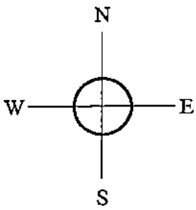
- Input format
- Invoking tbl
- Examples
- Command summary

EDITING WITH SED AND AWK

- Introduction
- Pattern matching with awk
- Editing with sed

FORMATTING MATHEMATICS

- Displayed equations
- Basic mathematic constructions
- Layout and design of mathematical text
- Definitions
- Invoking eqn
- Error messages
- In-line equations
- Complex mathematical constructions
- Keywords and precedence summary



CROSS REFERENCE

This section consolidates the XENIX documentation by grouping topics in categories, such as *Getting Started*, *Terminal Specifics*, and *Operations*.

Then it directs you to the appropriate guide and section number covering the topic you are looking for.

GETTING STARTED

• Release notes	
• Installation guide	IG 2
• System overview	IX 1
• System demonstration	IX 2
• Basic concepts	IX 3
• System tasks	IX 4

TERMINAL SPECIFICS

• Configuring a terminal	IG 4
• Setting terminal options	UR stty(C), OG 7
• Terminal capabilities	UR termcap(M)
• Terminal modes	UR tset(C)
• Restoring nonechoing terminal	OG 8.2
• Terminal identification	UG tty(C)
• Adding peripherals (printers, terminals)	IG 3
• Using peripheral devices	OG 7, IG 3

OPERATIONS

• The super user	OG 1,2
• Starting and stopping the system	OG 2
• Filesystems: maintenance and backups	OG 5,6
• XENIX Directories	OG B
• System security	OG 4,5
• Solving system problems	OG 8
• Maintaining free space	OG 5
• Adding user accounts	OG 3

STATUS

• How much of the disk is full?	UR du(C)
• How much of the disk is free?	UR df(C)
• Who is on the system?	UR who(C)
• What terminal is this?	UR tty(C)
• What processes are running?	UR ps(C)
• What day/time is it?	UR date(C)
• Information about a user	UR finger(C)

COMMUNICATIONS

• Using Mail	UG 3
• Between remote computers	UG 6, UR uucp(C)
• Building a communications network	OG 9, UR micnet(M)
• Writing from one terminal to another	IX 4, UR write(C)

FILES

• Conventions	IX 4
• Moving in the filesystem	IX 4, UR pwd, mv(C)
• Using file permissions	IX 4, UR chmod, l(C)
• Listing files	IX 4, UR ls, lc(C)
• Displaying file contents	IX 4, UR more, cat, head, tail(C)
• Deleting, creating a file	IX 4, UR rm(C)
• Moving, copying, naming files	IX 4, UR cp, mv(C)
• Finding files	IX 4, UR find(C)
• Editing files	UG 2, UR vi(C)
• Maintaining and backing up filesystems	OG 5,6 UR backup, restore, tar(C)

DIRECTORIES

• Printing your working directory	IX 4, UR pwd(C)
• Listing directory contents	IX 4, UR lc, ls(C)
• Creating, removing	IX 4, UR mkdir, rmdir, rm(C)
• Renaming, moving	IX 4, UR mv, copy, tar(C)
• Using directory permissions	IX 4, UR chmod, l(C)
• XENIX Directories	OG B
• XENIX Special Devices	OG A

THE SHELL

• Variables	UG 4
• Shell environment	UG 4
• Shell procedures/commands	UG 4
• C-shell	UG 7
• Bourne shell	UG 4
• Visual shell	UG 8
• Using expressions	UG 4
• Shell grammer	UG 4

EDITING TOOLS

• Text editors	UG 2, UR vi, ed(C)
• Formatting languages	TP 5,6 UR nroff, troff(C)
• Macro sets	TP 3,4
• Checking spelling	TP spell(CT)
• Checking style and diction	TP style, diction(CT)
• Creating tables	TP 7, TP tbl(CT)
• Formatting mathmatical equations	TP 8, TP eqn(CT)

BASIC COMMANDS

• Who am I?	UR who(C).
• Where am I?	UR where, pwd(C)
• What day/time is it?	UR date(C)
• Changing the account password	UR passwd(C)
• Listing Files/Directories	UR ls(C)
• Viewing a file	UR more, cat(C)
• Copying or moving a file	UR cp, mv(C)
• Removing files/directories	UR rm(C)
• Changing Directories	UR cd(C)
• Access to the Line printer	UR lpr(C)
• Print Working Directory	UR pwd(C)

COMMAND/TOPIC INDEX

This section lists common XENIX topics, showing on what page, section, and guide they are to be found.

Command/Topic Index

Commands, Systems Calls, Library Routines, and File Formats

COMMAND PAGE INDEX

This section lists all XENIX commands, showing on what page, section, and guide they are to be found.

Command Page Index

Commands, Systems Calls, Library Routines, and File Formats

XENIX® System V

Operating System

Installation Guide

This document was typeset with an IMAGEN® 8/300 Laser Printer.

SCO Document Number: XG-12-18-85-2.1/1.0

Contents

Chapter 1
Introduction

1.1 Overview

This is a guide to installing XENIX on your personal computer. Setting up user accounts and peripheral devices is also discussed.

Topics covered in this guide include:

- How the system is distributed in three packages and what each contains.
- Responding to system prompts in the installation procedure.
- What you need to get started.
- Starting the system for the first time.
- Preparing the hard disk for XENIX.
- Starting the system from the hard disk.
- Setting up user accounts.
- Using a second operating system (DOS) with XENIX.

1.2 Using This Guide

This guide contains the following sections:

Chapter 1: Introduction
: This chapter gives an introduction and overview of topics covered in this guide.

Chapter 2: Installation Procedure
: A step by step guide to installing XENIX on your computer. Starting XENIX for the first time, initializing the hard disk, and setting up user accounts are among the topics discussed.

Chapter 3: Using DOS and XENIX on the Same Disk
: How to use both the XENIX and DOS systems. Includes a description of the **fdisk**(C) utility used to partition the hard disk for concurrent use of both systems.

Chapter 2

Installation Procedure

2.1 Overview

This guide explains how to install the XENIX system on a personal computer. The XENIX Operating System is a powerful system of programs that gives your computer the same multi–user, multi–tasking capabilities as many large and expensive computers.

When you install the system, you:

- Initialize the hard disk, then;
- Copy the XENIX utilities from the XENIX distribution media to the initialized hard disk.

2.2 XENIX System Distribution

The complete XENIX System consists of the following three distribution sets:

- The Operating System
- The Development System
- The Text Processing System

The Operating System contains the XENIX programs you need to create multiple user accounts, manage file systems, create and manage files and perform system maintenance tasks.

The Development System contains the XENIX programs you need to create, compile, and debug assembly and high–level language programs.

The Text Processing System contains the XENIX programs you need to create, edit, and typeset documents.

2.3 How To Use This Guide

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, call the Support Center listed on the support information card included with the software.

Always follow these rules:

1. During the installation you use the keyboard to enter information. Always:

   - Begin each command after a prompt (often a $, % or # symbol). During the installation process, the prompt can be "<Installation>," under certain conditions.

   - Enter all requested names and numbers exactly as shown.

   - Complete a line by pressing the RETURN key.

     NOTE: The RETURN key is sometimes denoted on the keyboard by a "down-left" arrow, or referred to as the ENTER key.

2. If you make an entering error, you can erase the character:

   - By using the backspace key,

     OR

   - By pressing Ctrl-h.

   - To delete everything you have entered on a line, press Ctrl-u.

3. Some additional items to be aware of:

   - (y/n)? is asking "yes or no?", and always requires a response from you (type "y" or "n" and press the RETURN key).

   - The control symbol (^) refers to the Ctrl key, and is followed by another key which, used in combination with the Ctrl key, has a special meaning.

     Example: ^h means backspace

     Press the Ctrl key and the other key at the same time. This is the same action as using the Shift key.

   - The RETURN key on your keyboard may have a large single arrow pointing left (or the word ENTER) on it rather than the word "return." In this guide, however, it is written this way:

     RETURN

- Commands referred to in text are shown in **boldface** with the reference manual section next to the command in parentheses (for example **cat**(C)). Refer to the preface of the XENIX *User's Reference* for a guide to the various reference sections.

2.4 Before You Start

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:

- How to turn the computer off and on.
- How to insert floppies into the floppy drive.
- How to reset the computer.
- The hard disk is locked or "parked" when the system is shutdown properly. Refer to **shutdown**(C) in the XENIX *User's Reference* to shutdown the system. When you see this message:

  ** Normal System Shutdown **

  ** Safe to Power Off **
  – or –
  ** Press Any Key to Reboot **

  the hard disk is locked. Lock the hard disk if you move your 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 computer's *Operator's Guide* in order to detect possible hardware problems.

Other items to be aware of:

- Make sure the FILESYSTEM (N2) floppy is not write protected. (High density 96tpi distributions have a single BOOT/FILESYSTEM floppy.)

- Copy the BOOT and FILESYSTEM floppies (BOOT/FILESYSTEM) as soon as you can, then put the originals in a safe place and use the copies. Any floppy can fail with frequent use, so it is a good idea to back–up all your volumes, but especially these first, crucial floppies.

  Use the MS™–DOS floppy copy routine to make backups, before installing XENIX. If you cannot use the DOS routine, you can copy the floppies with the XENIX **diskcp**(C) command once you have installed the XENIX System. For more information on **diskcp**, see the XENIX *User's Reference*.

- If you are upgrading from XENIX 3.0 to XENIX System V, be sure to follow the upgrade procedure as outlined in the XENIX *Operating System Release Notes*.

2.5 Installation Procedure

2.5.1 Overview

The installation procedure has six steps:

1. Start XENIX from the BOOT Floppy (volume N1).

2. Initialize the hard disk with the **hdinit**(C) program and, if desired, modify the hard disk partition table to share space with the DOS operating system.

3. Re–start the system from the newly initialized hard disk.

4. Copy the XENIX utilities.

5. Create the super–user password.

6. Create the first user account.

The following sections describe each step. After you complete the installation, be sure to store this guide and the distribution floppies in a safe place. You will need them again if you wish to reinstall the system for any reason.

2.5.2 What You Need

To install the XENIX System you need:

- A personal computer with at least 384K bytes of memory.
- One hard disk with at least 10 megabytes of storage.
- One double-sided floppy drive.
- The XENIX System distribution floppy disks.
- A Serial Number (an alphanumeric code printed on the Serialization Card).
- An activation key (an alphabetic code printed on the Serialization Card).

The XENIX System distribution set contains these floppies:

- The XENIX Operating System floppies. The *Release Notes* delivered with this guide contain a list of the number of floppies in each set and the software packages contained in the Operating System.
- Optional: Development System floppies. The *Release Notes* delivered with the *Programmer's Guide* contain a list of the software packages contained in the Development System.
- Optional: Text Processing System floppies. The *Release Notes* delivered with the *Text Processing Guide* contain a list of the software packages contained in the Text Processing System.

2.5.3 Starting XENIX From The Boot Floppy Set

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

Note

If your XENIX distribution is on high density, 96tpi floppies, there is only one BOOT/FILESYSTEM floppy.

Follow these steps:

1. Insert the BOOT floppy into the drive. If you have more than one floppy drive, use the primary drive. It is 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 **bootstrap** program from the floppy disk and begins to execute 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. You see the default floppy boot, which looks something like this:

   fd(4,0)xenix

   Then the system performs a self-check to determine if there are any problems with the hardware. The letters A–H appear in succession.

   After XENIX is loaded in memory, the system displays information about how memory is allocated. You also see the number of multiscreens™ available on your system (see **multiscreen**(M) for more information on multiscreens). The number of multiscreens depends upon the amount of memory your computer has.

3. You see the prompt:

   Insert filesystem floppy and press <RETURN>

   Remove the BOOT floppy (N1) and insert the FILESYSTEM floppy (N2). If you have only one BOOT/FILESYSTEM floppy, leave it in the drive. Press:

   RETURN

4. The remaining letters of the alphabet are displayed. After Z, the message appears:

   No single–user login present
   Entering System Maintenance Mode

   If the letters stop displaying 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 again, call the Support Center listed on the support information card and be prepared to tell them at what letter the display ended.

The self–check using letters A–Z occurs every time you bring up your XENIX system. However, after this initial installation, the self–check does not display messages referring to the FILESYSTEM floppy (N2). Once the system begins to run, the following message is displayed:

XENIX System V Hard Disk Initialization

When you see this message, you are ready to proceed with the second installation step, *Initializing The Hard Disk*.

2.5.4 Initializing The Hard Disk

As part of the initialization process, you may partition the hard disk, using the **fdisk**(C) utility, to support both DOS and XENIX on the same hard disk, or you can allow XENIX to use the whole disk.

The following steps describe how to initialize the hard disk.

1. You see the message:

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

Note

If you have any files on the hard disk that you wish to save before installing XENIX follow this procedure. Enter:

n

and press RETURN.

The following message appears:

Aborting initialization procedure

At this point, the system shuts down automatically, as the following message is displayed on the screen:

** Normal System Shutdown **

** Safe to Power Off **
– or –
** Press Any Key to Reboot **

Remove the floppy, and reboot your existing operating system (DOS). After you have backed up the files you want to save, 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.

---

2. If you do not have any files you want to save, enter

   y

   and press RETURN.

   The **hdinit** program invokes **fdisk**(C), which partitions the hard disk.

Note

If you wish to partition the disk so that you can use another operating system in addition to XENIX, do not follow the **fdisk** steps shown here. Instead, see Chapter 3 of this *Installation Guide* and the **fdisk**(C) manual page in the XENIX *User's Reference*.

Install DOS on the hard disk before installing XENIX. Installing DOS after XENIX, in some circumstances, changes the **fdisk** table and makes the XENIX partition unbootable. If this happens, you must recreate the **fdisk** table. Note that DOS **fdisk** reports disk size in cylinders, XENIX **fdisk** reports disk size in tracks.

After you set up the desired partitions with **fdisk**, return to the next part of this installation procedure.

If you are only using XENIX on your hard disk, continue with the following steps.

---

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

   Select one of the following options or 'q' to exit the program

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

   Please enter your choice:

   Enter option "2" 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
1	Active	XENIX	1	1219	1219

Total disk size: 1220 tracks

Press <RETURN> to continue

Note

If you have previously installed an operating system on your disk, you see a table that might look something like this:

Current Hard Disk Drive: /dev/rhd00

Partition	Status	Type	Start	End	Size
1	Active	UNKNOWN	1	1219	1219

Total disk size: 1220 tracks

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

If you would like XENIX to occupy the whole disk, enter 'y' and press RETURN.

Even if you have already installed XENIX, and are just reinstalling the system, you should enter 'y' and press RETURN at this point if you want XENIX to occupy the whole disk. This insures that **fdisk** partitions the whole disk for XENIX.

Note that the track numbers and size of your disk may vary from this example.

Press RETURN, and you see the main **fdisk** menu. 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.

4. Now you see a menu from the program **badtrk**(C). With the **badtrk** program, you can scan your hard disk for defective tracks. The program maps any flaw locations to other, good tracks. It also creates a flaw map, which is a list of all the bad tracks on your hard disk.

   The main program menu looks like this:

```
Select one of the following options or 'q' to exit program:

 1. Print Current Bad Track Table
 2. Scan Media Surface for Possible Bad Spots
 3. Create New Bad Track Table
 4. Add Entries to Current Bad Track Table by Head/Cylinder #
 5. Add Entries to Current Bad Track Table by Sector Number
 6. Delete Entries from Current Bad Track Table

Please enter your choice:
```

   If you have never scanned your disk for bad tracks, enter:

```
2
```

   and press RETURN. The **badtrk** program scans the hard disk for possible flaws. The scanning process takes roughly one to two minutes per megabyte of storage on the disk.

   As the program scans the disk, it displays the number of each track it examines. Whenever it 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:

```
error on dev Fixed Disk (0/47), block=12954  cmd=0003  status=0018
sector = 12971, cylinder/head = 190/3
```

When the scan is complete, the menu reappears. Enter 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

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** runs again. Scan the disk for flaws.

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

Press RETURN to return to the main menu. If your disk comes with a flaw map, you should enter any flaws from it into the **badtrk** table. This procedure is described below.

Some disks are not furnished with a flaw maps. If this is the case with your disk, enter:

q

to leave **badtrk**. If you have no bad tracks, or made no changes to the bad track table, skip to the next step (5) on **divvy**(C).

If you created a new bad track table, or altered an existing one, the program prompts:

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

Enter:

y

If there are defective tracks, you are prompted:

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

Enter 'y' or 'n'. Either way, the process of mapping bad tracks takes several minutes. When **badtrk** finishes, continue the installation procedure.

If your disk is not furnished with a flaw map, skip to step "5" and proceed with the **divvy**(C) utility.

If your disk does come with a flaw map, proceed with the next steps to enter any defective tracks which are not already indicated in the **badtrk** flaw table.

Because most disk flaws are marginal or intermittent, your disk's flaw map will probably list more bad tracks than the scanning process reveals. If so, you should now add these defective tracks to the **badtrk** flaw table.

Select either option "4" or option "5" depending upon the format of the flaw map furnished with your disk. Enter the defective tracks, one per line. If you make a mistake, enter:

> q

and press RETURN. When you see the main **badtrk** menu, enter option "6" to delete a track.

When you are finished making changes to the flaw table, enter:

> q

and press RETURN to return to the main menu. At the main **badtrk** menu, enter:

> q

again and press RETURN. **badtrk** prompts:

> Do you want to update this device with the new table?

Enter:

y

and press RETURN to save the changes. You are prompted:

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

Answer 'y' or 'n'.

5. You now see several prompts from the **divvy**(C) 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. The **fsck**(C) program uses this *recover* area to place output when the *root* device is cleaned.

   If you create a very large root filesystem you are asked if you want to create an additional, small portion of the disk as scratch space for **fsck**. **fsck** uses the scratch space for temporary storage when cleaning very large filesystems. You should make a scratch filesystem if you have a very large *root* filesystem, since this makes booting XENIX much easier.

   After that, **divvy** prompts:

```
There are 10098 blocks in the XENIX area.
Between 1000 and 2000 blocks should be used for the swap area.

Please enter the swap-area allocation, or press <RETURN>
to get the default allocation of 1250 blocks:
```

3000

   The actual numbers in this prompt vary depending upon the size of your disk. If you do not plan to install the Development System, enter:

   RETURN

   for the default values. If you plan to install the Development System, enter a swap-area allocation that is approximately 500 blocks larger than the default.

   Next, you see:

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

If you are installing XENIX for the first time, 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 user's 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 answer 'y' you see a table from the **divvy** program. Refer to **divvy**(C) in the *User's Reference* for more information.

If your disk is larger than 20M bytes, 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 20M bytes, especially if you anticipate having many, active users. You should leave at least 15M bytes for the *root* filesystem.

You can call this separate filesystem anything you want, and, if desired, you can change the name later. The name should be different from any other directory or filesystem. We recommend that you use the name */u*. For more information on using a */u* filesystem, see Chapter 7 of the XENIX *Operations Guide*, "Using Peripheral Devices." That chapter describes adding a second hard disk (and filesystem), but the discussion of a second filesystem applies here.

If you have enough storage for a separate user filesystem, and want to create one, enter:

y

and press RETURN.

Note

If you create a */u* filesystem enter this command after you finish installing the XENIX system and **are in System Maintenance Mode:**

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

---

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

   Making filesystems

   When the **hdinit** program is finished, the system shuts down and displays instructions on booting the newly initialized hard disk. You see:

   ** Normal System Shutdown **

   Remove the FILESYSTEM floppy, and insert the BOOT floppy (N1) into the drive. If you have a single BOOT/FILESYSTEM floppy, leave it in the drive. Press any key to reboot the system.

If you want to add a second hard disk, finish the installation procedure. Then refer to Chapter 7 of the XENIX *Operations Guide*, "Using Peripheral Devices."

2.5.5 Starting XENIX On the Hard Disk

This section explains how to start the XENIX system using the hard disk.

1. After you see the boot prompt

   Boot
   :

   enter:

   xenix rootdev=hd(40,0)

   and press RETURN. You see some copyright information, and information about the memory configuration of your system.

   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.

   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.

2. Now the program **fsck**(C) is run. **fsck** checks the filesystem(s) on your hard disk.

If you have a very large *root* filesystem, you are prompted for a scratch filesystem name at this first boot up. If you created a scratch filesystem with **divvy**, enter:

/dev/scratch

This is the only time you have to enter a scratch filesystem name if you have a large filesystem and specified a scratch filesystem with **divvy**.

If you have a large filesystem, but do not specify a scratch filesystem with **divvy**, you must enter the name of a scratch filesystem every time you boot your system. An example scratch filesystem in this case is blank, formatted floppy:

/dev/fd0

3. When **fsck** finishes, you are prompted to make sure the BOOT (N1) floppy is in the drive. You see the following:

   Verify Operating System volume N1 is inserted
   and press <RETURN>

4. After this, 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.

   For example, the Link Kit is on one of your volumes and it is not installed at this time. You can install the Link Kit later in this procedure.

   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.

   If there is an error with the extraction procedure, such as a floppy error, the door of the floppy drive is not completely closed, or there is some other problem you may see the message:

   Extraction error: try again? (y/n)

   Enter:

   y

   and press RETURN.

5. When the last of the "N" volumes you are prompted for is read, you see the 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.

6. Next you are prompted to enter the "B" series of floppies. You see screen prompts like those for the "N" floppies.

   When the last of the "B" floppies are installed, you see:

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

   If daylight savings applies, enter 'y.' If not, enter 'n.' After you press RETURN, you see:

   Are you in North America? (y/n)

   If you enter 'y,' you see:

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

   Enter the number that represents your timezone:

   If, for example, your timezone is Nome Standard time, you would enter the number '8' and press RETURN.

   If you are not in the United States (or one of the time zones represented above), and entered "n", the following is displayed:

   What is the standard abbreviation of your timezone? (3 capital letters)

Type three upper case letters which represent your time zone and press RETURN. Next you will see this message:

How many hours west of Greenwich Mean Time are you?

Answer with a number between −23 and 23, then press RETURN. Do not use fractions.

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

You have now installed the minimal XENIX system, or run time system. You can use many of the standard XENIX utilities, or install other applications packages.

You see another menu which gives you the option of stopping or continuing with the installation. You can, at this point, stop the installation, or you can continue to install the XENIX Operating System in the next section, *Installing the XENIX Distribution*.

2.5.6 Installing the XENIX Distribution

You can now install more of the XENIX Operating System. With the **custom**(C) program you can selectively extract files from the distribution set to create your own custom XENIX system. You see information about your filesystem(s), including the number of blocks currently used. You also see this menu:

1. Continue installation
2. Stop installation

If you want to install all or part of the Operating System, enter option '1' and press RETURN.

Note that the entire XENIX distribution, including the Development System and Text Processing System requires more than 10M bytes of disk space.

If you choose option '1,' you are prompted for a set to install (customize). Choose from Operating System, Development System, and Text Processing System.

The **custom** program prompts for the necessary volume numbers. Insert the appropriate floppies and follow the screen prompts. For information on installing portions of the Operating System, see the XENIX *Reference* section on **custom**(C).

You can install the Link Kit at this time with **custom**. Note that you are prompted for your serial number and activation key again. Enter them as you did earlier in the installation.

When you are finished installing XENIX, you see a message about booting the system. You then see:

** Normal System Shutdown **

** Safe to Power Off **
– or –
** Press Any Key to Reboot **

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

Press any key to reboot the system and when you see the boot prompt:

Boot
:

press:

RETURN

You see:

hd(40,0)xenix

The screen clears and you see some self check diagnostics. You then see:

Type CONTROL–d to proceed with normal startup
(or give root password for system maintenance)

Press RETURN at this point since there is no root password. The next step is to create a password for the root account.

2.6 Creating the Super–User Password

The super–user password keeps the system safe from unauthorized use. It is important that you create a super–user password immediately after the system has been installed to ensure maximum protection of the system and prevent unnecessary use of the super–user (also known as "root") account.

It is very easy to make errors when logged in as super–user that could destroy files. **Login as super–user only to install programs and to do system maintenance tasks.** For a complete description of the super–user, see the XENIX *Operations Guide*.

To create the super–user password, follow these steps:

1. Enter:

   passwd root

   and press RETURN.

   The system displays the message:

   New password:

   The new password can be any sequence of letters, numbers, and/or punctuation marks, but should be at least 5 characters long.

2. Enter the new password and press RETURN.

   The system does not display the password as you enter so enter carefully. After you press RETURN the system displays the message:

   Retype new password:

3. Enter the new password once more and press RETURN. Make sure you enter it correctly, otherwise the program will ignore the change.

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.

Do not forget the super–user password. To restore a forgotten super–user password you must reinstall the XENIX system. If necessary, keep a copy of the super–user password in a safe place.

2.6.1 Sysinfo Account

There is a special account used to perform system backups. The "sysinfo" account is set up with permissions and privileges similar to the super–user account, but only for performing system backups. See the chapter "Backing Up File Systems" in the XENIX *Operations Guide* for more on this procedure.

We suggest you create a password for the "sysinfo" account when you create the password for the super–user account "root".

2.7 Creating the First User Account

The last step in the installation is to create the system's first user account, "guest". This guest account is a temporary workspace on the system that you may use to practice with the XENIX system. Later, after installation is complete and you are familiar with the XENIX commands, you can remove the guest account and create private accounts for all the system users.

To create the first user account, follow these steps:

1. Enter:

   mkuser

   and press RETURN. The system displays the message:

   Newuser
   Add a new user to the system

   Do you require detailed instructions? (y/n):

   Enter:

   n

   and press RETURN (you can examine the instructions at some other time). You can quit from the program at any "(y/n)" prompt by entering the letter 'q' and pressing RETURN.

2. The system prompts for more information:

   Enter new user's login name:

   Enter:

   guest

   and press RETURN. The name "guest" is now the login name for the new user account.

3. Next, the program prompts for a group name:

   Do you want to use the default group? (y/n)?

   Enter:

   y

   This sets the group to "group".

4. Next, the program prompts for the new user's password.

   Enter password

   Press RETURN. This allows you to use the guest account without giving a password.

5. Next, the program prompts for the shell type. The following displays:

   Please specify the type of shell (command interpreter)
   the user requires. You can type 1, 2, 3, 4, or 5 as follows:

   1 Standard (Bourne) Shell.
   2 Visual Shell.
   3 C Shell.
   4 Restricted Shell.
   5 Uucp.

   Enter "1" and press ENTER. The guest account has an **sh**(C) shell.

6. Finally, the program prompts you for comments.

   Please Enter Comment >........................
   >

   Enter:

   guest account

   and press RETURN.

7. The system then prompts if you want to change anything. Enter:

   n

   and press RETURN.

8. Finally, you see the prompt:

   Add another user? (y/n)

   Enter:

   n

   and press RETURN.

The new guest account is ready. Later, when you turn to the XENIX *User's Guide*, you may use this account to, for instance, practice logging in, make directories and run programs.

2.8 The Next Step

If you created a */u* filesystem earlier in the installation, you should now run this command:

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

Of course, if you chose a name other than *u* for the second filesystem, use that name in the command line above.

If you are familiar with the XENIX Operating System, you may continue with normal startup and begin working. Just press down the CONTROL key and enter "d". Refer to the explanation of normal startup in the XENIX *Operations Guide* if you have problems.

If you are not familiar with the XENIX Operating System, we recommend that you halt the system and turn to the XENIX *User's Guide* and the XENIX *Operations Guide* to learn how to start the system, how to log in, and how to run programs.

To halt the system, follow these steps:

1. Enter:

   ```
   /etc/shutdown 0
   ```

   and press RETURN.

2. Wait for the following message:

   ** Normal System Shutdown **

   ** Safe to Power Off **
   – or –
   ** Press Any Key to Reboot **

3. When you see the shutdown message, it is safe to turn off the power to the computer.

2.9 Troubleshooting

Sometimes things can go wrong in the installation procedure. 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.

You can avoid most simple errors by reading the *Release Notes* delivered with this product, and this *Installation Guide* completely before you try to install the software.

During the installation procedure, don't assume you know what is about to happen, even if you have installed the XENIX system before. Use the documentation wisely.

If you have difficulty installing the software, here is a list of some of the most common problems, how to avoid them, and how to fix them if they happen:

- *Some hardware* (for example, a disk drive) *doesn't seem to work, although it works fine under another operating system.*

  Certain hardware configurations do not work with XENIX. Refer to the "Compatible Hardware" section in the *Release Notes* for information on what hardware you can use with XENIX.

- *XENIX is installed after DOS, now both operating systems do not work*

  Do not install XENIX after installing DOS on that same hard disk. This is because of the way the two operating systems keep track of space on a disk. XENIX uses tracks, whereas DOS uses cylinders, and, if you install XENIX first, DOS may not find the correct end of its alloted partition.

  If you already have DOS installed on you hard disk, and with to install XENIX, you must first copy the DOS information to your backup media. Then install the XENIX Operating System. Finally, replace DOS.

  See the chapter in this *Installation Guide* "Using DOS And XENIX On The Same Disk."

- *Error reading a floppy.*

  Make sure you insert the correct floppy at the correct time. Insert the floppy correctly (see your owner's manual if you do not know how insert the floppy correctly).

Make sure the floppy drive door is closed after you insert a floppy.

If you are sure the proper floppy is inserted correctly, and you still have a read error, try tapping the floppy lightly against a hard surface, such as a table top. Be careful not to damage the media, though!

- *The system does not boot from the BOOT floppy.*

  Make sure you insert the BOOT floppy. If you insert another floppy instead, you do not see an error message, but the system still does not boot.

  The BOOT floppy is shipped with a write protect tab so that you do not accidentally erase it during the installation process. If you erase or damage the BOOT floppy, and you have not made a backup copy, call the SoftCare Support Center number to arrange for a new BOOT floppy.

- *The system won't boot from the hard disk*

  You may see a message such as:

      panic: iinit

  or another kind of error message. Sometimes this happens because you did not run a bad track scan during installation, and the boot block was written on a bad track.

  Whatever the reason, you must reinstall the XENIX system. If you didn't enter all the flaws furnished on a flaw map, do so this time.

  If XENIX still won't boot, run your system's and disk's hardware diagnostics.

- *You forget to enter more bad tracks from a manufacturer furnished flaw map.*

  You can add more flaw locations to the bad track map any time you are in System Maintenance mode. Follow these steps if you are still in the installation procedure:

  - Finish installing the run time system.

  - When you are prompted to continue installing, or stop the installation process, stop the installation process.

  - Enter system maintenance mode.

- Run **badtrk** to add the new flaws.
- Salvage any data (when prompted under **badtrk**).
- Run the **custom** utility to finish installing your system.

Chapter 3

Using DOS and XENIX On the Same Disk

3.1 Introduction

Many users received the MS–DOS, or other closely compatible DOS, operating system with their computer. This chapter explains how you can still use DOS utilities, files, and applications after you install the XENIX operating system. You can even access DOS files and directories from XENIX. You do not need to throw away your investment in DOS software, or buy another computer just to run XENIX.

Several programs make this 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 utility which partitions the disk is called **fdisk**(C) and is available in DOS and XENIX versions. The next section explains how to use **fdisk** to run DOS and XENIX on the same hard disk. Another section discusses installing XENIX on the hard disk along with DOS. There is also a section explaining various booting configurations, for users who mostly use XENIX and for users who mostly use DOS.

3.2 Partitioning the Hard Disk Using fdisk

Each version of **fdisk** is documented in the respective operating system's manual. **fdisk**(C) is found in the XENIX *Reference*.

fdisk is interactive and a menu displays your options. When you exit **fdisk,** the utility asks you whether or not you actually desire to make the specified changes. This feature lessens the chance of making a mistake. Here is an example of an **fdisk** menu:

```
1. Create Partition
2. Change Active Partition
3. Delete Partition
4. Display Partition Table
5. Create XENIX Partition for Whole Disk

Enter Choice:
```

The **fdisk** utility allows you to set up separate areas (partitions) on your hard disk for both operating systems. 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 files. Not all operating systems can be used with **fdisk.**

The **fdisk** command allows you to specify which partition is "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 the XENIX operating system.

The **fdisk** command allows you to specify the number of tracks used by the partition. This will vary according to the size of your hard disk. We recommend using at least a 10 megabyte hard disk to run XENIX. The size of the XENIX partition also depends on the number of software packages you want to install. Refer to the **custom**(C) manual page for information on the installing and removing packages from the three XENIX distribution Systems. 6 megabytes is a good starting point. You can install the XENIX Operating System package in this space, and have space for user files.

The **fdisk** command allows you to specify where the partition begins. **fdisk** will not allow you to construct overlapping partitions, or a partition that begins with a bad track. You do not need to install XENIX in the first partition. When you are running XENIX, the device name of the partition running XENIX is */dev/hd0a*.

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 track used, and the associated operating system. If you enter the Display Partition Table option and press RETURN to see the partition table, the result may look like this:

PARTITION	STATUS	TYPE	START	END	SIZE
1	A	XENIX	001	800	800
2	N	DOS	801	1219	420

There are two ways to switch operating systems once you have set up separate XENIX and DOS partitions. You can use a floppy diskette with the files necessary to boot the DOS operating system. Because the XENIX partition must be active for XENIX to operate, you cannot use a bootable floppy to boot XENIX. You can use **fdisk** to change the current active partition. This method is appropriate for an occasional change of the active operating system. Turn the computer off. When you re-boot, the operating system associated with the active partition starts running. It is recommended you use a boot floppy to boot the DOS operating system. Booting from a floppy is generally easier, faster and safer than constantly using **fdisk** to change active partitions.

3.3 Installing XENIX on a DOS System

If you wish to set up XENIX on a hard disk which previously contained only DOS, first back up all the DOS files and directories. Run **fdisk**, under DOS, and assign a partition to DOS, leaving enough disk space so that XENIX has at least 6 megabytes. Delete DOS from the partition you will assign to XENIX. It will display as *Other* when you display the Partition Table. Make a note of the tracks assigned to each operating system. Designate *Other* as the active operating system.

Follow the installation procedure outlined in Chapter 2 of this guide to install XENIX. During the installation procedure **fdisk** is invoked to partition the hard disk. Use **fdisk**, to duplicate the assignments made under DOS and assign a partition which is at least 6 megabytes to XENIX. You will see a message warning that the contents of the hard disk will be destroyed. Don't worry, you backed up the DOS files in this area, and will restore them later. This partition will contain XENIX. Designate "XENIX" as the active operating system.

Note

Under XENIX, **fdisk** DOS partitions display as *DOS* while under 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** usually reports the size of the hard disk in terms of cylinders.

Reboot the system using a DOS boot floppy in the floppy drive or use **fdisk** to designate *DOS* as the active operating system, and reboot. You can reinstall your DOS files and directories in the separate partition anytime after you have set up the DOS partition, and finish the XENIX installation. You are then accessing the DOS partition and can reinstall DOS.

If you have an active XENIX partition and boot DOS from a floppy you can transfer to C: to work with the DOS files.

3.4 Using XENIX and DOS With Tw

Your computer always boots the op ... on the first hard disk. XENIX must bc ... are several ways to configure your syste ... ays are discussed here.

debug

-g=c800:5

One configuration consists of desig ... IX partition. You then use a DOS boot

```
A>  A: D:
```

to switch to the DOS area on the second hard disk, where **D** is the designation for the second hard disk.

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 operating system.

Note

Be sure to make a backup copy of your boot floppies if you use them to boot your secondary operating system.

3.5 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 **fdisk** under both operating systems to delete the partition in question. 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 **mkfs**(C). Refer to Chapter 7, "Using Peripheral Devices," of the XENIX *Operations Guide* if you add a second XENIX partition and want to designate this partition as a mounted filesystem.

3.6 DOS Accessing Utilities

There is a set of other utility programs which help you bridge the two operating systems. These are the XENIX commands, such as **dosls** and **doscat**, described in the XENIX manual page **dos**(C). These programs allow you to access DOS files and directories which reside in a non-active DOS partition while running XENIX. You can list, copy, move and view the contents of files, and create directories. 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) command tells you what type of floppies you have (various DOS and XENIX types).

Note

> You can not execute (run) DOS programs or applications under XENIX.

The XENIX Development System, with the cmerge compiler, has the capability of using the XENIX System to create and compile programs which can be run under DOS operating systems. Refer to the XENIX *C User's Guide* appendix entitled "XENIX to MS–DOS: A Cross Development System" and the *C Library Guide* Appendix entitled "A Common Library for XENIX and MS–DOS" for more on using XENIX to create DOS programs.

XENIX® System V

Operating System

Introduction to XENIX

This document was typeset with an IMAGEN® 8/300 Laser Printer.

SCO Document Number: XG-4-1-87-4.0

Contents

Chapter 1

Introduction

1.1 Overview

This guide introduces key concepts of the XENIX system by presenting them in a tutorial format.

It begins with a "demonstration" that explains an actual computer session, including command usage and correcting typing errors. Basic concepts such as files, commands, and pattern matching are also introduced.

Finally, these and other concepts are applied to many "real world" examples, such as file manipulation, terminal configuration, process control, and status information.

Note

This guide should be read before the other XENIX documentation; however, for more detailed discussions of all topics covered here, consult the other user's and reference guides in the set.

1.2 The XENIX System

The XENIX system consists of a general-purpose multi-user operating system and over one hundred utilities and application programs. In addition to the XENIX Operating System described in this guide, two other XENIX system packages are available: the XENIX Development System and the XENIX Text Processing System.

1.3 The XENIX Working Environment

The XENIX system is built around the XENIX operating system. The purpose of an operating system is to efficiently organize and control the resources of a computer so that they can be used by real people. These resources include memory, disks, lineprinters, terminals, and any other peripheral devices connected to the system. The heart of the XENIX system is a "multi-user" and "multi-tasking" operating system. A multi-user system permits several users to use a computer simultaneously, thus providing lower cost in computing power per user. A multi-tasking system permits several programs to run at the same time and increases productivity because multiple programs can run simultaneously rather than in sequence.

Because UNIX ™ (and thus XENIX) is an accepted standard for "high-end" operating systems, a great deal of software is available for this environment. In addition, XENIX provides file access to the MS-DOS™ operating system, the most widely used 16-bit operating system in the world. For systems that support DOS, XENIX provides commands that let you access DOS format files and disks. The XENIX system also includes several widely praised enhancements developed at the University of California at Berkeley, and a visual interface similar to other Microsoft productivity tool interfaces.

Other characteristics of the XENIX system include:

- A powerful command language for programming XENIX commands. Unlike other interactive command languages, the XENIX "shell" is a full programming language.

- Simple and consistent naming conventions. Names can be used absolutely, or relative to any directory in the file system.

- Device-independent input and output: each physical device, from interactive terminals to main memory, is treated like a file, allowing uniform file and device input and output.

- A set of related text editors, including a full screen editor.

- Flexible text processing facilities. In XENIX , commands exist to find and extract patterns of text from files, to compare and find differences between files, and to search through and compare directories. Text formatting, typesetting, and spelling error-detection facilities, as well as a facility for formatting and typesetting complex tables and equations are also available.

- A sophisticated "desk-calculator" program.

- Mountable and dismountable file systems that permit addition of floppy disks to the file system.

- A complete set of flexible directory and file protections that allows all combinations of read, write, and execute access for the owner of each file or directory, as well as for groups of users.

- Facilities for creating, accessing, moving, and processing files and directories in a simple and uniform way.

1.4 About This Guide

This guide is organized as follows:

Chapter 1, "Introduction," gives an introduction and overview of the XENIX system.

Chapter 2, "Demonstration," gives you hands-on experience in using the XENIX system.

Chapter 3, "Basic Concepts," explains the fundamental concepts that you need to understand before you begin to use the system. Included here are sections on the file system, naming conventions, commands, and input and output.

Chapter 4, "Tasks" explains how to perform everyday tasks using appropriate XENIX commands.

Replace this Page
with Tab Marked:

Demonstration

Chapter 2

Demonstration

2.1 Introduction

This chapter contains a demonstration run designed to help you get used to the XENIX system, so that you can quickly start to make effective use of it. It shows you how to log in, how to enter at your keyboard, what to do about mistakes in entering, how to enter commands and how to log out.

2.2 Before You Log In

Before you can log in to the system, your name must be added to the XENIX user list. At that time you will be given a login name and a password. You may have to add your name yourself, or someone else may be assigned this task; it all depends on the environment in which your system is used. In any case, see the XENIX *Operations Guide* and **mkuser**(C) for detailed information on adding users.

When you are given an account on the XENIX system you will also receive a user name, a password, and a login directory. Once you have these, all you need is a terminal from which you can log in to the system. XENIX supports most terminals and you should have no problem getting your terminal to work with XENIX . Once again, see the XENIX *Operations Guide* for more information on how to configure your terminal.

2.3 Logging In

Normally the system is sitting idle with a "login:" prompt on the terminal screen. If the system displays nonsense characters when you enter text, then your terminal is probably receiving information at the wrong speed and you should check your terminal switches. If the switches are set correctly, push the **BREAK** or **INTERRUPT** key a few times.

When you get a "login:" message, enter your login name, then press **RETURN**; the system will not do anything until you do. If a password is required, you will be asked for it. The password that you enter does not appear on the screen. This prevents others from viewing it. Do not forget to press **RETURN** after you enter your password. Next you see the line

TERM = (unknown)

Enter your terminal type (for example, ansi) and press **RETURN**.

A successful log in produces a "prompt character", a single character that indicates the system is ready to accept commands. The prompt is usually a dollar sign ($) or a percent sign (%).

You may also get a login message such as:

You have mail

telling you that another system user has sent you mail.

2.4 Typing Commands

Once the prompt character appears, the system is ready to respond to commands entered at the terminal. Try entering:

date

followed by **RETURN**. The system responds by displaying something like:

Mon Jun 16 14:17:10 EST 1985

Do not forget to press the **RETURN** key after the command, or nothing will happen. The **RETURN** key will not be mentioned again, but do not forget -- it has to be entered at the end of each command line. On some terminals **RETURN** may be labeled "ENTER" or "CR", but in all cases, the key performs the same function.

Another command you might try is **who**, which lists the names of everyone who is logged in to XENIX. A typical display from the **who** command might look something like this:

```
you    console    Jan 16    14:00
joe    tty01      Jan 16    09:11
ann    tty02      Jan 16    09:33
```

The time, given in the fourth column, indicates when the user logged in; tty*nn* is the system name for each user's terminal, where *nn* is a unique two-digit number. The console is the special name of the master terminal that is the default for most operations.

If you make a mistake entering the command name, you will see a message on your screen. For example, if you enter:

whom

the system responds with the message:

whom: not found

Note that case is significant in XENIX. The commands

who

and

WHO

are not the same; this differs from some operating systems, where case does not matter.

Now try displaying a message on your screen using the **echo** command. Type:

echo hello world

The **echo** command does what its name implies and echoes the rest of the command line to your terminal:

hello world

Now try this:

echo hello world >greeting.file

This time the **echo** command sends its output to a new file named *greeting.file*, instead of to your terminal. Note the use of the greater-than sign (>) to "redirect" the output of the command. Now enter:

ls

to list just the name of the file. To look at the contents of display it by entering:

cat greeting.file

Here "cat" stands for concatenate. One purpose of the **cat** command is to combine the contents of several files (that is, "concatenate") and put them in some new file. However, since your terminal display is treated like any other file in XENIX, **cat** is most commonly used to display the contents of files on the screen. Therefore the above command sends the following output to your terminal screen:

hello world

To remove greeting.file, enter:

rm greeting.file

Note that XENIX command names are often shortened to mnemonic names. For example, **cp** is short for "copy", **ls** is short for "list", **rm** is short for "remove", **cat** is short for "concatenate", **mkdir** is short for "make directory", and **chmod** is short for "change mode".

2.5 Mistakes in Typing

If you make a mistake in entering while entering a command, there are two ways to edit the line, provided you have not yet pressed **RETURN**. Pressing the **BKSP** key causes the last character entered to be erased. Backspacing with the **BKSP** key can erase characters back to the beginning of the line, but not beyond. Thus, if you type badly, you can correct as you go. For example, entering:

ddBKSPateRETURN

is the same as

dateRETURN

The XENIX kill character, **Ctrl-u**, erases all of the characters entered so far on the current input line. So, if the line is irretrievably fouled up, enter **Ctrl-u** and start the line over.

If you must enter a **BKSP** or **Ctrl-u** as part of the text, precede it with a backslash (\), so that the character loses its special ""erase"" meaning. To enter a **BKSP** or **Ctrl-u** in text, enter "\BKSP" or "\Ctrl-u". The system always prints a new line on your terminal after your **Ctrl-u**, even if preceded by a backslash. Nevertheless, the **Ctrl-u** will have been recorded.

To erase a backslash, backspace twice with the BKSP key, as in "\BKSPBKSP". The backslash is used extensively in XENIX to indicate that the following character is in some way special. Note that the functions performed by **BKSP** and **Ctrl-u** are available on all XENIX systems; however, the keys used to perform these functions may vary and can be set by the user with **stty**(C).

2.6 Read-Ahead and Type-Ahead

XENIX has full read-ahead, which means that you can type as fast as you want, whenever you want, and XENIX will remember what you have entered. If you enter any text while a command is displaying text on the screen, your input characters appear intermixed with the output characters on the screen, but they are stored away and interpreted in the correct order. Therefore, you can enter several commands (i.e., "type ahead") one after another without waiting for the first to finish. Note that this does not work when you log in; type-ahead does not work until *after* you have entered your password and the dollar sign ($) prompt appears.

2.7 Strange Terminal Behavior

Occasionally, your terminal may act strangely. You can often fix such behavior by either turning your terminal off, then quickly turning it back on, or logging out and logging back in; this will reset your terminal characteristics. It is often helpful to enter a **Ctrl-q**. This restores terminals that are (inadvertantly or otherwise) in a non-echoeing mode. **Ctrl-s** stops display to the screen, **Ctrl-q** restarts display. If logging out and back in, turning the terminal off and on, and entering **Ctrl-q** does not work, read the description of the command **stty**(C) in the *XENIX Reference Manual* for more information about setting terminal characteristics. Also, refer to the next section, "Stopping a Program."

2.8 Stopping a Program

You can abort the execution of most programs and commands by pressing the **INTERRUPT** key (perhaps called **DEL**, **DELETE**, **Ctrl-c**, or **RUBOUT** on your terminal). The **BREAK** key found on many terminals can also be used. Inside some programs, like most text editors, entering **INTERRUPT** stops whatever the program is doing without aborting the program itself. Throughout this manual, when we say "send an interrupt" we mean press the **INTERRUPT** key.

2.9 Logging Out

To end a session with XENIX , you must log out. This is done by entering **Ctrl-d** as the first character on a line. It is not sufficient just to turn off the terminal, since this does not log you out. Some programs can also be ended by entering **Ctrl-d**, so beware.

Replace this Page
with Tab Marked:

Basic Concepts

Chapter 3

Basic Concepts

3.1 Introduction

This chapter will give you an understanding of the basic concepts you need to function in the XENIX environment. After reading this chapter you should understand how the system's files, directories, and devices are organized and named, how commands are entered, and how a command's input and output can be manipulated. This chapter begins with a discussion of files.

3.2 Files

The file is the fundamental unit of the XENIX file system. In XENIX there are really three different types of files: ordinary files (what we usually mean when we say "file"), directories, and special files. Each of these types of files is described below.

3.2.1 Ordinary Files

Ordinary files typically contain textual information such as documents, data, or program sources. Executable binary files are also of this type. An ordinary file is simply a named concatenation of 8-bit bytes. Whether these bytes are interpreted as text characters, binary instructions, or program statements is up to the programs that examine them. Every ordinary file has the following attributes:

- A filename (not necessarily unique)
- A unique system number called an inode number
- A size in bytes
- A time of creation
- A time of modification
- A time of last access
- A set of access permissions

Files can be protected by assigning appropriate access permissions to assure privacy and security. This is done by providing read-write-execute permissions to files so that the user can control access by the owner, by a group of users, and by anyone else. By default, the owner of a file is its creator. The owner can read the file or write to it. By default, other users can read a file owned by another, but not write to it. File permissions can be altered with the **chmod** command. This command is discussed in Chapter 4 of this manual.

3.2.2 Special Files

Special files correspond to physical devices such as hard and floppy disks, lineprinters, terminals, and system memory. They are called "device special files". These files are not discussed in this manual.

3.2.3 Directory files

Directory files are read-only files containing information about the files or directories that are conceptually (but not physically) contained within them. This information consists of the name and inode number of each file or directory residing within the given directory. An inode number is a unique number associated with any given file. *All* files on the system have inode numbers. A name/inode number pair is called a link. The **ls** command is used to examine directory files and to list the information about the files conceptually within the named directory. With the inode number, the **ls** command can also find other information about a file.

The nesting of directories inside other directories is the way in which XENIX implements its characteristic tree-structured directory system. Directories are discussed further in the next section.

Like ordinary files, directories can be protected by assigning appropriate access permissions to assure privacy and security. This is done by giving read-write-search permissions to directories so that the user can control directory access by the owner, by a group of users, and by anyone else. Write permission determines whether files can be added or removed from a directory. By default, the owner of a directory is its creator, and the owner can read, create or remove files within that directory. Similarly by default, a user can read files within the directory of another, but not add or remove files. As with file permissions, directory permissions can be altered with the **chmod** command. Default permissions can be altered with the **umask** command.

3.2.4 Directory Structure

With multiple users and multiple projects, the number of files in a file system can proliferate rapidly. Fortunately, as mentioned earlier, XENIX organizes all files into a tree-structured directory hierarchy. This tree structure should be thought of as a physical world in which the user can move from place to place. "Places" are directories. Each user of the system has his own personal directory. Within that directory, the user may have directories or other subdirectories owned and controlled only by the user.

When you log in to XENIX , you are "in" your directory. Unless you take special action when you create a file, the new file is created in your working directory. This file is unrelated to any other file of the same name in someone else's directory.

A diagram of part of a typical user directory is shown in Figure 3-1.

```
                usr
              /  |  \
            /    |    \
          /      |      \
      adam      eve      mary
      /        /   \        \
  text        /     \       text
            text    temp
```

Figure 3-1 A Typical User Directory

In Figure 3-1, the *usr* directory contains each user's own personal directory. Notice that Mary's file named *text* is unrelated to Eve's. This is not important if all the files of interest are in Eve's directory, but if Eve and Mary work together, or if they work on separate but related projects, this division of files becomes handy indeed. For example, Mary could print Eve's text by typing:

```
pr /usr/eve/text
```

Similarly, Eve could find out what files Mary has by typing:

```
ls /usr/mary
```

3.3 File Systems

A file system is a set of files organized in a certain way. In XENIX , this set of files consists of all available resources including data files, directories, programs, lineprinters, and disks. Thus, the XENIX file system is a system for accessing all system resources.

To logically structure the resources of the system, the XENIX file system is organized hierarchically in an inverted "tree structure". See Figure 3-2 for an illustration of a typical tree-structured file system. In this typical tree of files, the root of the tree is at the top and branches of the tree grow downward. Directories correspond to nodes in the tree; ordinary files correspond to "leaves". If a directory contains a downward branch to other files or directories, then those files and directories are "contained" in the given directory. It is possible to name any file in the system by starting at the root (where the root is at the top) and traveling down any of the branches to the desired file. Similarly, you can specify any file in the system, relative to any directory. Specification of these files depends on a knowledge of the XENIX naming conventions, discussed in the next section.

```
                    /  (root)
                 /  |  \
               /    |    \
             /      |      \
        bin      usr      dev
        /       /    \        \
    cmd       /        \        tty
           doug      neil
           /  |        |  \
         /    |        |    \
    mail  news     text   data
```

Figure 3-2 A Typical File System

In the typical tree-structured file system of Figure 3-2, the "tree" grows downward. The names *bin*, *usr*, *dev*, *doug*, and *neil* all represent directories, and are all nodes in the tree. In XENIX the name of the root directory is given the one-character name, " / ". The names *mail*, *news*, *text*, and *data* all represent normal data files, and are all "leaves" of the tree. Note that the file *cmd* is the name of a command that can be executed. The name *tty* represents a terminal and is also represented in the tree.

3.4 Naming Conventions

Every single file, directory, and device in XENIX has both a filename and an absolute pathname. This pathname is a map of the file or directory's location in the system. The absolute pathname is unique to all names in the system; filenames are unique only within directories and need not be unique system-wide. This is similar to someone whose "global" name is John Albert Smith in a telephone directory, but who may be listed simply as John in an office phone list.

3.4.1 Filenames

A simple filename is a sequence of one to fourteen characters other than a slash (/). Every single file, directory, and device in the system has a filename. Filenames are used to uniquely identify directory contents. Thus, no two filenames in a directory may be the same. However, filenames in different directories may be identical.

Although you can use almost any character in a filename, it is best to confine filenames to the alphanumeric characters and the period. Other characters, especially control characters, are discouraged for use in filenames. When a filename contains an initial period, it is "hidden", and is not displayed by the lc command. However the ls–a command will display the hidden files. The dash (–) is used in specifying command options, and should be avoided when naming files. In addition, the question mark (?), the asterisk (*), brackets ([and]), and all quotation marks should *never* be used in filenames, since they are treated specially when entering commands.

3.4.2 Pathnames

A pathname is a sequence of directory names followed by a simple filename, each separated from the previous name by a slash. If a pathname begins with a slash, it specifies a file that can be found by beginning a search at the *root* of the entire tree. Otherwise, files are found by beginning the search at the user's *current directory* (also known as the *working directory*). The current directory should be thought of as your location in the file system. Think of it as a physical place. When you change your current directory you are moving to some other directory or place in the file system.

A pathname beginning with a slash is called a *full* (or *absolute*) *pathname* because it does not vary with regard to the user's current directory. A pathname *not* beginning with a slash is called a *relative pathname*, because it specifies a path relative to the current directory. The user may change the current directory at any time by using the **cd** command. The user may display the current directory by using the **pwd** command.

3.4.3 Sample Names

Some sample names follow:

/	The absolute pathname of the root directory of the entire file system.
/bin	The directory containing most of the frequently used XENIX commands.
/usr	The directory containing each user's personal directory. The subdirectory, */usr/bin* contains frequently used XENIX commands not in */bin*.
/dev	The directory containing files corresponding to physical devices (e.g., terminals, lineprinters, and disks).
/dev/console	The name of the system master terminal.
/dev/tty	The name of the user's terminal.
/lib	The directory containing files used by some standard commands.

/tmp — This directory contains temporary scratch files.

/usr/joe/project/A — A typical full pathname; this one happens to be a file named *A* in the directory named *project* belonging to the user named *joe*.

bin/x — A relative pathname; it names the file *x* in subdirectory *bin* of the current working directory. If the current directory is /, it names */bin/x*. If the current directory is */usr/joe*, it names */usr/joe/bin/x*.

file1 — Name of an ordinary file in the current directory.

When using the XENIX system, each user resides "in" a directory called the current directory. All files and directories have a "parent" directory. This directory is the one immediately above, which "contains" the given file or directory. The XENIX file system provides special shorthand notations for this directory and for the current directory:

. — The shorthand name of the current directory. Thus *./filexxx* names the same file as *filexxx*, if such a file exists in the current directory.

.. — The shorthand name of the current directory's parent directory. The shorthand name ../.. refers to the directory that is two levels "above" the current directory

3.4.4 Special Characters

XENIX provides a pattern-matching facility for specifying sets of filenames that match particular patterns. For example, examine the problem that occurs when naming the parts of a large document, such as a book. Logically, it can be divided into many small pieces such as chapters or sections. Physically, it must be divided too, since the XENIX editor **vi** cannot handle really big files.

Thus, you should divide a large document into several files. The points at which the document is divided should follow a logical order. You might have a separate file for each chapter:

```
chap1
chap2
...
```

Or, if each chapter is broken into several files, you might have:

```
chap1.1
chap1.2
chap1.3
...
chap2.1
chap2.2
...
```

You can then tell at a glance where a particular file fits into the whole.

There are other advantages to a systematic naming convention that are not so obvious. What if you want to print the whole book on the lineprinter? You could enter:

```
lpr chap1.1 chap1.2 chap1.3...
```

but you will tire of this quickly and will probably even make mistakes. Fortunately, there is a shortcut: a sequence of names containing a common pattern can be specified with the use of special characters. The special characters discussed in this chapter are:

- * Matches zero or more characters
- [] Matches any character inside the brackets
- ? Matches any single character

For example, you can enter:

```
lpr chap*
```

The asterisk (*), sometimes called "star" or "splat" in XENIX , means "zero or more characters of any type", so this translates into "send all files whose names begin with the word "chap" to the lineprinter".

This shorthand notation is not a unique property of the **lpr** command; it can be used in any command.

Using this fact, you can list the names of the files in the book by typing:

```
ls chap*
```

This produces

```
chap1.1
chap1.2
chap1.3
...
```

The star is not limited to the last position in a filename; it can be used anywhere and can occur several times. A star by itself matches all filenames not containing slashes or beginning with periods, so:

```
cat *
```

displays all files in the current directory on your terminal screen.

The star is not the only pattern-matching feature available. Suppose you want to print only chapters 1 through 4, and 9. You can say:

```
lpr chap[12349]*
```

The brackets ([and]) mean "match any of the characters inside the brackets." A range of consecutive letters or digits can be abbreviated, so you can also do this with:

```
lpr chap[1-49]*
```

(Note that this does *not* match forty-nine filenames, but only five.) Letters can also be used within brackets: "[a-z]" matches any character in the range "a" through "z".

The question mark (?) matches any single character, so

```
ls ?
```

lists all files that have single-character names, and

```
ls -l chap?.1
```

lists information about the first file of each chapter (i.e., *chap1.1*, *chap2.1*, ...).

If you need to turn off the special meaning of any of the special characters (*, ?, and [...]) enclose the entire argument in single quotation marks.

For example, the following command will print out only files named "?" rather than all one-character filenames:

```
ls '?'
```

Pattern-matching features are discussed further in Chapter 4 of the XENIX *User's Guide*, "The Shell."

3.5 Commands

Commands are used to invoke executable programs. When you enter the name of a command, XENIX reads the command line that you have entered, looks for a program with the given name, and then executes the program if it finds it. Command lines may also contain arguments that specify options or files that the program may need. The command line and command syntax are discussed in the next two sections.

3.5.1 Command Line

Whether you are entering commands at a terminal, or XENIX is reading commands from a file, XENIX always reads commands from command lines. The command line is a line of characters that is read by the shell command interpreter to determine what actions to perform. This interpreter, or "shell" as it is known, reads the names of commands from the command line, finds the executable program corresponding to the name of the command, then executes that program. When the program finishes executing, the shell resumes reading the command line. Thus, when you are entering at a terminal, you are editing a line of text called the *command-line buffer* that becomes a command line only when you press **RETURN**. This command-line buffer can be edited with the **BKSP** and Ctrl-u keys. Pressing **RETURN** causes the command-line buffer to be submitted to the shell as a command line. The shell reads the command line and executes the appropriate command. If you press **INTERRUPT** before you press **RETURN**, the command-line buffer is erased. Multiple commands can be entered on a single command line provided they are separated by a semicolon (;). For example, the following command line prints out the current date and the name of the current working directory:

```
date ; pwd
```

Commands can be submitted for processing in "the background" by appending an ampersand (&) to the command line. This mode of execution is similar to "batch" processing on other systems. The main advantage to placing commands in the background is that you can execute other commands from your terminal in the "foreground" while the background commands execute. Thus:

du /usr >diskuse&

determines the disk usage in the directory */usr*, a fairly time-consuming operation, without tying up your terminal. Note that the output is placed in the file *diskuse* by redirecting output with the greater-than symbol. Redirection is discussed in Section 3.6.1.

3.5.2 Syntax

The general syntax for commands is as follows:

cmd [*switches*][*arguments*] [*filename*] [...]

By convention, command names are lowercase. Switches, also called options, are flags that select various options available when executing the command. They are optional and usually *precede* other arguments and filenames. Switches consist of a dash prefix (–) and an identifying letter. For example, the **ls** command's **–l** switch (pronounced "minus ell") specifies a long directory listing and the command

ls –r

specifies a directory listing in reverse alphabetical order. In some cases, switches can be grouped to form a single switch argument. For example, the command

ls –rl

is really a combination of two switches, where the **–rl** switch selects the option that lists all files in the directory in both reverse alphabetical order and with the long format.

Sometimes multiple switches must be given separately, as in:

copy –a –v source destination

Here the **–a** switch tells the **copy** command to ask the user for confirmation before copying the *source* to the *destination*. The **–v** switch specifies the "verbose" option, which reports copying as it happens.

Other arguments, such as search strings, can also be given, as in:

grep´ string of text ´outfile

In the above example,

´string of text´

is a single argument and is the search string the **grep** command searches for in the file *outfile*. *filename* is the argument that specifies the name of a file required by the command.

Most commands are executable programs compiled by the C compiler or by some other language compiler. Some commands are executable command files called "shell procedures". Shell procedures are discussed in Chapter 4 of the XENIX *User's Guide*, "The Shell."

3.6 Input and Output

By default, XENIX assumes that terminal input comes from the terminal keyboard and output goes to the terminal screen. To illustrate typical command input and output, enter:

cat

This command now expects input from your keyboard. As input, it accepts as many lines of text as you enter until you press Ctrl-d as an end-of-file or end-of-transmission indicator.

For example, enter:

```
this is two linesRETURN
of inputRETURN
Ctrl-d
```

When you press Ctrl-d, input ends. The **cat** command immediately outputs each line as you enter it. Since output is sent to the terminal screen by default, that is where the two lines are sent. Thus, the complete session will look like this on your terminal screen:

```
$ cat
this is two lines
this is two lines
of input
of input
$
```

The flow of command input and output can be "redirected" so that input comes from a file instead of from the terminal keyboard and output goes to a file or lineprinter, instead of to the terminal screen. In addition, you can create "pipes" to make the output from one command become the input to another. Redirection and pipes are discussed in the next two subsections. When you use **cat** to send input to a file or pipe, the output is not sent until the Ctrl-d end-of-transmission indicator is entered.

3.6.1 Redirection

In XENIX a file can replace the terminal for either input or output. For example:

```
ls
```

displays a list of files on your terminal screen. But if you say:

```
ls > filelist
```

a list of your files is placed in the file *filelist* (which is created if it does not exist). The symbol for output redirection, the greater-than sign (>), means "put the output from the command into the following file, rather than display it on the terminal screen." As another example of output redirection, you can combine several files into one by capturing the output of **cat** in a file:

```
cat f1 f2 f3 > temp
```

The output append symbol (>>) operates very much like the output redirection symbol, except that it means "add to the end of". So:

```
cat file1 file2 file3 >> temp
```

means "concatenate *file1*, *file2*, and *file3* to the end of whatever is already in *temp*, instead of overwriting and destroying the existing contents." As with normal output redirection, if *temp* doesn't exist, it is created for you.

In a similar way, the input redirection symbol (<) means "take the input for a program from the following file, instead of from the terminal". Thus, you could make a script of editing commands and put them into a file called *script*. Then you could execute the commands in the script on a file by typing:

```
ed file < script
```

As another example, if you used an editor to prepare a letter in the file *letter.txt*, you could send it to several people with:

```
mail adam eve mary joe < letter.txt
```

3.6.2 Pipes

One of the major innovations of the XENIX system is the concept of a "pipe". A pipe is simply a way to connect the output of one command to the input of another, so that the two run as a sequence of commands called a pipeline.

For example:

```
sort frank.txt george.txt hank.txt
```

combines the three files named *frank.txt*, *george.txt*, and *hank.txt*, then sorts the output. Suppose that you want to then find all unique words in these files and view the result. You could enter:

```
sort frank.txt george.txt hank.txt > temp1
uniq < temp1 > temp2
more temp2
rm temp1 temp2
```

But this is more work than is necessary. What you want is to take the output of **sort** and connect it to the input of **uniq**, then take the output of uniq and connect it to **more**. You would use the following pipe:

```
sort frank.txt george.txt hank.txt | uniq | more
```

The vertical bar character (|) is used between the **sort** and **uniq** commands to indicate that the output from **sort**, which would normally have been sent to the terminal, is to be redirected from the terminal to the standard input of the **uniq** command, which in turn sends its output to the **more** command for viewing.

There are many other examples of pipes. For example, this command formats and paginates a list of your files in three columns:

```
ls | pr -3
```

The program **wc** counts the number of lines, words, and characters in its input, and **who** prints a list of users currently logged on, one per line. Thus, this command tells you the number of users who are logged in by counting the number of lines that comes from the **who** command:

```
who | wc -l
```

This command counts the number of files in the current directory:

```
ls | wc -l
```

Notice the difference in output between **wc -l** and **wc**. By default, **wc** tells you how many lines, word and characters, there are in the input. However, **wc -l** tells you only how many lines.

Any program that reads from the terminal keyboard can read from a pipe instead. Any program that displays output to the terminal screen can send input to a pipe. You can have as many elements in a pipeline as you wish.

Replace this Page
with Tab Marked:

Tasks

Chapter 4

Tasks

4.1 Introduction

This chapter explains how to perform common tasks with XENIX . The individual commands used to perform these tasks are discussed more thoroughly in the XENIX *Reference Manual.*

4.2 Gaining Access to the System

To use the XENIX system, you must first gain access to it by logging in. When you log in you are placed in your own personal working area. Logging in, changing your password, and logging out are described below.

4.2.1 Logging In

Before you can log in to the system, you must be given a system "account." Your name must be added to the user list, and you must be given a password and a mailbox.

Depending on how your system is administered, you may have to add your name to the user list yourself, or someone else may be assigned this task. If you must add your own account to the system, see the XENIX *Operations Guide* and **mkuser**(C) in the XENIX *Reference Manual* for more information. This section assumes your account has already been set up.

Normally, the system sits idle and the prompt "login:" appears on the terminal screen. If your screen is blank, or displays nonsense characters, press the **INTERRUPT** key a few times.

When the "login:" prompt appears, follow these steps:

1. Enter your login name and press **RETURN**. If you make a mistake, press **Ctrl-u** to start the line again. After you press **RETURN** the word "Password:" appears on your screen.

2. Enter your password carefully, then press **RETURN**. The letters do not appear on your screen as you enter, and the cursor does not move. If you make a mistake, press **RETURN** to restart the login procedure.

If you have entered your login name and password correctly the "prompt character" appears on the screen. This is usually a dollar sign($). The prompt tells you that the XENIX system is ready to acccept commands from the keyboard.

If you make a mistake, the system displays the message:

```
Login  incorrect
login:
```

If you get this message, follow the above procedure again. You must enter all the letters of your user name and password correctly before you are given access to the system; XENIX does not allow you to correct your mistakes when entering your password.

Depending on how your system is set up, after you log in you may see a "message of the day" that says something like "Welcome to XENIX", or an announcement that is of interest to all users.

4.2.2 Logging Out

The logout procedure is simple–all you need to do is press:

```
Ctrl-d
```

alone on a line. In general, **Ctrl-d** signifies the end-of-file in XENIX, and is often used within programs to signal the end of input from the keyboard. In such cases, **Ctrl-d** will *not* log you out; it will simply terminate input to a particular program if you are within that program. This means that it may sometimes be necessary to press **Ctrl-d** several times before you can log yourself out. For example, if you are in the **mail** program you must press **Ctrl-d** once to exit the mail program, then again to log out.

4.2.3 Changing Your Password

To prevent unauthorized users from gaining access to the system, each authorized user must have a password. When you are first given an account on a XENIX system you are assigned a password by the system administrator. Some XENIX systems require you to change your password at regular intervals. Whether yours does or not, it is a good idea to change your password regularly to maintain system security. This section tells you how to change your password.

Use the **passwd** command to change your password. Follow these steps:

1. Enter:

   ```
   passwd
   ```

   and press **RETURN**.

The following message appears:

```
Changing password for user
Old password:
```

2. Carefully enter your old password. It is not displayed on the screen. If you make a mistake, press **RETURN**. The message "Sorry" appears, then the system prompt. Begin again with step 1.

3. When you have entered your old password the message:

```
New password:
```

   appears. Enter your new password and press **RETURN**.

4. The message:

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

   appears. Enter your new password again. If you make a mistake, press **RETURN**. The message:

```
They don't match; try again
```

   appears, and you must begin again with step 1. When you have completed the procedure, the system prompt appears.

4.3 Configuring Your Terminal

On most systems, the primary monitor is already configured for use with XENIX. However, other terminals of various types may be connected to a XENIX system. In these cases it is important to know how to set terminal options and how to specify the terminal you are using. You may also want to change the standard configuration of the standard primary monitor. The following section discusses these topics.

4.3.1 Changing Terminals

The terminal type is displayed each time you log in. If you ever need to log in to XENIX on a terminal of a type different than the terminal you normally use, you may need to change your environment by editing the *.profile* file in your home directory. To do this, use a text editor to locate the **tset** line that looks something like this:

```
eval `tset -m :\?unknown -s -r -Q`
```

Change *unknown* in this line to the terminal type of your terminal. For example, if you normally log in on a vt100 terminal, change the line to:

```
eval 'tset -m :\?vt100 -s -r -Q'
```

Each time you log in you then see the message:

```
TERM = (vt100)
```

Press **RETURN** and the terminal type is set to vt100, or enter another terminal type and press **RETURN**.

4.3.2 Setting Terminal Options

There are a number of terminal options that can be set with the command **stty**. When entered without parameters, **stty** displays the current terminal settings. For example, typical output might look like this:

```
speed 9600 baud
erase '^h' ; kill '^u'
even -nl
```

Each of the above characteristics can be set with **stty**. For more information, see **stty**(C) in the XENIX *Reference Manual.*

4.4 Editing the Command Line

When you sit in front of a terminal and enter commands at your keyboard, there are a number of special keys that you can use. The most useful ones are described below.

4.4.1 Entering a Command Line

From your terminal, entering a command line consists of typing characters then pressing **RETURN**. Once you have pressed **RETURN** the computer reads the command line and commands specified on that line are executed. You may enter as many command lines as you want without waiting for commands to complete, because XENIX supports type-ahead of characters.

4.4.2 Erasing a Command Line

When entering commands, typing errors are bound to occur. To erase the current command line, press **Ctrl-u**.

4.4.3 Halting Screen Output

In many cases, you will be examining the contents of a file on the terminal screen. For longer files, the contents will often scroll off the screen faster than you can examine them. To temporarily halt a program's output to the terminal screen, press **Ctrl-s**. To resume output, press **Ctrl-q**.

4.5 Manipulating Files

File manipulation (creating, displaying, combining, copying, moving, naming, and deleting files), is one of the most important capabilities an operating system provides. The XENIX commands that perform these functions are described in the following sections.

4.5.1 Creating a File

To create a file and place text in it, use the editor **vi**, described in Chapter 2 of the XENIX *User's Guide*, "vi: A Text Editor." If for some reason you wish to create an empty file, enter:

```
> filename
```

where *filename* is the name of the empty file. In general, new files are created by commands as needed.

4.5.2 Displaying File Contents

The **more** command displays the contents of a file one screenful at a time. It has the form:

more *options filename*

more is useful for looking at a file when you do not want to make changes to it. For example, to display the contents of the file *memos*, enter:

```
more memos
```

more can be invoked with options that control where the display begins, and how the file is displayed.

These options include:

+linenumber
: Begins the display at the line in the file designated by *linenumber*.

+/*text*
Begins the display two lines before *text*, where *text* is a word or number. If *text* is two or more words, they must be enclosed in double quotation marks.

−c Redraws the screen instead of scrolling.

−r Displays control characters, which are normally ignored by **more**.

To begin looking at the file *memo* at the first occurrence of the words "net gain", for example, enter:

    more +/"net gain" memo

If the file is more than one screenful long, the percentage of the file that remains is displayed on the bottom line of the screen. To look at more of the file, use the following scrolling commands:

RETURN Scrolls down one line.

d Scrolls down one-half screen.

SPACE Scrolls down a full screen.

*n*SPACE Scrolls down *n* lines.

. Repeats the previous command.

You cannot scroll backward, toward the beginning of the file.

You can search forward for patterns in **more** with the slash (/) command.

For example, to search for the pattern "net gain", enter:

    /net gain/

and press **RETURN**. **more** displays the message:

    ...skipping

at the top of the screen, and scrolls to a location two lines above "net gain."

If you are looking at a file with **more** and decide you want to change the file, you can invoke the **vi** editor by pressing:

    v

See Chapter 2 of the XENIX *User's Guide*, "vi: A Text Editor," for information on using **vi**.

more quits automatically when it reaches the end of a file. To exit **more** before the end of a file, enter:

```
q
```

The **head** and **tail** commands display the first and last ten lines of a file, respectively. They are useful for checking the contents of a particular file.

For example, to look at the first ten lines of the file *memo*, enter:

```
head memo
```

You can also specify how many lines the **head** and **tail** commands display. For example:

```
tail -4 memo
```

displays the last four lines of *memo*.

The **cat** command also displays the contents of a file. **cat** scrolls the file until you press **Ctrl-s** to stop it. Pressing **Ctrl-q** will continue the scrolling. **cat** stops automatically at the end of a file. If you wish to stop the display before the end of the file, press **INTERRUPT**. To display the contents of one file, enter:

```
cat file1
```

To display the contents of more than one file, enter:

```
cat file1 file2 file3
```

4.5.3 Combining Files

The **cat** command is frequently used to combine files into some other new file.

Thus, to combine the two files named *file1* and *file2*, into a new file named *bigfile*, enter:

```
cat file1 file2 >bigfile
```

Note here that we are putting the contents of the two files into a new file with the name *bigfile*. The greater than sign (>) is used to *redirect* output of the **cat** command to the new file.

You can also use **cat** to append one file to the end of another file. For example, to append *file1* to *file2*, enter:

```
cat file1 >> file2
```

The contents of *file1* are added to *file2*. *file1* still exists as a separate entity.

4.5.4 Moving a File

The **mv** command moves a file into another file in the same directory, or into another directory.

For instance, to move a file named *text* to a new file named *book*, enter:

```
mv text book
```

After this move is completed, no file named *text* will exist in the working directory, because the file has been renamed *book*.

To move a file into another directory, give the name of the destination directory as the final name in the **mv** command. For instance, to move *file1* and *file2* into the directory named */tmp*, enter:

```
mv file1 file2 / tmp
```

The two files you have moved no longer exist in your working directory, but now exist in the directory */tmp*. The above command has exactly the same effect as entering the following two commands:

```
mv file1 /tmp/file1
mv file2 /tmp/file2
```

The **mv** command always checks to see if the last argument is the name of a directory and, if so, all files designated by filename arguments are moved into that directory.

4.5.5 Renaming a File

To rename a file, simply "move" it to a file with the new name: the old name of the file is removed. Thus, to rename the file *anon* to *johndoe*, enter:

```
mv anon johndoe
```

Note that moving and renaming a file are essentially identical operations.

4.5.6 Copying a File

There are two forms of the **cp** command: one in which files are copied into a directory, and another in which a file is copied to another file. Thus, to copy three files into a directory named *filer*, enter:

```
cp file1 file2 file3 filer
```

In the above command, three files are copied into the directory *filer;* the original versions still reside in the working directory. Note that the filenames are identical in the two directories. Like the **mv** command, **cp** always checks to see if the last argument is the name of a directory, and, if so, all files designated by filename arguments are copied into that directory.

To create two copies of a file in your own working directory, you must rename the copy. To do this, the copy command can be invoked as follows:

```
cp file filecopy
```

After the above command has executed, two files with identical contents reside in the working directory. To learn how to copy directories, see section 4.6.7, "Copying a Directory", later in this chapter.

4.5.7 Deleting a File

To delete or remove files, enter:

```
rm file1 file2
```

In the above command, the files *file1* and *file2* are removed from your working directory.

The command:

```
rm –i file1 file2
```

allows you to interactively remove files by asking you if you really want to delete each of the files *file1* and *file2.* If you press *y* followed by a **RETURN**, the given file is removed; if you press *n* the file is left untouched. This command is useful when cleaning up a directory that contains many files.

4.5.8 Finding Files

The **find** command searches for files that have a specified name. **find** is useful for locating files that have the same name, or just for finding a file if you have forgotten which directory it is in.

The command has the form:

find *pathname* –name *filename* –print

The *pathname* is the pathname of the directory you want to search. **find** searches recursively, that is, it starts at the named directory and searches downward through all files and subdirectories under the directory specified in *pathname*.

The **–name** option indicates that you are searching for files that have a specific *filename*. (There are other search conditions you can use with **find**; see **find**(C) in the XENIX *Reference Manual*.)

filename is the name of the file you are searching for.

The **–print** option indicates you want to print the pathnames of all the files that match *filename* on your terminal screen. You may direct this output to a file instead of your screen with the output redirection symbol, >. (There are other actions that can be performed with **find**, such as removing and moving files; see **find**(C) in the XENIX *Reference Manual*.) For example, the following command finds every file named *memo* in the directory */usr/joe* and all its subdirectories:

```
find /usr/joe –name memo –print
```

The output might look like this:

```
/usr/joe/memo
/usr/joe/accounts/memo
/usr/joe/meetings/memo
/usr/joe/mail/memo
```

4.5.9 Linking a File to Another File

The **ln** command joins two files in different directories so that when the file is changed in one directory, it is also changed in the other directory. This can be useful if several users need to share information, or if you want a file to appear in more than one directory. This command has the form:

ln *file newfile*

where *file* is the original file, and *newfile* is the new, linked file. For example, the following command links *memos* in */usr/joe* to *joememos* in */usr/mary:*

ln /usr/joe/memos /usr/mary/joememos

Whenever */usr/joe/memos* is updated, the file */usr/mary/joememos* is also changed.

When you link files a name is associated with an *inode*. An inode specifies a unique set of data on the disk. One or more names can be associated with this data. Thus, the above command assures that the files *dir1/file1* and *dir2/file2* have identical contents.

There are three things that are not immediately obvious about linking files:

1. Linking large sets of files to other parallel files can save a considerable amount of disk space.

2. Linking files used by more than one person is risky, because any party can alter the file and thus affect the contents of all files linked to it.

3. Removing a file from a directory does not remove other links to the file. Thus the file is not truly deleted from the system. For example, if you delete a file that has 4 links, 3 links remain.

For more information about linking see **ln**(C) in the XENIX *Reference Manual.*

4.6 Manipulating Directories

Because of the hierarchical organization of the file system, there are many directories and subdirectories in the XENIX system. Within the file system are directories for each user of the system. Within your user directory you can create, delete, and copy directories. Commands that let you manipulate directories are described in the following sections.

4.6.1 Printing the Name of Your Working Directory

All commands are executed relative to a "working" directory. The name of this directory is given by the **pwd** command, which stands for "print working directory." For instance, if your current working directory is */usr/joe*, when you enter:

pwd

you will get the output:

```
/usr/joe
```

You should always think of yourself as residing "in" your working directory.

4.6.2 Listing Directory Contents

You can list the contents of a directory with the **lc** command. This command sorts the names of files and directories in a given directory, and lists them in columns. If no directory name is given, **lc** lists the contents of the current directory. The **lc** command has the form:

lc *options name*

For example, to list the contents of the directory *work*, enter:

```
lc work
```

Your output might look like this:

```
accounts    meetings    notes
mail        memos       todo
```

If no *name* is specified, **lc** lists the contents of the current directory. If *accounts* is the current directory, for example, the command:

```
lc
```

lists the names of the files and subdirectories in that directory.

The following options control the sort order and the information displayed by the **lc** command:

−a Lists all files in the directory, including the "hidden" files (filenames that begin with a dot, such as *.profile* and *.mailrc*).

−r Lists names in reverse alphabetical order.

−t Lists names in order of last modification, the latest (most recently modified) first. When used with the **−r** option, lists the oldest first.

–R Lists all files and directories in the current directory, plus each file and directory *below* the current one. The "R" stands for "recursive."

–F Marks directories with a backslash(\) and executable files with an asterisk (*).

The ls command works much like the **lc** command except that it lists files in vertical, rather than collumnar, form. The **ls –l** command gives a "long" listing of a directory, producing an output that might look something like this:

```
total 501
drwxr-x---  2 boris grp1 272  Apr 5 14:33 dir1
drwxr-x---  2 enid  grp1 272  Apr 5 14:33 dir2
drwxr-x---  2 iris  grp1 592  Apr 6 11:12 dir3
-rw-r-----  1 olaf  grp2 282  Apr 7 15:11 file1
-rw-r-----  1 olaf  grp2 72   Apr 7 13:50 file2
-rw-r-----  1 olaf  grp2 1403 Apr 1 13:22 file3
```

Reading from left to right, the information given for each file or directory includes:

- Permissions
- Number of links
- Owner
- Group
- Size in bytes
- Time of last modification
- Filename

The information in this listing and how to change permissions are discussed below in Section 4.8, "Using File and Directory Permissions."

For more information about listing the contents of a directory, see ls(C) in the XENIX *Reference Manual*.

4.6.3 Creating a Directory

To create a subdirectory in your working directory, use the **mkdir** command. For instance, to create a new directory named *phonenumbers,* simply enter:

```
mkdir phonenumbers
```

After this command has been executed, a new empty directory will exist in your working directory.

4.6.4 Removing a Directory

To remove a directory located in your working directory, use the **rmdir** command. For instance, to remove the directory named *phonenumbers* from the current directory, simply enter:

```
rmdir phonenumbers
```

Note that the directory *phonenumbers* must be *empty* before it can be removed; this prevents catastrophic deletions of files and directories. If you want to live dangerously, it is possible to recursively remove the contents of a directory using the **rm** command, but that will not be explained here. See **rm**(C) in the XENIX *Reference Manual* for more information.

4.6.5 Renaming a Directory

To rename a directory, use the **mv** command. For instance, to rename the directory *little.dir* to *big.dir,* enter:

```
mv little.dir big.dir
```

This is a simple renaming operation; no files are moved.

4.6.6 Moving a Directory

The **mv** command also moves directories. This command has the form:

mv *oldirectory newdirectory*

where *newdirectory* is a directory that already exists.

For example, to move the directory */usr/joe/accounts* into */usr/joe/overdue* enter:

```
mv /usr/joe/accounts /usr/joe/overdue
```

The new pathname of */usr/joe/accounts* is */usr/joe/overdue/accounts*.

4.6.7 Copying a Directory

The **copy** command copies directories. This command has the form:

copy *options olddir newdir*

To copy all the files in the directory */usr/joe/memos* into */usr/joe/notes* enter:

```
copy /usr/joe/memos /usr/joe/notes
```

The files in */usr/joe/memos* are copied into */usr/joe/notes*. The **copy** command has the following options:

–l Links the copied files to the original.

–m Gives the copied files the same modification dates as the original files.

–r Copies the directory recursively, i.e., copies all the directories under the named directory.

4.7 Moving in the File System

When using the XENIX system, it helps to imagine a large tree structure of files and directories. Each directory should be thought of as a place that you can move into or out of. At all times you are "someplace" in the tree structure. This place is called either your working directory or current directory. The commands used to find out where you are and to move around in the tree structure are discussed below.

4.7.1 Finding Out Where You Are

Your current location in the file system is the name of the working directory. You can find out this name by using the **pwd** command, which stands for "print working directory." For example, if you are in the directory */usr* then entering the command:

```
pwd
```

prints out the name:

```
/usr
```

4.7.2 Changing Your Working Directory

Your working directory represents your location in the file system: it is "where you are" in XENIX . To alter this location in the XENIX file system, use the change directory (**cd**) command:

```
cd
```

This changes your working directory to your home directory. To move to any other directory, specify that directory as an argument to **cd**.

For instance, the following command:

```
cd /usr
```

moves you to the */usr* directory. Because you are always "in" your working directory, changing working directories is much like "traveling" from directory to directory.

To move up one directory from your current directory, enter:

```
cd ..
```

For example, the above command would move you from the directory */usr/joe/work* to */usr/joe*. Similarly, the command:

```
cd ../..
```

would move you from the directory */usr/joe/work* to */usr*, moving you up *two* directories.

4.8 Using File and Directory Permissions

The XENIX system allows the file owner to restrict access to files and directories, limiting who can read, write and execute files owned by him. To determine the permissions associated with a given file or directory, use the **ls -l** command. The output from the **ls -l** command should look something like this:

```
total 501
drwxr-x---  2 boris grp1 272  Apr 5 14:33 dir1
drwxr-x---  2 enid  grp1 272  Apr 5 14:33 dir2
drwxr-x---  2 iris  grp1 592  Apr 6 11:12 dir3
-rw-r-----  1 olaf  grp2 282  Apr 7 15:11 file1
-rw-r-----  1 olaf  grp2 72   Apr 7 13:50 file2
-rw-r-----  1 olaf  grp2 1403 Apr 1 13:22 file3
```

Permissions are indicated by the first ten characters of the output. The permissions for dir1, the first file in the above list, are:

```
drwxr-x---
```

The first character indicates the type of file and must be one of the following:

-	Indicates an ordinary file.
d	Indicates a directory.
c	Indicates a character special device such as a lineprinter or terminal.
b	Indicates a block special device such as a hard or floppy disk.
n	Indicates a name special file (i.e., a semaphore used for controlling access to some resource).
s	Indicates a shared data file.
p	Indicates a named pipe.

From left to right, the next nine characters are interpreted as three sets of three permissions each. Each respective set of three indicates the following permissions:

- Owner permissions
- Group permissions
- All other user permissions

Within each set, the three characters indicate permission to read, to write, and to execute the file as a command, respectively. For a directory, "execute" permission means permission to search the directory for any included files or directories.

Ordinary file permissions have the following meanings:

r	The file is readable.
w	The file is writeable.
x	The file is executable.
–	The indicated permission is not granted.

For directories, permissions have the following meanings:

r	Files can be listed in the directory; the directory must also have "x" permission.
w	Files can be created or deleted in the directory; as with "r", the directory itself must also have "x" permission.
x	The directory can be searched. A directory must have "x" permission before you can move to it with the **cd** command (i.e., **cd** to it), access a file within it, or list the files in it. Remember that a user must have "x" permission to do anything useful to the directory.

The following are some typical directory permission combinations:

d---------	No access at all. This is the mode that denies access to the directory to a class of users.
drwx------	Allows access by only the owner to use **lc**, create files, delete files, access files (subject to file permissions), and **cd** to the directory. This is the typical permission for the owner of a directory.
drwxr-x---	Allows access by members of the group to use **lc**, and access files subject to file permissions. Group members can **cd** to this directory, but cannot create or delete files in it. This is the typical permission an owner gives to others who need access to files in his directory.
drwx--x--x	With these permission settings users other than the owner cannot use **lc** but can **cd** to the directory. Other

users can only access a file within this directory by its exact name; they cannot use special characters. Files cannot be created or deleted in the directory by anyone except the owner. This mode is rarely used, but can be useful if you want to give someone access to a specific file in a directory without permitting access to other files in the same directory.

This chapter discusses ordinary files, executable files, and directories only. For information about other types of files, see **ls** (C) in the XENIX *Reference Manual*.

4.8.1 Changing Permissions

The **chmod** command changes the read, write, execute, and search permissions of a file or directory. This command is useful if you have created a file in one mode, but want to give others permission to read, write or execute it.

The **chmod** command has the form:

chmod *instruction filename*

The *instruction* segment of the command indicates which permissions you want to change for which class of users. There are three classes of users, and they are are indicated as follows:

u User, the owner of the file or directory

g Group, the group the owner of the file belongs to

o Other, all users of the system

a All classes of users

There are three types of permissions, as follows:

r Read, which allows permitted users to look at but not change or delete the file.

w Write, which allows permitted users to change or even delete the file.

x Execute, which allows permitted users to execute the file as a command.

For example, assume *file1* exists with the following permissions:

    -rw-r-----

In the above example, the owner of the file has read and write permission, group members have read permission, and others have no access at all.

To give *file1* read permission for *all* classes of users, enter:

    chmod a+r file1

In the instruction segment of the command (a+r) the "a" stands for "all."

The resulting permissions are:

    -rw-r--r--

For *file1* with the attributes:

    -rw-------

The following command gives write and execute permissions to members of a group only:

    chmod g+wx file1

This command would alter the permission attributes so they look like this:

    -rw--wx---

To remove write and execute permission by the user (owner) and group associated with *file1*, enter:

    chmod ug-wx file1

4.8.2 Changing Directory Search Permissions

Directories also have an execute permission. This attribute signifies search permission, rather than execute permission, since directories cannot be executed. If this permission is denied to a particular user, then that user cannot even list the names of the files in the directory.

For example, assume that the directory *dir1* has the following attributes:

```
drwxr-xr-x
```

To remove search permission for other users to examine *dir1*, enter:

```
chmod o-xr dir1
```

The new attributes for *dir1* are:

```
drwxr-x---
```

4.9 Processing Information

In many cases, files will contain information that you may want to process. Various utility programs exist on XENIX to process information. A set of these programs and their uses are described in the following sections.

4.9.1 Comparing Files

To compare two text files use the **diff** command to print out those lines that differ between the files that you specify.

For example, suppose that a file named *men* has the contents:

```
Now is the time for all good men to
Come to the aid of their party.
```

and that a file named *women* has the following contents:

```
Now is the time for all good women to
Come to the aid of their party.
```

If this is the case, then the command:

```
diff men women
```

produces the following results:

```
1c1
< Now is the time for all good men to
---
> Now is the time for all good women to
```

A three-way difference listing can be created with the **diff3** command. For information about **diff3** see **diff3**(C) in the XENIX *Reference Manual.*

4.9.2 Echoing Arguments

The **echo** command echos arguments to the standard output. For example, entering:

```
echo hello
```

outputs:

```
hello
```

on the terminal screen. To output several lines of text, surround the echoed argument in double quotation marks and press **RETURN** between lines. A secondary prompt will appear until you enter the final double quotation mark. For example, enter:

```
echo "Now is the time
For all good men
To come to the
Aid of their party."
```

This will output:

```
Now is the time
For all good men
To come to the
Aid of their party.
```

echo is particularly useful if you should ever program in the shell command language. For more information about the shell, see Chapter 4, "The Shell", XENIX *User's Guide*.

4.9.3 Sorting a File

One of the most useful file processing commands is **sort**. By default, **sort** sorts the lines of a file according to the ASCII collating sequence (i.e., it alphabetizes them).

For example, to sort a file named *phonelist*, enter:

```
sort phonelist
```

In the above case, the sorted contents of the file are displayed on the screen. To create a sorted version of *phonelist* named *phonesort*, enter:

```
sort phonelist >phonesort
```

Note that **sort** is useful for sorting the output from other commands. For example, to sort the output from execution of a **who** command, enter:

```
who | sort >whosort
```

This command takes the output from **who**, sorts it, and then sends the sorted output to the file *whosort*.

A wide variety of options are available for **sort**. For more information, see **sort**(C) in the XENIX *Reference Manual*.

4.9.4 Searching for a Pattern in a File

The **grep** command selects and extracts lines from a file, printing only those lines that match a given pattern. For example, to print out all lines in a file containing the word "tty38", enter:

```
grep 'tty38' file
```

In general, you should always enclose the pattern you are searching for in single quotation marks ('), so that special metacharacters are not expanded unexpectedly by the shell.

As another example, assume that you have a file named *phonelist* that contains a name followed by a phone number on each line. Assume also that there are several thousand lines in this list. You can use **grep** to find the phone number of someone named Joe, whose phone number prefix is 822, as follows:

```
grep 'joe' phonelist | grep '822-' >joes.number
```

grep finds all occurrences of lines containing the word "joe" in the file *phonelist*. The output from this command is then filtered through another **grep** command, which searches for an "822-" prefix, thus removing any unwanted joes. Finally, assuming that a unique phone number for joe exists with the "822–" prefix, that name and number are placed in the file *joes.number*.

For more information about **grep**, its relatives **fgrep** and **egrep**, and the types of patterns it can be used to search for (called regular expressions) see **grep** (C) in the XENIX *Reference Manual*.

4.9.5 Counting Words, Lines, and Characters

wc is a utility for counting words in a file. The letters "wc" stand for word count. Words are presumed to be separated by punctuation, spaces, tabs, or newlines. **wc** also counts characters and lines; all three counts are reported by default.

For example, to count the number of lines, words, and characters in the file *textfile*, enter:

```
wc  textfile
```

Typical output describing lines, words and characters might be:

```
    4432   18188    97808  textfile
```

To specify a count of characters, words, or lines only, you must use an appropriate mnemonic switch.

To illustrate, examine the following three commands and the output produced by each:

```
wc  -c  textfile
  97808  textfile

wc  -w  textfile
  18188  textfile

wc  -l  textfile
    4432  textfile
```

The first example prints out the number of characters in *textfile*, the second prints out the number of words, and the third prints out the number of lines.

4.9.6 Delaying a Process

The **at** program allows you to set up commands to be executed at a specified time. It is useful if you want to execute a command when you are not planning to be at your terminal, or even logged in.

The **at** command accepts standard input and has the form:

cat *file* | at *time day*

file is the name of the file that contains the command or commands to be executed. *time* is the time of day, in digits, followed by "am" or "pm." One- and two-digit numbers are interpreted as hours, three- and four-digit numbers as hours and minutes. More than four digits is not permitted.

day is optional. It is either a month name followed by a day number, or a day of the week. If no *day* is specified, the command will be executed today.

For example, if you want to find out what processes are running at 10 pm on Tuesday, place the following line in a file named *use:*

```
ps -a > /usr/myname/use
```

(See Chapter 2, "vi: A Text Editor," of the XENIX *User's Guide* for information on creating and inserting text into files.)

After you have written out the file and returned to command level, enter:

```
cat use | at 10pm tues
```

Press RETURN. The XENIX prompt reappears and you may continue working. At 10 pm on Tuesday, XENIX will execute **ps -a** and place the output in the file *use*. **at** is unaffected by logging out.

To check what files you have waiting to be processed, use the **at -l** command. **at -l** lists the files the user owns to be processed, along with the following information:

- The file's ID number
- The command invoking the file (**at** or **batch**).
- The date and time the file will be processed

To cancel an **at** command, first check the list of files you have to be processed and note the file ID number. Then use the **at -r** command to remove the file or files from the list.

The **at -r** command has the form:

```
at -r number
```

For example:

```
at -r 504510300.a
```

removes file number 504510300.a, canceling whatever commands were included in that file. A user can only remove his own files.

4.10 Controlling Processes

In XENIX, several processes can run at the same time. For example, you may run the **sort** program on a file in the "background", and edit another file in the "foreground" while the **sort** program is running. Things that you directly control at your keyboard are called foreground processes. Other processes, which you can initiate but that you otherwise have little control over, are called background processes. At any one time you can have only one foreground process executing, but multiple background processes may execute simultaneously. Controlling foreground and background processes is the subject of this section.

4.10.1 Placing a Process in the Background

Normally, commands sent from the keyboard are executed in strict sequence; one command must finish executing before the next can begin. Executing commands of this type are called foreground processes. A background process, in contrast, need not finish executing before you give your next command. Background commands are especially useful for commands that may take a long time to complete.

To place a process in the background, enter an ampersand (&) at the end of the command. For example, to count the number of words in several large files while simultaneously continuing with whatever else you have to do, enter:

```
wc file1 file2 file3 >count&
```

Output is collected in the file *count*. If output were not put in *count*, it would appear on the screen at unpredictable times as you continue with your work.

When processes are placed in the background, you lose control of them as they execute. For instance, entering **INTERRUPT** does *not* abort a background process. You must use the **kill** command, described in the following section, instead.

4.10.2 Killing a Process

To stop execution of a foreground process, press your terminal's **INTERRUPT** key. This kills whatever foreground command is currently running. To kill all your processes executing in the background, enter:

```
kill 0
```

To kill only a specified process executing in the background, first enter:

```
ps
```

ps displays the Process Identification Numbers (PIDs) of your existing processes, for example:

```
  PID TTY TIME  CMD
 3459 03   0:15 -sh
 4831 03   1:52 cc program.s
 5185 03   0:00 ps
```

In the above example, you might enter:

```
kill 4831
```

where 4831 is the PID of the process that you want killed.

Note

Killing a process associated with the **vi** editor may leave the terminal in a strange mode. Also, temporary files that are normally created when a command starts, and then deleted when the command finishes, may be left behind after a **kill** command. Temporary files are normally kept in the directory */tmp*. This directory should be checked periodically and old files deleted.

4.11 Getting Status Information

Because XENIX is a large, self-contained computing environment, there are many things that you may want to find out about the system itself, such as who is logged in, how much disk space there is, what processes are currently running. This section explains the types of information available from the system and how to get it.

4.11.1 Finding Out Who is on the System

The **who** command lists the names, terminal line numbers, and login times of all users currently logged on to the system. For example, enter:

```
who
```

This command produces something like the following output on your terminal screen:

```
arnold   tty02  Apr  7  10:02
daphne   tty21  Apr  7  07:47
elliot   tty23  Apr  7  14:21
ellen    tty25  Apr  7  08:36
gus      tty26  Apr  7  09:55
adrian   tty28  Apr  7  14:21
```

The **finger** command provides more detailed information, such as office numbers and phone extensions. For more information, about using **finger** see **finger**(C) in the XENIX *Reference Manual.*

4.11.2 Finding Out What Processes Are Running

Because commands can be placed in the background for processing, it is not always obvious which processes you are responsible for. The **ps** command stands for "process status" and displays information about currently running processes associated with your terminal. For instance, the output from a **ps** command might look like this:

```
  PID TTY  TIME  CMD
10308  38  1:36  ed  chap02.man
   49  38  0:29  -sh
11267  38  0:00  ps
```

The PID column gives a unique process identification number that can be used to kill a particular process. The TTY column shows the terminal that the process is associated with. The TIME column shows the cumulative execution time for the process. Processes can be killed using the **kill** command. See section 4.10.2, "Killing a Process," for information on how to use the **kill** command.

To find out all the processes running on the system, use the **a** option:

```
ps -a
```

To find out about the processes running on a terminal other than the terminal you are using, use the **-t** option and specify the terminal number. For example, to find out what processes are associated with terminal 13, enter:

    ps -t13

For more information about **ps** and its options, see **ps**(C) in the XENIX *Reference Manual.*

4.11.3 Finding Out Lineprinter Information

You can find out the status of files you requested printed with the **lpstat** command. **lpstat** displays information on an individual file or on all your files waiting to be printed.

To find out the status of one file, you need to know the "request ID." When you make print requests using the **lp** command, you find a request ID displayed on your terminal screen. The request ID has the form:

printer–idnumber

printer is the name of the printer your file will be printed on (check with your system manager for the names of printers available to you) and *idnumber* is a unique number identifying your file.

To find out the status of a particular file, enter:

lpstat *request ID*

lpstat responds by displaying the date and time you made your print request and the number of characters remaining to be printed.

To find out the status of all your files waiting to be printed on the lineprinters, enter:

    lpstat

lpstat responds by displaying the request IDs and status information for all your files.

You can find out what files are waiting to be printed on a particular printer by using **lpstat** with the **-p** option. This command has the form:

lpstat -p *printer*

lpstat responds by printing the request IDs and status information for all the files waiting to be printed on the named printer.

For more information on **lpstat** and its options, see **lpstat**(C) in the XENIX *Reference Manual.*

4.12 Using the Lineprinter

The XENIX lineprinter commands are easy to use and give you great flexibility when you want to print a file. With a few simple commands, you can print multiple copies of a file, cancel a print request, or ask for a special option on a particular printer. Since the XENIX lineprinter system is designed to be easily adapted to many different environments, check with your system manager to find out what lineprinters and printer options are available to you.

4.12.1 Printing Files: lp

To print copies of your files, you can use either the **lp** command or **lpr**. These commands are equivalent. The examples in this section use **lp**.

For example, to print one copy of a file named *memo*, enter:

```
lp memo
```

You can request that several files be printed. For example, to print three files named *memo,report,* and *letter*, enter:

```
lp memo report letter
```

When you make print requests, **lp** responds by displaying your "request ID" on your terminal screen. Your request ID might look like this:

```
pr4-532
```

The first part (pr4) is the name of the printer your file will be printed on. The second part (532) identifies your file. Should you later wish to cancel your print request or check its status, you will find it useful to remember your request ID. For more information on these tasks, see sections 4.12.3, "Canceling a Print Request," and 4.12.4, "Finding Out the Status of A Print Request: lpstat."

One copy of each file you named will be printed on the default destination printer on your system.

You can use **lp** with pipes and other commands. The command to paginate a file is **pr**. To paginate and print a file named *textfile*, enter:

```
pr textfile | lp
```

To sort, paginate, and print a file named *datafile*, enter:

```
sort datafile | pr | lp
```

4.12.2 Using lp Options

The **lp** command has several options to help you control the output from your printer.

You can specify the number of copies you want printed by using the number option, **-n**. For example, to print two copies of a file named *report*, enter:

```
lp report -n2
```

Another option, **-d**, specifies your file's destination, that is, which printer your file will be printed on. Check with your system manager for the names of the printers available to you. To have two copies of a file named *report* printed on a printer named *quick*, enter:

```
lp report -n2 -dquick
```

Other useful options include:

- **-c** Makes a copy of the files you are printing. This prevents you from inadvertently removing or changing the file before it is printed.
- **-m** Sends you mail telling you your file has been printed.
- **-o** Specifies printer options. For example, you may be able to request that your document be printed using 12 pitch type. Check with your system manager to see what options are available for each printer or groups of printers on your system.
- **-r** Removes your files after printing.

For more information on options available for the **lp** command, see **lp**(C) in the XENIX *Reference Manual*.

4.12.3 Cancelling a Print Request: cancel

You can cancel a print request. For example, to stop printing a file with a request ID of *laser-245*, enter:

```
cancel laser-245
```

The **cancel** command immediately stops the file from being printed, even if the printer has already begun the print request.

You can also use the **cancel** command to stop whatever is currently printing on a particular printer. With **cancel**, you can easily free up a printer to print the next file, or stop it from printing strange output without contacting your system manager.

For example, to cancel whatever file is currently printing on a printer named *slow*, enter:

```
cancel slow
```

If the file did not belong to you, mail will automatically be sent to the file's owner reporting that the print request was canceled.

4.12.4 Finding Out the Status of A Print Request: lpstat

To find out the status of your files waiting to be printed, enter:

```
lpstat
```

lpstat gives output similar to:

```
prt1-121     chrisw    450    Dec 15 09:30
laser-450    chrisw   4968    Dec 15 09:46
```

The first column shows the request ID for each of your files being printed; the second column is your login name. In the third column, the number of characters to be printed is shown, and the fourth column lists the dates and times you made your print requests.

To learn the status of a particular file, use the **lpstat** command with your request ID. For example, to find out the status of a file with the request ID of *daisy-256*, enter:

```
lpstat daisy-256
```

lpstat displays the status of that file only.

You can also request the status of various printers on your system by using the **-p** option or by giving the name of the particular printer you are interested in.

To find out the status of all the printers on your system, enter:

```
lpstat -p
```

To find out the status of a printer named *quick*, enter:

```
lpstat -pquick
```

lpstat displays the request ID and status information for each file currently waiting to be printed on the printer named *quick*.

For more information on **lpstat** and its options, see **lpstat**(C) in the XENIX *Reference Manual*.

4.13 Communicating with Other Users

Because the XENIX system supports multiple users, it is very convenient to communicate with other users of the system. The various methods of communication are described below.

4.13.1 Sending Mail

mail is a system-wide facility that permits you and other system users to send and receive mail. To send mail to another user on the system, enter:

```
mail joe
```

where *joe* is the name of any user of the system. Following entry of the command, you enter the actual text of the message you want to send. Entry of text is terminated by pressing **Ctrl-d**.

A complete session at the terminal might look like this on your screen:

```
mail -s "Meeting today" joe
There will be a meeting at 2:00 today
to review recent problems with the
new system.
```
Ctrl-d

Note the use of the **-s** switch to specify the subject of the message.

For practice, send mail to yourself. (This is not as strange as it might sound – mail to yourself is a handy reminder mechanism.) You can also send a previously prepared letter, and you can send mail to a number of people all at once. For more details, see Chapter 3, "Mail", of the XENIX *User's Guide*, and **mail**(C) in the XENIX *Reference Manual*.

4.13.2 Receiving Mail

When you log in, you may sometimes get the message:

```
you have mail
```

To read your mail, enter:

```
mail
```

A heading for each message is then displayed on your terminal screen. When you press **RETURN**, the contents of the first message are displayed. Subsequent messages are displayed, one message at a time, most recent message first, each time you press **RETURN**.

After each message is displayed, **mail** waits for you to tell it what to do with the message. The two basic responses are **d**, which deletes the message, and **RETURN**, which does not delete the message (so it will still be there the next time you read your mailbox). To exit mail, enter: **q**, for "quit." Other responses are described in the XENIX *Reference Manual* under **mail**(C).

4.13.3 Writing to a Terminal

To write directly to another user's terminal, use the **write** command. For example, to write to joe's terminal, enter:

```
write joe
```

After you have executed the command by pressing **RETURN**, each subsequent line that you enter is displayed both on your own terminal screen and on joe's. To terminate the writing of text to joe, enter a **Ctrl-d** alone on a line.

The procedure for a two-way write is for each party to end each message with a distinctive signal, normally (o) for "over"; when a conversation is about to be terminated use the signal (oo) for "over and out."

4.14 Using the System Clock and Calendar

There are several XENIX commands that will tell you the date and time, or display a calendar for any month or year you choose. The following sections explain these commands.

4.14.1 Finding Out the Date and Time

The **date** command displays the time and date. Enter:

```
date
```

The date and time are displayed.

4.14.2 Displaying a Calendar

The **cal** command displays the calendar of any month or year you specify. This command has the form:

cal *month year*

For example, to display the calendar for March, 1952 enter:

```
cal 3 1952
```

The result is:

```
         March 1952

 S  M Tu  W Th  F  S
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

The month must always be expressed as a digit. To display the calendar for an entire year, leave out the month. The year must always be expressed in full; the command "cal 85" displays the calendar for the year 85, not 1985.

4.15 Using the Automatic Reminder Service

An automatic reminder service is normally available for all XENIX users. Once each day, XENIX uses the **calendar** command to examine each user's home directory for a file named *calendar*, the contents of which might look something like this:

```
1/23 David's wedding
2/9  Mira's birthday
3/30 Paul's birthday
4/27 Meeting at 2:00
9/1  Karen's birthday
10/3 License renewal
```

calendar examines each line of the calendar file, extracting from the file those lines containing today's and tomorrow's dates. These lines are then mailed to you to remind you of the specified events.

4.16 Using Another User's Account

You can easily access another user's files, regardless of the permission settings, with the **su** command. The **su** procedure resembles logging in, and you must know the other user's password.

For example, to become user Joe, enter:

```
su joe
```

and press **RETURN**. When the password prompt appears, enter Joe's password. To cancel the effect of the **su** command and return to your own account, press **Ctrl-d**.

4.17 Calculating

The **bc** command invokes an interactive desk calculator that can be used as if it were a hand-held calculator. A typical session with **bc** is shown below. Comments explain what action is performed after each input line.

```
/* This is a comment */
123.456789 + 987.654321      /* Add and output        */
1111.111110
9.0000000 - 9.0000001 /* Subtract and output */
-.0000001
64/8        /* Divide and output      */
8
1.12345678934 * 2.3     /* Note precision    */
2.58395061548
19%4        /* Find remainder  */
3
3^4 /* Exponentiation   */
81
2/1*2       /* Note precedence        */
4
2/(1*2)     /* Note precedence again */
1
x = 46.5  /* Assign value to x       */
y = 52.5  /* Assign value to y       */
x + y + 1.0000 /* Add and output        */
100.0000
obase=16 /* Set hex output base */
15  /* Convert to hex        */
F
16  /* Convert to hex        */
10
64  /* Convert to hex        */
40
255 /* Convert to hex        */
FF
256 /* Convert to hex        */
100
512 /* Convert to hex        */
200
quit/* Must type whole word */
```

Also available are scaling, function definition, and programming statements much like those in the C programming language. Other features include assignment to named registers and subroutine calling. For more information, see Chapter 5, "bc: A Calculator", XENIX *User's Guide*.

Index

CHARACTERS

A

B

C

D

G

H

I

K

L

M

N

O

P

XENIX® System V

Operating System

Operations Guide

This document was typeset with an IMAGEN® 8/300 Laser Printer.

SCO Document Number: XG-4-1-87-4.0

Contents

Chapter 1

Introduction

1.1 Overview

The XENIX operating system is a powerful system of programs, which allows you to accomplish a full spectrum of tasks, from developing high-level and assembly language programs to creating, editing, and typesetting documents. To keep this powerful system running smoothly, the XENIX system requires careful control of its operation and a regular schedule of maintenance. This guide explains how to operate and maintain the XENIX operating system on your computer, ensuring maximum performance with the least number of system problems.

This guide also explains how to expand a XENIX system with a Micnet network. A Micnet network allows serial communication between other XENIX systems in your work environment. The Micnet programs and commands include the **netutil** program, which is used to install the network, and the **mail, rcp,** and **remote** commands, which are used to pass messages, files, and commands over the network.

See Chapter 9, "Building A Micnet Network" for a complete explanation of this facility.

1.2 The System Manager

Every XENIX system should have one person in charge of system maintenance and operation. In this guide, that person is called the system manager. It is the system manager's duty to ensure the smooth operation of the system and to perform tasks that require special privileges.

Depending on the size of the system and the number of users on the system, a system manager's job can be anything from a once-a-week task to a full-time job. Even if the system is small, the system manager should faithfully perform each required maintenance task, since sloppy maintenance can affect XENIX performance.

All tasks in this guide are presented from the system manager's point of view, but many can also be accomplished by ordinary users. Since some of the tasks dramatically change the system's operation, we recommend that, whenever possible, the system manager perform these tasks. This can prevent unwanted or unnecessary changes to the system.

1.3 The Super-User Account

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

Users who are authorized to act as the super-user, including the system manager, should log in as the super-user only when it is necessary to perform a system maintenance task. Even if the system manager is the only person using the system, he should create a user account for himself and use it for day-to-day work, reserving the super-user account for system maintenance tasks only.

The number of individuals who are given the super-user password should be kept to a minimum. Misuse of the super-user powers by naive users can result in a loss of data, programs, and even the XENIX system itself.

1.4 The Keyboard

Many keys and key combinations have special meanings in the XENIX system. These keys and key combinations have special names that are unique to the XENIX system, and may or may not correspond to the keytop labels on your keyboard. To help you find the special keys, the following table shows which keys on a typical terminal correspond to XENIX system keys. A list for your particular terminal is in **keyboard**(HW).

In this table, a hyphen (-) between keys means "hold down the first key while pressing the second."

XENIX Name	Keytop	Action
BREAK	Delete	Stops current action and returns to the shell. This key is also called the INTERRUPT or DELETE key.
BACKSPACE	Backspace	Deletes the first character to the left of the cursor.
Ctrl-d	Ctrl-d	Signals the end of input from the keyboard; also exits current shell or initiates the "logout" procedure if the current shell is the login shell.
Ctrl-h	Erase	Deletes the first character to the left of the cursor. Also called the ERASE key.

Ctrl-q	Ctrl-q	Restarts printing after it has been stopped with Ctrl-s.
Ctrl-s	Ctrl-s	Stops printing at the standard output device, for example a terminal. Does not stop the program.
Ctrl-u	Ctrl-u	Deletes all characters on the current line. Also called the KILL key.
Ctrl-\	Ctrl-\	Quits current command and creates a *core* file (Recommended for debugging only). See **core**(F) for more information.
ESCAPE	Esc	Exits the current mode; for example, exits insert mode when in the editor **vi**.
RETURN	Return	Terminates a command line and initiates an action from the shell.

Many of these special function keys can be modified by the user. See **stty**(C) for more information.

1.5 About This Guide

The tasks presented in this guide range from simple ones requiring very little knowledge about XENIX , to quite complex tasks requiring extensive knowledge about XENIX 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 required to refer to other manuals, such as the XENIX *User's Guide*.

Chapter 1, "Introduction," introduces this guide.

Chapter 2, "Starting and Stopping the System," explains how to start and stop the XENIX system and how to log in as the super-user, the XENIX system's special system manager account.

Chapter 3, "Preparing XENIX for Users," explains how to create accounts for the users who work on your system, how to assign groups, and how to manage user IDs.

Chapter 4, "Using File Systems," explains how to create and mount file systems, how to set permissions, and how to keep the system secure.

Chapter 5, "Maintaining File Systems," explains how to maintain free space on the root file system and other file systems.

Chapter 6, "Backing Up File Systems," explains how to create backup copies of the root file system and other file systems.

Chapter 7, "Using Peripheral Devices," explains how to add terminals, printers and other peripheral devices.

Chapter 8, "Solving System Problems," explains how to solve system problems such as a jammed lineprinter or a forgotten password.

Chapter 9, "Building a Micnet Network," explains how to create a multiple system mailing network with Micnet.

Appendix A, "XENIX Special Device Files," presents a list of the XENIX system special files, and explains how to use these files when creating and maintaining file systems.

Appendix B, "XENIX Directories," presents a list of commonly used XENIX directories and log files.

Replace this Page
with Tab Marked:

Starting/Stopping System

Chapter 2

Starting and Stopping the System

2.1 Introduction

This chapter explains how to start and stop the XENIX system. It also explains how to log in as the super-user.

2.2 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. Starting the system requires:

- Loading the operating system
- Cleaning the file system (if the system was improperly stopped)
- Choosing the mode of system operation

The following sections describe each of these procedures.

2.2.1 Loading the Operating System

The first step in starting the system is to load the operating system from the computer's hard disk. Follow these steps:

1. Turn on power to the computer and hard disk. The computer loads the XENIX bootstrap program and displays the message:

   ```
   Boot
   :
   ```

2. Press the **RETURN** key. The bootstrap program loads the XENIX operating system.

When the system is loaded, it displays information about itself and checks to see if the "root file system" (i.e., all files and directories) is clean. If it is clean, you may choose the mode of operation. If not, the system requires you to clean the file system before choosing.

2.2.2 Cleaning the File System

You must clean the file system if the system displays the message:

```
Proceed with cleaning (y or n)?
```

This message indicates that the system was not stopped properly as described in the section, "Stopping the System", given later in this chapter. The XENIX operating system requires a clean file system to perform its tasks.

To clean the file system, enter *y* (for "yes") and press the **RETURN** key. The system cleans the file system, repairing damaged files or deleting files that cannot be repaired. It reports on its progress as each step is completed. At some point, it may ask if you wish to salvage a file. Always answer by entering *y* and pressing the **RETURN** key.

When cleaning is complete, the system usually asks you to choose the mode of operation, but it may also display the message:

** Normal System Shutdown **

** Safe to Power Off **
-or-
** Press Any Key to Reboot **

If it displays this message, you must reload the system. You can do this by pressing any key or resetting the computer and repeating the steps given in the previous section. For instructions on how to reset your computer, see the hardware manual provided with the computer.

2.2.3 Choosing the Mode of System Operation

You may choose the mode of XENIX 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* mode. Normal operation is for ordinary work on the system. This is the mode you should choose to allow multiple users to log in and begin work. System maintenance mode is a specialized mode reserved for work to be done by the system manager. It does not allow multiple users.

To choose normal operation, press the Ctrl-d key. The system displays a startup message and begins to execute commands found in the command file */etc/rc* described in Chapter 8, "Solving System Problems." When the commands are finished, the system displays the "login:" prompt. You may then log in as a normal user, as decribed in the XENIX *Introduction to XENIX*, or as the super-user, as described in the next section.

To choose system maintenance mode, enter the super-user's password (sometimes called the "root password") and press the **RETURN** key. The system displays the message of the day and the maintenance mode prompt (#). The commands in the */etc/rc* file 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.)

2.3 Logging In As the Super-User

Many system maintenance tasks, when performed during normal operation, require that you log in as the super-user. For example, you must be logged in as the super-user to stop the system.

Before you may log in as the super-user, you need 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 the Ctrl-d key until it appears.

To login as the super-user, follow these steps:

1. When you see the "login:" message, enter the super-user's login name

   root

   and press the **RETURN** key. The system prompts you for the super-user's password.

2. Enter the super-user's password and press the **RETURN** key. The system does not display the password as you enter it, so enter each letter carefully.

The system opens the super-user account and displays the message of the day and the super-user prompt (#).

Take reasonable care when you are logged in as the super-user. In particular, you should be very careful when deleting or modifying files or directories. Avoid using wildcard designators in filenames and frequently check your current working directory. Small errors can cause annoying and unwanted changes to the system and user files. Some errors can cause irretrievable damage to a file or the system.

You can leave the super-user account at any time by pressing Ctrl-d.

2.4 Stopping the System

Stopping the XENIX system takes more than just turning off the computer. You must prepare the system for stopping by using either the **shutdown** or the **haltsys** command. The following sections describe each command.

2.4.1 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.

To stop the system with the **shutdown** command, follow these steps:

1. Log in as the super-user (see the section "Logging in as Super-User" in this chapter). The system opens the super-user account and displays the message of the day and the super-user's prompt.

2. Enter:

   /etc/shutdown

   and press the RETURN key. The system loads the command which in turn prompts you for the number of minutes you wish to elapse before the computer stops:

   Minutes till shutdown? (0-15):

3. Enter any number from 0 to 15 and press the RETURN key. The system displays a warning message at each terminal, asking logged in users to finish their work and log out. As soon as all users are logged out or the specified time has elapsed, the system closes all accounts and displays the message:

   ** Normal System Shutdown **

   ** Safe to Power Off **
   -or-
   ** Press Any Key to Reboot **

   and stops.

You may now turn off the computer.

2.4.2 Using the haltsys Command

The **haltsys** command may be used to halt the system immediately. In general, it should be used only when no other users are on the system or when the system is in system maintenance mode.

To stop the system with the **haltsys** command, follow these steps:

1. Log in as the super-user (not required when in system maintenance mode). The system opens the super-user account and displays the message of the day and the super-user prompt.

2. Enter:

   /etc/haltsys

   and press the **RETURN** key. The system displays the message:

   ** Normal System Shutdown **

   ** Safe to Power Off **
   -or-
   ** Press Any Key to Reboot **

   and stops.

You may now turn off the hard disk and computer.

Replace this Page
with Tab Marked:

User Accounts

Chapter 3

Preparing XENIX for Users

3.1 Introduction

User accounts help the XENIX system manager keep track of the people using the system, and control their access to the system's 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 does his work.

It is the system manager's job to create accounts for all users on the system. It is also the manager's job to maintain user accounts by changing user passwords, login groups, and user IDs when necessary.

This chapter explains how to:

- Add user accounts to the system
- Change an account's password
- Force new passwords
- 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 in detail.

3.2 Adding a User Account

You may add a user account to the system with the **mkuser** program. The program creates a new entry in the XENIX system's */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 a *.profile* or *.login* file which contains 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

   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):

3. Enter the letter *y* (for "yes"), if you want information about the program, otherwise type the letter *n* (for "no"). 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.

   When the program continues, it prompts you to enter the new user's login name:

   Enter new user's login name:

   The login name is the name by which XENIX will know the user. It is usually a short version of the user's actual name, typed in lowercase 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.

4. Enter the new name, and press the RETURN key. The program now prompts you for information about the new user's group name and group number.

   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 not given, the XENIX system's common group "group" (with group ID 50) is used.

   The program prompts:

   Do you want to use the default group? (y/n):

   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:

> Existing groups are:
>
> Group "group" (50): demo vdemo cdemo
>
> 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.

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.

5. After entering the group name and ID, you are prompted for the initial 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 Chapter 4 of the *Introduction to XENIX*.)

6. Enter the password, and press the **RETURN** key.

7. Next, you are prompted for a shell type. You see a list and brief explanation of the available shells (**sh, vsh, csh, rsh,** and **uucp login**) and the prompt:

   > ENTER Shell type (1, 2, 3, 4 or 5) 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, and **uucp login** is an entry in */usr/lib/uucp/uucico* enabling logging in to the system via **uucp**. For more information, see **vsh** (C) in the XENIX *Reference Guide*.

8. 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** 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

John Doe, 123

shows the recommended form for a comment.

9. Enter the comment. Make sure it is 20 characters or less. If you do not wish to enter a comment, just press the **RETURN** key.

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 password (encrypted), the user ID, the group ID, the comment, the user's home directory, and the startup program. Items in the entry are separated by colons (:). (For a full description of each item, see *passwd*(M) in the XENIX *Reference Manual*.)

The program then gives you an opportunity to change the user name, password, group, or comment:

Username is "johnd", user ID is 2001.
Group name is "group", group number is 50.
Comment field is: John Doe, 123
Shell is "/bin/csh"

Do you want to change anything? (y/n):

10. 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.)

If you enter *y*, the program prompts for the item you wish to change:

username
password
group
comment
shell

Enter the name of the item you wish to change and press **RETURN**. After you have changed an item, you see the complete list of items and are asked if you wish to make other changes. When you are finished with any changes, the program adds the user.

11. The program displays the message:

    Password file updated

    followed by a description of the actions it has taken to add the new user account to the system. The program then asks if you wish to add another user to the system.

12. 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 *Introduction to XENIX*.

3.3 Changing a User's Password

Normally, an ordinary user can change the password of his own account with the **passwd** command (see the *Introduction to XENIX*). Sometimes, however, it may be necessary for the super-user to change the password for him, for example, if the user has forgotten his password and cannot get into the account to change it. The super-user may change the password of any user (including himself) with the **passwd** command.

To change a password, follow these steps:

1. Log in as the super-user.

2. Enter:

    passwd *login-name*

    (where *login-name* is the user's login name) and press the **RETURN** key. The command displays the message:

    New password:

3. Enter the new password and press the **RETURN** key. 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:

4. Enter the password again and press the **RETURN** key.

To see how an ordinary user can change his own password with the **passwd** command, see the *Introduction to XENIX*.

3.4 Forcing a New Password

From time to time, a user account may need a higher level of security than ordinary. Since the security of any account depends its password, it is important to keep the password as secret as possible. One way to provide greater security is to force users to change their passwords on a regular basis.

You can force users to change their passwords by using the **pwadmin** command. This command automatically dates each password and requires the user to provide a new password when the specified number of weeks have passed. The command also requires users to wait a minimum number of weeks before allowing them to restore their previous password. To use the **pwadmin** command, you must log in as the super-user.

You can enable password aging for a specified user by using the **-a** option. Enter:

pwadmin -a *login-name*

where *login-name* is the login name of a user. The user will then be required to wait a minimum number of weeks before he can change his password, and will be forced to change his password after a maximum number of weeks have elapsed. The **-a** option uses the default minimum and maximum values found in the */etc/default/passwd* file.

You can choose your own minimum and maximum number of weeks by using the **-min** and **-max** options. For example, a common pair of minimum and maximum values is 2 and 8. To set the minimum and maximum dates, enter:

pwadmin -min *num* -max *num* *login-name*

where *num* is a number in the range 0 to 63, and *login-name* is simply the login name of the user whose password you are administering. Note that the minimum and maximum cannot both be 0, and that the minimum must not be greater than the maximum.

If you are unsure of the current minimum and maximum values for a password, you can display them by entering:

pwadmin -d *login-name*

This command does not change the current values.

If you wish to force a user to change his password immediately, enter:

pwadmin -f *login-name*

The user is prompted on his next login to supply a new password.

When a password no longer requires extra security, you can remove the current minimum and maximum values for the password by entering:

pwadmin -n *login-name*

The system will no longer prompt for changes.

For more information about password aging, see **pwadmin**(C) and **passwd**(M) in the XENIX *Reference Manual*.

3.5 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

   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:demo
sys:x:2:
group::50:johnd,suex
```

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 XENIX text editor (see the XENIX *User's Guide*) and specify */etc/group* as the file to edit.

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 "shipping" with group ID "142" and users "johnd", "marym", and "suex", enter:

   ```
   shipping::142:johnd,marym,suex
   ```

6. Exit the editor.

To make sure you have entered the group names correctly, use the **grpcheck** 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, he can gain access to each group by using the **newgrp** command. See the *Introduction to XENIX* for details.

3.6 Changing a User's Login Group

When a user logs in, the system automatically places the user in his "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.

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 item in the entry. Items are separated by colons (:). For example, the entry:

   marym:9iKlwp:205:50:Mary March, 122:/usr/marym:/bin/sh

   has group ID "50".

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.

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** 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 whose user IDs are less than 200. The user ID is the third item in the password entry.

Note that changing a user's login group does not change the "group ownership" of his 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** (for "change group") command. For details, see the section "Changing Group Ownership" in Chapter 4.

3.7 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 his 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 item in the entry. For example, the entry:

   ```
   marym:9iKlwp:205:50:Mary March, 122:/usr/marym:/bin/sh
   ```

   has a user ID "205". Substitute the new user ID for the old one.

7. Exit the text editor.

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 at 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. Just make sure 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** (for "change owner") command described in Chapter 4, "Using File Systems."

For example, to change the ownership of johnd's home directory, enter:

    chown johnd /usr/johnd

Note that you may use the **find** command described in Chapter 6, "Backing Up File Systems," to locate all files and directories with the user's old user ID.

3.8 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** 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 may use the **tar** command to copy the files to a floppy disk (see the section "Copying Files to a tar disk" in Chapter 6).

To remove a user account, follow these steps:

1. Log in as the super-user.

2. Enter:

   cd /usr/*login-name*

   and press the **RETURN** key to change to the user's home directory. The *login-name* must be the user's login name.

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 bin

   removes the directory named *bin* and all files within this directory.

5. After removing all files and directories, make sure the user's mailbox is empty. Enter:

   cat /usr/spool/mail/*login-name*

   and press the **RETURN** key, where *login-name* is the user's login name. If the mailbox contains text, enter:

   cat /dev/null > /usr/spool/mail/*login-name*

   and press the **RETURN** key.

6. When the user's home directory and mailbox are empty, enter:

   cd /usr

   and press the **RETURN** key. The user's home directory cannot be removed until you have moved to another directory.

7. Enter:

   rmuser

   and press the **RETURN** key. The program displays a message explaining how to remove a user:

   ****rmuser-remove a user from the system****

   Press ENTER 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*.

Replace this Page
with Tab Marked:

File Systems

Chapter 4

Using File Systems

4.1 Introduction

This chapter describes one of the most important responsibilities of a system manager: controlling and recording users' access to the files and directories on the system. It introduces file systems, permissions, system security, and process accounting.

4.2 File Systems

A file system is the XENIX system's way of organizing files on mass storage devices such as hard and floppy disks. A file system consists of files, directories, and the information needed to locate and access these items.

Each XENIX system has at least one file system. This file system is called the root file system and is represented by the symbol "/". The root file system contains all the XENIX programs that may be used by the system manager. It usually contains all the user directories as well.

A XENIX system may also have other file systems that contain user directories and application programs. One reason for using other file systems is to expand the available storage space of the system. Each additional file system adds its free space to the system's total space. New file systems can be specifically created by a user, then mounted onto the system so they can be used.

You can create a file system with the **mkfs** command. This command sets the size and format of the file system and may also copy some files to the new system. You can mount a file system with the **mount** command. Once mounted, you may access the files and directories in the file system as easily as files and directories in the root file system. (The root file system is permanently mounted.) When you are finished with a file system, you can unmount it with the **umount** command.

You can create new file systems on any media with the **mkfs** command. A reason for creating new file systems on floppy disks is to establish a collection of application programs and data files that can be easily mounted and used when needed.

The following sections explain how to create and use file systems.

4.2.1 Creating a File System

You can create a file system on a formatted floppy disk by using the **mkfs** command.

To create a file system on a floppy disk, you need:

- A formatted floppy disk
- The special filename of a floppy disk drive
- The disk block size of the disk
- The gap and block numbers for the disk

To format a floppy disk, see the section "Formatting Floppy Disks" in this chapter. The special filenames for the disk drives, the disk block size, and the gap and block numbers depend on the specific system and are given in Appendix A.

Note that if a file system already exists on the disk, it will be destroyed by this procedure. For this reason, be particularly careful not to create a new file system on the root file system. If you destroy the root file system, you will have to reinstall the XENIX system.

To make a file system on a floppy disk, follow these steps:

1. Log in. You do not have to be logged in as the super-user to use the **mkfs** command.

2. Insert a formatted floppy disk into a floppy disk drive. Make sure there is no read-only tab on the disk jacket.

3. Enter:

   /etc/mkfs *specialfile blocksize gap block*

   (where *specialfile*, *blocksize*, *gap*, and *block* are supplied by you) and press the **RETURN** key. The system automatically creates the file system. If it discovers data already on the disk, the system displays the message:

   mkfs: *specialfile* contains data. Overwrite? (y/n):

   If you are sure the disk contains nothing that you want to save, enter *y* and press the **RETURN** key to overwrite the data and continue creating the file system. Otherwise, enter *n*. If you enter *n*, no file system is created.

For example, the following command creates a file system on the floppy disk drive */dev/fd1*, with blocksize 320 and gap and block numbers 2 and 8:

```
/etc/mkfs /dev/fd1 320 2 8
```

The actual filename, blocksize, gap, and block numbers vary. For the information specific to your machine see Appendix A.

4.2.2 Mounting a File System

Once you have created a file system, you can mount it with the **mount** command. To mount a file system you need:

- The special filename of a disk drive
- The name of an empty directory

The special filenames of disk drives are given in Appendix A. The directory to receive the file system may be any directory as long as it is empty (contains no files) and is not your current working directory. Note that the directory */mnt* is specifically reserved for mounted file systems.

To mount a file system, follow these steps:

1. Log in. You do not have to be logged in as the super-user to use this command.

2. Insert the disk containing the file system into a floppy disk drive.

3. Enter the appropriate **mount** command, and press the **RETURN** key. The command should have the form:

   /etc/mount *specialfile directoryname*

   where *specialfile* is the special filename of the disk drive containing the disk, and *directoryname* is the name of the directory to receive the file system. If the disk has a read-only tab, make sure you include the switch **-r** at the end of the command.

For example, you can use the following command to mount the disk in disk drive */dev/fd1* onto the directory named */account:*

```
/etc/mount /dev/fd1 /account
```

Remember to make sure that the specified directory is empty before issuing the command. If the command displays the message:

```
mount: Structure needs cleaning
```

use the **fsck** command to clean the file system and try to mount it again (see the section "File System Integrity" in Chapter 5). If the command displays the message:

mount: Device busy

either the file system has already been mounted and cannot be mounted twice, or a user is currently in the directory. You must wait for users to leave a directory before you can mount the directory.

To check that the file system was properly mounted, use the **cd** command to change to the directory containing the mounted system and the **l** command to list the contents. The command displays the files and directories in the file system. Be sure to use the **cd** command to leave the directory after finishing your work in it.

Note that frequently used file systems can be mounted automatically when starting the system by appending the appropriate **mount** commands to the */etc/rc* file. See the section, "Changing the */etc/rc* File," in Chapter 8 for details.

4.2.3 Unmounting a File System

You can unmount a mounted file system with the **umount** command. Unmounting a file system does not destroy its contents. It merely removes access to the files and directories in the file system.

To unmount a mounted file system, enter:

/etc/umount *specialfile*

and press the **RETURN** key. The *specialfile* is the name of the special file corresponding to the disk drive containing the disk with the file system. The command empties the directory that previously contained the file system, and makes the directory and the corresponding disk drive available for mounting another file system.

For example, the following command unmounts a file system from the disk drive */dev/fd1*:

/etc/umount /dev/fd1

Before unmounting a file system, make sure that no files or directories are being accessed by other commands or programs. The **umount** command displays the following message if you or another user are currently in the directory containing the file system:

umount: Device busy

4.2.4 Formatting Floppy Disks

You can format floppy disks with the **format** program. Formatted disks are required whenever you create a file system. They are also required when you back up a file system with the **sysadmin** program (see Chapter 6, "File System Backups").

To format a floppy disk, follow these steps:

1. Insert a disk into floppy disk drive 0. Make sure there is no read-only tab on the disk jacket.

2. Enter:

   ```
   format /dev/device-name
   ```

   and press the **RETURN** key. The program formats the disk.

Warning: The format command is machine-specific. Refer to the **format** (C) manual page originally included in the XENIX *Installation* Guide for more information.

If you do not insert the disk properly into the drive, the program displays an error message and stops.

In general, the system manager should format spare floppy disks in advance. Note that formatting removes all data from the disk, so if you are formatting a disk that already contains data, make sure that the data is nothing you wish to save.

4.3 Permissions

Permissions control access to all the files and directories in a XENIX system. In XENIX , ordinary users may access those files and directories for which they have permission. All other files and directories are inaccessible.

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 permissions apply to all other users.

4.3.1 Displaying Permissions

You can display the permission settings for all the files in a directory with the **l** (for "list directory") command. This command lists the permissions along with the name of the file's owner, the size (in bytes), and the date and time the file was last changed. The command display has the following format:

```
-rw-rw---- 1 johnd group 11515 Nov 17 14:21 file1
```

The permissions are shown as a sequence of ten characters at the beginning of the display. The sequence is divided into four fields. The first field (the "type" field) has a single character, the other fields ("user", "group", and "other") have three characters each. The characters in the fields have the following meanings.

In the type field:

d Indicates the item is a directory

\- Indicates the item is an ordinary file

b Indicates the item is a device special block I/O file

c Indicates the item is a device special character I/O file

In the "user", "group", and "other" fields:

r Indicates read permission. Read permission for a file means you may copy or display the file. Read permission for a directory means you may display the files in that directory.

w Indicates write permission. Write permission for a file means you may change or modify the file. Write permission for a directory means you may create files or subdirectories within that directory.

x Indicates execute permission (for ordinary files) or search permission (for directories). Execute permission for a file means you may invoke the file as you would a program. Execute permission for a directory means you may enter that directory with the **cd** command.

\- Indicates no permission.

For example, the permissions:

```
-rwxrwxrwx
```

indicate an ordinary file with full read, write, and execute access for everyone (user, group, and other).

The permissions:

```
-rw-------
```

indicate an ordinary file with read and write access for the user only.

The permissions:

```
drwxr-x--x
```

indicate a directory with search access for everyone, read access for the user and group, and write access for only the user.

When you create a file, the XENIX system automatically assigns the following permissions:

```
-rw-r--r--
```

This means that everyone may read the file, but only the user may write to it. When you create a directory, the system assigns the permissions:

```
drwxr-xr-x
```

This means everyone may search and read the directory, but only the user may create and remove files and directories within it.

4.3.2 Changing Permissions

You can change the permissions of a file or a directory with the **chmod** (for "change mode") command. This command requires that you tell it how to change the permissions of a specific file or directory. You do so by indicating which levels of permissions you wish to change (user "**u**", group "**g**", or other "**o**"), how you wish to change them (add "**+**" or remove "**-**"), and which permissions you wish to change (read "**r**", write "**w**", or execute "**x**"). For example, the pattern:

```
u+x
```

adds execute permission for the user. The pattern:

```
go-w
```

removes write permission for group and other.

The **chmod** command has the form:

chmod *pattern file* ...

where *file* is the name of a file or directory. If more than one name is given, they must be separated by spaces. For example, to change the permissions of the file "receivables" from "-rw-r--r--" to "-rw-------", enter:

```
chmod go-r receivables
```

Press the **RETURN** key.

After using **chmod**, use the **l** command to check the results. If you have made a mistake, use **chmod** again to correct the mistake.

4.3.3 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 means every file receives the permissions:

```
-rw-r--r--
```

Every directory receives the permissions:

```
drwxr-xr-x
```

You can change the mask and the initial permissions your files and directories receive by using the **umask** command.

The **umask** command has the form:

umask *value*

where *value* is a three-digit number. The three digits represent user, group, and other permissions, respectively. The value of a digit defines which permission is given as shown by the following table:

Digit	Permission
0	Read and write (also execute for directories)
1	Read and write
2	Read (also execute for directories)
3	Read
4	Write (also execute for directories)
5	Write
6	Execute for directories
7	No permissions

For example, the command:

    umask 177

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.

4.4 Managing File Ownership

Whenever a file is created by a user, the system automatically assigns "user ownership" of that file to that user. This allows the creator to access the file according to the "user" permissions. The system also assigns a "group ownership" to the file. The group ownership defines which group may access the file according to the "group" permissions. The group is the same group to which the user who created the file belongs.

Only one user and one group may have ownership of a file at any one time. (These are the owner and group displayed by the **l** command.) However, you may change the ownership of a file by using the **chown** and **chgrp** commands.

4.4.1 Changing User Ownership

You can change the user ownership of a file with the **chown** command. The command has the form:

    chown *login-name file* ...

where *login-name* is the name of the new user, and *file* is the name of the file or directory to be changed. For example, the command:

    chown johnd projects.june

changes the current owner of the file *projects.june* to "johnd".

The **chown** command is especially useful after changing the user ID of a user account (see the section, "Changing a User's ID" in Chapter 3).

You must be logged in as the super-user to use this command.

4.4.2 Changing Group Ownership

You can change the group ownership of a file with the **chgrp** command.

The command has the form:

chgrp *group-name file* ...

where *group-name* is the name of a group given in the */etc/group* file and *files* are the name of the file you wish to change. For example, the command:

chgrp shipping projects.june

changes the group ownership of the file *projects.june* to the group named "shipping".

The **chgrp** command is especially useful if you have changed the login group of a user (see the section, "Changing a User's Login Group" in Chapter 3).

4.5 System Security

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 manager to protect the system.

4.5.1 Physical Security

You can protect the physical components of the computer, especially system disks, by taking these steps:

1. Keep unessential personnel out of the work area.
2. Organize and lock up all floppy disks when not in use. They should not be stored with the computer itself.
3. Keep disks away from magnetism, direct sunlight, and severe changes in temperature.
4. Do not use ball point pens to write labels on disks.
5. Make backup copies of all floppy disks (see the section, "Copying Floppy Disks," in this chapter).

Replace this Page
with Tab Marked:

File Systems

Chapter 4

Using File Systems

4.1 Introduction

This chapter describes one of the most important responsibilities of a system manager: controlling and recording users' access to the files and directories on the system. It introduces file systems, permissions, system security, and process accounting.

4.2 File Systems

A file system is the XENIX system's way of organizing files on mass storage devices such as hard and floppy disks. A file system consists of files, directories, and the information needed to locate and access these items.

Each XENIX system has at least one file system. This file system is called the root file system and is represented by the symbol "/". The root file system contains all the XENIX programs that may be used by the system manager. It usually contains all the user directories as well.

A XENIX system may also have other file systems that contain user directories and application programs. One reason for using other file systems is to expand the available storage space of the system. Each additional file system adds its free space to the system's total space. New file systems can be specifically created by a user, then mounted onto the system so they can be used.

You can create a file system with the **mkfs** command. This command sets the size and format of the file system and may also copy some files to the new system. You can mount a file system with the **mount** command. Once mounted, you may access the files and directories in the file system as easily as files and directories in the root file system. (The root file system is permanently mounted.) When you are finished with a file system, you can unmount it with the **umount** command.

You can create new file systems on any media with the **mkfs** command. A reason for creating new file systems on floppy disks is to establish a collection of application programs and data files that can be easily mounted and used when needed.

The following sections explain how to create and use file systems.

4.2.1 Creating a File System

You can create a file system on a formatted floppy disk by using the **mkfs** command.

To create a file system on a floppy disk, you need:

- A formatted floppy disk
- The special filename of a floppy disk drive
- The disk block size of the disk
- The gap and block numbers for the disk

To format a floppy disk, see the section "Formatting Floppy Disks" in this chapter. The special filenames for the disk drives, the disk block size, and the gap and block numbers depend on the specific system and are given in Appendix A.

Note that if a file system already exists on the disk, it will be destroyed by this procedure. For this reason, be particularly careful not to create a new file system on the root file system. If you destroy the root file system, you will have to reinstall the XENIX system.

To make a file system on a floppy disk, follow these steps:

1. Log in. You do not have to be logged in as the super-user to use the **mkfs** command.

2. Insert a formatted floppy disk into a floppy disk drive. Make sure there is no read-only tab on the disk jacket.

3. Enter:

   /etc/mkfs *specialfile blocksize gap block*

   (where *specialfile*, *blocksize*, *gap*, and *block* are supplied by you) and press the **RETURN** key. The system automatically creates the file system. If it discovers data already on the disk, the system displays the message:

   mkfs: *specialfile* contains data. Overwrite? (y/n):

   If you are sure the disk contains nothing that you want to save, enter *y* and press the **RETURN** key to overwrite the data and continue creating the file system. Otherwise, enter *n*. If you enter *n*, no file system is created.

For example, the following command creates a file system on the floppy disk drive */dev/fd1*, with blocksize 320 and gap and block numbers 2 and 8:

```
/etc/mkfs /dev/fd1 320 2 8
```

The actual filename, blocksize, gap, and block numbers vary. For the information specific to your machine see Appendix A.

4.2.2 Mounting a File System

Once you have created a file system, you can mount it with the **mount** command. To mount a file system you need:

- The special filename of a disk drive
- The name of an empty directory

The special filenames of disk drives are given in Appendix A. The directory to receive the file system may be any directory as long as it is empty (contains no files) and is not your current working directory. Note that the directory */mnt* is specifically reserved for mounted file systems.

To mount a file system, follow these steps:

1. Log in. You do not have to be logged in as the super-user to use this command.

2. Insert the disk containing the file system into a floppy disk drive.

3. Enter the appropriate **mount** command, and press the **RETURN** key. The command should have the form:

   /etc/mount *specialfile directoryname*

   where *specialfile* is the special filename of the disk drive containing the disk, and *directoryname* is the name of the directory to receive the file system. If the disk has a read-only tab, make sure you include the switch **-r** at the end of the command.

For example, you can use the following command to mount the disk in disk drive */dev/fd1* onto the directory named */account:*

```
/etc/mount /dev/fd1 /account
```

Remember to make sure that the specified directory is empty before issuing the command. If the command displays the message:

```
mount: Structure needs cleaning
```

use the **fsck** command to clean the file system and try to mount it again (see the section "File System Integrity" in Chapter 5). If the command displays the message:

mount: Device busy

either the file system has already been mounted and cannot be mounted twice, or a user is currently in the directory. You must wait for users to leave a directory before you can mount the directory.

To check that the file system was properly mounted, use the **cd** command to change to the directory containing the mounted system and the **l** command to list the contents. The command displays the files and directories in the file system. Be sure to use the **cd** command to leave the directory after finishing your work in it.

Note that frequently used file systems can be mounted automatically when starting the system by appending the appropriate **mount** commands to the */etc/rc* file. See the section, "Changing the */etc/rc* File," in Chapter 8 for details.

4.2.3 Unmounting a File System

You can unmount a mounted file system with the **umount** command. Unmounting a file system does not destroy its contents. It merely removes access to the files and directories in the file system.

To unmount a mounted file system, enter:

/etc/umount *specialfile*

and press the **RETURN** key. The *specialfile* is the name of the special file corresponding to the disk drive containing the disk with the file system. The command empties the directory that previously contained the file system, and makes the directory and the corresponding disk drive available for mounting another file system.

For example, the following command unmounts a file system from the disk drive */dev/fd1*:

/etc/umount /dev/fd1

Before unmounting a file system, make sure that no files or directories are being accessed by other commands or programs. The **umount** command displays the following message if you or another user are currently in the directory containing the file system:

umount: Device busy

4.2.4 Formatting Floppy Disks

You can format floppy disks with the **format** program. Formatted disks are required whenever you create a file system. They are also required when you back up a file system with the **sysadmin** program (see Chapter 6, "File System Backups").

To format a floppy disk, follow these steps:

1. Insert a disk into floppy disk drive 0. Make sure there is no read-only tab on the disk jacket.

2. Enter:

   ```
   format /dev/device-name
   ```

   and press the **RETURN** key. The program formats the disk.

Warning: The format command is machine-specific. Refer to the **format** (C) manual page originally included in the XENIX *Installation*Guide for more information.

If you do not insert the disk properly into the drive, the program displays an error message and stops.

In general, the system manager should format spare floppy disks in advance. Note that formatting removes all data from the disk, so if you are formatting a disk that already contains data, make sure that the data is nothing you wish to save.

4.3 Permissions

Permissions control access to all the files and directories in a XENIX system. In XENIX , ordinary users may access those files and directories for which they have permission. All other files and directories are inaccessible.

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 permissions apply to all other users.

4.3.1 Displaying Permissions

You can display the permission settings for all the files in a directory with the l (for "list directory") command. This command lists the permissions along with the name of the file's owner, the size (in bytes), and the date and time the file was last changed. The command display has the following format:

```
-rw-rw---- 1 johnd group 11515 Nov 17 14:21 file1
```

The permissions are shown as a sequence of ten characters at the beginning of the display. The sequence is divided into four fields. The first field (the "type" field) has a single character, the other fields ("user", "group", and "other") have three characters each. The characters in the fields have the following meanings.

In the type field:

d Indicates the item is a directory

- Indicates the item is an ordinary file

b Indicates the item is a device special block I/O file

c Indicates the item is a device special character I/O file

In the "user", "group", and "other" fields:

r Indicates read permission. Read permission for a file means you may copy or display the file. Read permission for a directory means you may display the files in that directory.

w Indicates write permission. Write permission for a file means you may change or modify the file. Write permission for a directory means you may create files or subdirectories within that directory.

x Indicates execute permission (for ordinary files) or search permission (for directories). Execute permission for a file means you may invoke the file as you would a program. Execute permission for a directory means you may enter that directory with the **cd** command.

- Indicates no permission.

For example, the permissions:

```
-rwxrwxrwx
```

indicate an ordinary file with full read, write, and execute access for everyone (user, group, and other).

The permissions:

```
-rw-------
```

indicate an ordinary file with read and write access for the user only.

The permissions:

```
drwxr-x--x
```

indicate a directory with search access for everyone, read access for the user and group, and write access for only the user.

When you create a file, the XENIX system automatically assigns the following permissions:

```
-rw-r--r--
```

This means that everyone may read the file, but only the user may write to it. When you create a directory, the system assigns the permissions:

```
drwxr-xr-x
```

This means everyone may search and read the directory, but only the user may create and remove files and directories within it.

4.3.2 Changing Permissions

You can change the permissions of a file or a directory with the **chmod** (for "change mode") command. This command requires that you tell it how to change the permissions of a specific file or directory. You do so by indicating which levels of permissions you wish to change (user "**u**", group "**g**", or other "**o**"), how you wish to change them (add "**+**" or remove "**-**"), and which permissions you wish to change (read "**r**", write "**w**", or execute "**x**"). For example, the pattern:

```
u+x
```

adds execute permission for the user. The pattern:

```
go-w
```

removes write permission for group and other.

The **chmod** command has the form:

chmod *pattern file ...*

where *file* is the name of a file or directory. If more than one name is given, they must be separated by spaces. For example, to change the permissions of the file "receivables" from "-rw-r--r--" to "-rw-------", enter:

```
chmod go-r receivables
```

Press the **RETURN** key.

After using **chmod**, use the **l** command to check the results. If you have made a mistake, use **chmod** again to correct the mistake.

4.3.3 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 means every file receives the permissions:

```
-rw-r--r--
```

Every directory receives the permissions:

```
drwxr-xr-x
```

You can change the mask and the initial permissions your files and directories receive by using the **umask** command.

The **umask** command has the form:

umask *value*

where *value* is a three-digit number. The three digits represent user, group, and other permissions, respectively. The value of a digit defines which permission is given as shown by the following table:

Digit	Permission
0	Read and write (also execute for directories)
1	Read and write
2	Read (also execute for directories)
3	Read
4	Write (also execute for directories)
5	Write
6	Execute for directories
7	No permissions

For example, the command:

```
umask 177
```

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.

4.4 Managing File Ownership

Whenever a file is created by a user, the system automatically assigns "user ownership" of that file to that user. This allows the creator to access the file according to the "user" permissions. The system also assigns a "group ownership" to the file. The group ownership defines which group may access the file according to the "group" permissions. The group is the same group to which the user who created the file belongs.

Only one user and one group may have ownership of a file at any one time. (These are the owner and group displayed by the **l** command.) However, you may change the ownership of a file by using the **chown** and **chgrp** commands.

4.4.1 Changing User Ownership

You can change the user ownership of a file with the **chown** command. The command has the form:

chown *login-name file* ...

where *login-name* is the name of the new user, and *file* is the name of the file or directory to be changed. For example, the command:

```
chown johnd projects.june
```

changes the current owner of the file *projects.june* to "johnd".

The **chown** command is especially useful after changing the user ID of a user account (see the section, "Changing a User's ID" in Chapter 3).

You must be logged in as the super-user to use this command.

4.4.2 Changing Group Ownership

You can change the group ownership of a file with the **chgrp** command.

The command has the form:

chgrp *group-name file* ...

where *group-name* is the name of a group given in the */etc/group* file and *files* are the name of the file you wish to change. For example, the command:

chgrp shipping projects.june

changes the group ownership of the file *projects.june* to the group named "shipping".

The **chgrp** command is especially useful if you have changed the login group of a user (see the section, "Changing a User's Login Group" in Chapter 3).

4.5 System Security

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 manager to protect the system.

4.5.1 Physical Security

You can protect the physical components of the computer, especially system disks, by taking these steps:

1. Keep unessential personnel out of the work area.
2. Organize and lock up all floppy disks when not in use. They should not be stored with the computer itself.
3. Keep disks away from magnetism, direct sunlight, and severe changes in temperature.
4. Do not use ball point pens to write labels on disks.
5. Make backup copies of all floppy disks (see the section, "Copying Floppy Disks," in this chapter).

4.5.2 Access Security

You can protect the system from access by unauthorized individuals by taking these steps:

1. Remind users to log out of their accounts before leaving the terminal.

2. Discourage users from choosing passwords that are easy to guess. Passwords should be at least six characters long and include letters, digits, and punctuation marks.

3. Keep the super-user password secret from all but necessary personnel.

4.5.3 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 other devices, but can also be used by ordinary users to gain direct access to these devices.

Since direct access bypasses the system's normal protection mechanisms and allows ordinary users to examine and change all files in the system, it is wise to protect the special files to ensure system security.

To protect the XENIX special files, log in as the super-user and use the **chmod** command to set appropriate permissions. For example, to disallow any access by ordinary users, set the permissions of such special files as */dev/mem*, */dev/kmem*, */dev/root*, and */dev/usr* to read and write access for the user only. Note that you must not change the permissions for the */dev/tty* files.

4.5.4 Copying Floppy Disks

To ensure against the loss of data stored on floppy disks, you 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 under XENIX, you can use them over again without reformatting.

If you have disks that have been formatted under another operating system, you must reformat them under XENIX 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**(C) command to format floppies. This command is described in the section "Formatting Floppy Disks" in this chapter. The **diskcp** can also format floppies for you. This is discussed in the next steps.

To copy a floppy disk using **diskcp**:

1. Insert the disk you want to copy, also known as the *source* floppy, in drive 0, your primary floppy drive.

2. Insert another floppy in the other drive. This floppy is also known as the *target* disk. Note that any information already on the target disk will be destroyed.

   If you have only one disk drive, leave the source floppy in the drive. **diskcp** will prompt 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
   ```

   and press **RETURN**.

   If your computer has dual floppy drives. Enter the following command to copy the image directly on the target floppy:

   ```
   diskcp -d
   ```

   and press **RETURN**.

   If you do not need to format the target floppy, simply enter:

   ```
   diskcp
   ```

   and press **RETURN**.

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.

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 in "Formatting Floppy Disks" in this chapter.

3. Enter:

   dd if=/dev/fd0 of=/dev/fd1 count=*blkcount*

   and press the **RETURN** key. The *blkcount* is the number of blocks on the disk to be copied (see Appendix A for details).

This command copies the first disk to the second, then displays a record of the number of blocks copied.

4.6 Using XENIX Accounting Features

The XENIX system provides a set of commands that allow the system manager 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 (i.e., 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.

Process accounting is helpful on systems where users are being charged for their access time, but it may also be used to develop a detailed record of system, command, and system resource usage.

There are several commands which may be used to do process accounting. Of these, the most useful are **accton** and **acctcom**. The **accton** command starts and stops process accounting. When invoked, the command copies pertinent information about each process to the file named */usr/adm/pacct*. The **acctcom** command is used to display this information. The command has several options for displaying different types of accounting information.

4.6.1 Starting Process Accounting

Process accounting can be started at any time, but is typically started when the system itself is started. You can start process accounting with the **accton** command. Enter:

    accton /usr/adm/pacct

The command automatically creates a new file */usr/adm/pacct* and begins to copy process accounting information to it. If the */usr/adm/pacct* file exists before starting **accton**, the file contents are deleted.

You can start process accounting automatically whenever you reboot the system. To do this, edit the system startup file */etc/rc*. Among other commands, there are several that start up process accounting, as well as backup the accounting log file */usr/adm/pacct*.

These commands are commented out, but if you remove the comment characters at the beginning of those lines, the commands are executed everytime you reboot the system. Make these lines in */etc/rc*:

```
# mv /usr/adm/pacct /usr/adm/opacct
# > /usr/adm/pacct ; chmod 644 /usr/adm/pacct
# [ -x /etc/accton ] && /etc/accton /usr/adm/pacct
```

look like this:

```
mv /usr/adm/pacct /usr/adm/opacct
> /usr/adm/pacct ; chmod 644 /usr/adm/pacct
[ -x /etc/accton ] && /etc/accton /usr/adm/pacct
```

Note that when you start the system after editing */etc/rc*, the contents of the */usr/adm/pacct* file is saved in the file */usr/adm/opacct* overwriting the contents of */usr/adm/opacct*.

4.6.2 Displaying Accounting Information

The **acctcom** command reads processing accounting information from the */usr/adm/pacct* file by default, then displays selected information on your terminal screen. The command usually displays basic accounting information, such as the process's program name, the name of the user who invoked the process, the start and stop times for the process, and the number of execution seconds in real time and CPU time. The command has several options that can be used to display selected information.

To display the average size of each process, enter:

```
acctcom
```

The command displays the basic information plus the average size of each process.

To display basic accounting information about a specific command, enter:

acctcom -n *command*

where *command* is the name of the command you are interested in. The command responds by displaying each entry for the specified *command*. For example, to display each entry for the system command, enter:

acctcom -n units

displays each entry for the system command **units**.

To display information about the number and size of input and output counts, enter:

acctcom -i

The command displays basic program information plus the number of characters and blocks transferred or read by each program.

To display information about a program's use of system resources, enter:

acctcom -h

The command displays the basic information plus the "use factor." The use factor is a number generated and used by the system to determine how each process should be scheduled for execution. Processes with high use factors use a high percentage of the system resources and are therefore scheduled after processes with lower factors.

Chapter 5

Maintaining File Systems

5.1 Introduction

File system maintenance, an important task of the system manager, keeps the XENIX system running smoothly, keeps the file systems clean, and ensures adequate space for all users. To maintain the file systems, the system manager must monitor the free space in each file system, and take corrective action whenever it gets too low.

This chapter explains the file system maintenance commands. These commands report how much space is used, locate seldom-used files, and remove or repair damaged files.

5.2 Maintaining Free Space

The XENIX system operates best when at least 15% of the space in each file system is free. In any system, the amount of free space depends on the size of the disk containing the file system and the number of files on the disk. Since all disks have a fixed amount of space, it is important to carefully control the number of files stored on the disk.

If a file system 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 file system. This means that the user's normal work on the computer (creating new files and expanding existing ones) stops.

The only remedy for a file system which has less than 15% free space is to delete one or more files from the file system. The following sections describe strategies for keeping the free space available.

5.2.1 Strategies for Maintaining Free Space

The system manager should regularly check the amount of free space of all mounted file systems and remind users to keep their directories free of unused files. You can remind users by including a reminder in the message of the day file */etc/motd*. (See the section, "Changing the */etc/motd* File" in Chapter 8).

If the amount of free space slips below 15%, the system manager should:

1. Send a system-wide message asking users to remove unused files.

2. Locate exceptionally large directories and files, and send mail to the owner asking him to remove unnecessary files.

3. Locate and remove temporary files and files named *core*.

4. Clear the contents of system log files.

Finally, if the system is chronically short of free space, it may be necessary to create and mount an additional file system.

5.2.2 Displaying Free Space

You can find out how much free space exists in a particular file system with the **df** (for "disk free") command. This command displays the number of "blocks" available on the specific file system. A block is 512 characters (or bytes) of data.

The **df** command has the form:

df *specialfile*

where *specialfile* can be the name of a XENIX special file corresponding to the disk drive containing the file system (see Appendix A, "XENIX Special Device Files"). If you do not give a special filename, then the free space of all normally mounted file systems is given.

For example, to display the free space of the root file system */dev/root*, enter:

df /dev/root

and press the **RETURN** key. The command displays the special filename and the number of free blocks. You may compute the percentage of free space by comparing the displayed value with the total number of blocks in the file system. See Appendix A, "XENIX Special Device Files," for a list of the total blocks.

5.2.3 Sending a System-Wide Message

If free space is low, you may 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
```

and press the **RETURN** key. Enter the message, pressing the **RETURN** key 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 link to other terminals.

5.2.4 Displaying Disk Usage

You can display the number of blocks used within a directory by using the **du** command. This command is useful for finding 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 file system. 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 */usr/johnd*, enter:

```
du /usr/johnd
```

and press the **RETURN** key. The command displays the name of each file and directory in the */usr/johnd* directory and the number of blocks used.

5.2.5 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 disk drive containing the file system (see Appendix A, "XENIX Special Devices Files").

For example, to display the owners of files in the file system mounted on the disk drive */dev/fd1*, enter:

```
quot /dev/fd1
```

and press the **RETURN** key. The command displays the users who have files in the file system and the number of blocks in these files.

5.2.6 Mailing a Message to a User

If a particular user has excessively large directories or files, you may send a personal message to the user with the **mail** command.

To begin sending a message through the mail, enter:

mail *login-name*

and press the **RETURN** key. The *login-name* must be the login name of the recipient. To send a message, enter the message, press the **RETURN** key, and then press Ctrl-d. If the message has more than one line, press the **RETURN** key at the end of each line. The **mail** command copies the message to the user's mailbox, where he may view it also by using the **mail** command. See the XENIX *User's Guide* for details.

5.2.7 Locating Files

You may locate all files with a specified name, size, date, owner, and/or last access date by using the **find** command. The command is useful for locating seldom-used and excessively large files.

The **find** command has the form:

find *directory parameters*

The *directory* must be the name of the first directory to be searched. (It will also search all directories within that directory.) The parameters are special names and values that tell the command what to search for (see **find** (C) in the XENIX *Reference Manual* for complete details). The most useful *parameters* are:

-name *file*

-atime *number*

-print

The **-name** parameter causes the command to look for the specified *file*. The **-atime** parameter causes the command to search for files which have not been accessed for the *number* of days. The **-print** parameter causes the command to display the locations of any files it finds.

For example, to locate all files named *core* in the directory */usr*, enter:

```
find /usr -name core -print
```

and press the **RETURN** key. The command displays the locations of all files it finds.

5.2.8 Locating *core* and Temporary Files

You can locate *core*, and temporary files with the **find** command.

A *core* file contains a copy of a terminated program. The 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. These files 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 which have not been accessed for a reasonable period of time. For example, to find all *core* files in the */usr* directory which have not been accessed for a week, enter:

```
find /usr -name core -atime +7 -print
```

and press the **RETURN** key.

5.2.9 Clearing Log Files

The XENIX system maintains a number of files, called log files, that contain information about system usage. When new information is generated, the system automatically appends this information to the end of the corresponding file, preserving the file's previous contents. This means the size of each file grows as new information is appended. Since the log files can rapidly become quite large, it is important to periodically clear the files by deleting their contents.

You can clear a log file by entering:

cat < /dev/null > *filename*

where *filename* is the full pathname of the log file you wish to clear. A log file normally receives information to be used by one and only one program, so its name usually refers to that program. Similarly, the format of a file depends on the program that uses it. See Appendix B, "XENIX Files and

Directories," for descriptions of the log files.

In some cases, clearing a file affects the subsequent output of the corresponding program. For example, clearing the file */etc/ddate* forces the next backup to be a periodic backup (see Chapter 6, "Backing Up File Systems").

5.2.10 Expanding the File System

If free space is chronically low, it may be to your advantage to expand the system's storage capacity by creating and mounting a new file system. Once mounted, you may use this new file system for your work, or even copy user or system directories to it.

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 file system allows some of the users and directories to be transferred from the hard disk, freeing a significant amount of space on the existing file system and improving system operation. For details about creating and mounting file systems, see Chapter 4, "Using File Systems."

5.3 File System Integrity

Since file systems are normally stored on hard and floppy disks, occasional loss of data from the file system through accidental damage to the disks is not unusual. Such damage can be caused by conditions such as an improper system shutdown, hardware errors in the disk drives, or a worn out disk.

Such damage usually affects one or two files, making them inaccessible. In very rare cases, the damage causes the entire file system to become inaccessible.

The XENIX system provides a way to restore and repair a file system if it has been damaged. The **fsck** (for "file system check") command checks the consistency of file systems and, if necessary, repairs them. The command does its best to restore the information required to access the files, but it cannot restore the contents of a file once they are lost. The only way to restore lost data is to use backup files. For details about backup disks, see Chapter 6, "Backing Up File Systems."

5.3.1 Repairing the File System

You can repair a file system with the **fsck** command. The command has the form:

fsck *specialfile*

The *specialfile* must be the name of the special file corresponding to the disk drive containing the file system (see Appendix A, "XENIX Special Device Files").

For example, to check the file system on the disk in the disk drive */dev/fd1*, enter:

fsck /dev/fd1

and press the **RETURN** key. The program checks the file system 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, the command 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. If you suspect a file system problem and wish to try to save some of the damaged file or files, check other possible remedies before you invoke the command.

5.3.2 Automatic File System Check

The XENIX system sometimes requests a check of the file system when you first start it. This usually occurs after an improper shutdown (for example, after a power loss). The file system check repairs any files disrupted during the shutdown. For details, see the section "Cleaning the File System" in Chapter 2.

Chapter 6

Backing Up File Systems

6.1 Introduction

A file system backup is a copy, on backup media such as floppy disks or tape, of the files in the root directory and other regularly mounted file systems. A backup allows the system manager, when logged in as *root* or *sysinfo* to save a copy of the file system as it was at a specific time. The copy may be used later to restore files that are accidentally lost or temporarily removed from the file system to save space.

This chapter explains how to create backups of the root directory and other file systems, and how to restore files from the backups.

6.2 The Sysinfo Account

Always use the *sysinfo* account whenever you make or restore backups. This is because:

- You should always perform backups using the same user and group ID, and;
- *sysinfo* has restricted root permissions for backups.

If you try to create backups as an ordinary user, you do not have access permissions for all files. If you make backups as *root* you may accidentally destroy files because *root* can do anything at all.

The *sysinfo* account solves this dilemma by having restricted *root* permissions. The *sysinfo* user can read all files but may overwrite only those owned by *sysinfo*.

The *sysinfo* account is already set in */etc/passwd*. Only *root* can become *sysinfo*, and only then using the **su**(C) command. No one may log in as *sysinfo*.

6.3 Strategies for Backups

The system manager should back up the root directory, and any other mounted file systems, on a regular basis. In particular, the manager should make daily copies of all files modified during the day, and should make periodic (e.g., weekly) copies of the entire root directory and other mounted file systems.

The XENIX system offers two ways to back up file systems, the **sysadmin**(C) program and the **tar**(C) command.

The **sysadmin** program is a formal maintenance program for systems that require a rigorous schedule of file system backups. The program automatically locates modified files, copies them to backup media, and optionally produces a list of the files. If your system has many users and a large number of files that are modified daily, use **sysadmin** to make regular backups.

The **tar** command is useful on systems with one or two users, or on any system where ordinary users wish to make personal copies of their directories and files. The command lets the system manager or user choose the files and directories to be copied. The command does not, however, automatically locate modified files.

A typical backup schedule includes a daily backup and a periodic backup once a week. A daily backup copies only those files modified during that day; a periodic backup copies all files in the file system. The appropriate schedule for a system depends on how heavily the system is used, and how often files are modified. In all cases, a periodic backup should be done at least once a month.

The system manager should schedule backups at times when few, if any, users are on the system. This ensures that the most recent version of each file is copied correctly.

A regular schedule of backups requires a large number of floppy disks and adequate storage for the disks. Daily backups should be saved at least two weeks; periodic backups should be saved indefinitely. Disks 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 disk may be used to create new backups.

Note

> If the number of floppies needed for making backups grows too large, the system manager can use the **backup**(C) command instead of **sysadmin**. **backup** is called by **sysadmin** and by using **backup** directly, the system administrator can tailor the number of floppies used to fit the needs of the individual site. Refer to **backup**(C) in the XENIX *Reference*.

6.4 Using the sysadmin Program

The **sysadmin** program performs daily and periodic backups, lists backup files, and restores individual files from backup disks and tapes. The program presents each task as an item in a menu. To perform a task, simply choose the appropriate item from the menu and supply the required information.

6.4.1 Creating Backups

You may create backups on any archival media available on your system. 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 a floppy disk, see the section "Formatting Floppy Disks" in Chapter 4. You also have the option to do formatting from the **sysadmin** program, in which case the program issues the formatting command.

To create a backup, follow these steps:

1. Log in as the super-user.

2. Enter:

    sysadmin

    and press **RETURN**. The program displays a file system maintenance menu.

```
Filesystem Maintenance Options

        1. Do a daily backup (level 9)
        2. Do a periodic backup (level 0)
        3. Get a backup listing
        4. Restore backed-up file(s)
        5. Restore an entire filesystem

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

3. Enter "1" for a daily backup or "2" for a periodic backup and press **RETURN**. Note that if the system has never had a periodic backup, it automatically performs one, even if you have chosen a daily backup.

4. You see two menus in sequence. The first is a selection of filesystems to back up, and the second is a selection of backup, or archival media.

   Each menu also presents the option "Other:". The "Other" option appears in many of the **sysadmin** menus. Using this option, you can backup or restore filesystems that are not described in */etc/default/filesys* and use backup devices that are not described in */etc/default/archive*. **sysadmin** uses these two files, which are described in the section of this chapter "Editing */etc/default/filesys* and */etc/default/archive*", to create the filesys and archive device menus. With the "Other" option, you can use filesystems and devices you have not yet described in the default files.

   If you select this option, you are prompted for further information. See "Editing */etc/default/filesys* and */etc/default/archive*" for more information.

5. Load a volume, tape or disk, 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:

   Change volumes

6. 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

When doing a periodic backup, 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.

6.4.2 Getting a Backup Listing

You can keep a record of the files you have backed up by invoking the **sysadmin** program and selecting the third item in the 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 the super-user.

2. Enter:

   sysadmin

   and press **RETURN**. The program displays the system maintenance menu.

3. Enter "3" and press **RETURN**. You are prompted for the name of the file in which to place the listing. Enter **RETURN** 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 **RETURN**. 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:

lpr *filename*

where *filename* is the name of the file that contains the backup list, and press **RETURN**. To save space after printing the file, you should remove it with the **rm**(C) command.

6.4.3 Restoring Backup Files

You can restore files from the backup volumes by invoking the **sysadmin** program and selecting the fourth item in the menu. You will need the complete set of backup volumes containing the latest version of the file or files you wish to restore, and you need the "full pathname" of the file or files you wish to restore. This pathname is given in the backup listing.

To restore a file, follow these steps:

1. Log in as the super-user.

2. Enter:

   sysadmin

   and press **RETURN**. The program displays the file system maintenance menu.

3. Enter "4" and press **RETURN**. You see:

```
1. Restore file(s) to their original location(s)
2. Restore file(s) to another location

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

4. When you select option 1, be aware that files with the same names as files currently on your disk will overwrite the disk files. If you are not absolutely sure that your backup contains the preferred version of the files, you should restore them in a temporary location, such as */tmp*, and compare them with your current files on disk using **diff**(C) or **cmp**(C).

5. If you select option 2, you are prompted for a directory name. Your restored files are placed in that directory, and are named by the numbers (inode numbers) displayed by the restore program. Be sure to write these numbers down, in order to identify your files.

6. The program prompts you to enter the full pathname of a file you wish to restore. Enter the pathname and press **RETURN**. The program prompts for another pathname. Enter another pathname, or press **RETURN** to continue the program. If you press **RETURN**, the program prompts you to insert the first volume in the backup set.

7. Next you are prompted for the archive device. Select the same drive and medium on which your backup was created, for example, tape or floppy disk.

8. Load volume 1 of the backup set into the drive, then press **RETURN**. The program displays the inode numbers of the files you have given, then prompts for a volume number.

9. 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.

10. The program prompts for volume numbers until all of the files have been found. Continue to read volumes in reverse order until the first volume made has been loaded and you have returned to the main menu. See the **restore**(C) manual page in the XENIX *User's Reference* for more information.

11. If you select another directory as the destination for the restored files, the files are placed in that directory and named by their inode numbers. You can move them to any location using the **mv**(C) command:

    mv *inode filename*

    inode is what **sysadmin** names the file when it restores it. *filename* is the new name you want for the file. For example, to restore a file */usr/johnd/projects.june* from *224*, enter:

    mv 224 /usr/johnd/projects.june

6.4.4 Restoring an Entire Filesystem

To restore an entire filesystem, select option "5" in the main **sysadmin** menu. Do not use this option carelessly; all information currently in the target filesystem will be overwritten. Restore an entire filesystem only onto a clear or newly-made filesystem, or onto a filesystem that is completely corrupted or inaccessible. See **divvy**(C), **mkfs**(C) and other documentation on making filesystems.

You can use this option to restore any filesystem, including the root filesystem. However, there are special considerations; if your system will boot and run but there are missing or corrupted files, you can use option 4 from **sysadmin** to recover the necessary files.

If you need to restore the entire root filesystem, you must at least partially reinstall XENIX . If your system is very similar to the standard XENIX distribution, you should reinstall the original XENIX distribution and use **sysadmin** to replace single files that you have modified. Note that it is faster and easier to replace files from the distribution disks than from backup copies. However, if your system has been significantly changed from the distribution, you may decide to only partially reinstall XENIX and recover the filesystem from backups.

Note

If you decide to partially reinstall XENIX , be very careful during the hard disk initialization portion of the installation. It is vital to preserve the hard disk partitions and parameters if there are other divisions resident on the disk.

To completely reinstall XENIX, consult the XENIX *Installation Guide* you received with your distribution. To partially reinstall, and recover your filesystem from backups, perform the following steps:

- Use your original distribution disks and install only the XENIX run time system and the **sysadmin** package.
- When the packages have been installed, quit **custom** and reboot your system.
- While in single-user mode, invoke **sysadmin** and select option 5 to restore an entire filesystem. You are warned that this is a serious task and are asked to decide whether you wish to continue.
- Next you see the filesystem menu. Select the filesystem that you wish to restore, or the "Other" option to enter the name of another filesystem, perhaps the one that you have just created. If you select the root filesystem, you see a shortened version of the previous warning, and you are asked if you wish to continue. If you select "Other", you are prompted for the device name of the filesystem that you wish to back up or restore.
- Next you see the archive menu. Select the medium on which your filesystem is backed up, for example, tape or floppy.
- You are prompted to load the first volume of the backup into the selected drive. The restore program prompts you to change volumes if necessary.

6.4.5 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.

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/rfd048ds9  desc="Floppy Drive 0 (48dsdd)"\
blocking=18  size=360  format="format  -f  /dev/rfd048ds9"
```

The value part of "name=value" pairs "desc="Floppy Drive 0 (48dsdd)"" and "format="format -f /dev/rfd048ds9"" contain spaces, therefore they must be surrounded by quotes in order to be interpreted correctly.

Note that the newline is preceded by a backslash (\), which causes the two lines to be interpreted as one line. This convention makes the files easier to read.

The minimum necessary information about an archive device is:

- a character device name (cdev=) or a block device name (bdev=), and
- a size, (size=). Without these, **sysadmin** will not use the entry.
- In addition to these, a description, (desc=), can be added which appears in the archive device menu. If a format command is included, then you may format media from the **sysadmin** program.
- A blocking factor for the medium, (blocking=), is included in the above example for use by other programs. It is not used by **sysadmin.**

The size (size=), in the above example, is considered to be a number of K, or kilobytes. If an entry is included for density, (density=), as in the following example, then the size is taken to mean a number of feet, which is more appropriate for tape than for floppy disks:

```
cdev=/dev/rmt0  desc="Cartridge Drive (600 ft tape)"  \
blocking=20  size=600  density=20000
```

The minimum necessary information about a filesystem is:

- a character device name, (cdev=), or a block device name, (bdev=).
- Additionally, you may include a description, (desc=), which appears in the **sysadmin** filesystem menu,

- and the mount directory, (mountdir=), which is also used by **/etc/mnt**, 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
```

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 that line, and it will be treated as a comment and ignored. Later, if you need the entry again, then you can delete the #. For example, the entry below is ignored. You only need a # in front of the first line, since the back slash still has the effect of causing the two lines to be treated as one line.

```
#cdev=/dev/rfd048ds9 desc="Floppy Drive 0 (48dsdd)"     \
blocking=18 size=360 format="format -f /dev/rfd048ds9"
```

6.5 Using the tar Command

The **tar** command copies files and directories to and from your backup media (floppy disks or tape). On systems with one or only a few users, it gives the system manager a direct way to make backup copies of the files modified during a day. On systems with many users, it gives ordinary users a way to make personal copies of their own files and directories.

6.5.1 Copying Files With tar

The **tar** command has the form:

tar cvf *specialfile files*

The *specialfile* is the name of the special file corresponding to a disk or tape drive (see Appendix A, "XENIX Special Device Files"). Also, the section in this chapter "tar Defaults" discusses default *specialfiles*. The *files* are the names of the files or directories you wish to copy.

To use the **tar** command, you need:

- Media; tape or formatted floppies. For details on formating a disk, see the section "Formatting Floppy Disks" in Chapter 4, and **format**(C) in the XENIX *Reference*.

- The names of the files and/or directories you wish to copy.

For example, to copy the files *a*, *b*, and *c* to a floppy disk in the drive */dev/fd1*, enter this command and press RETURN:

```
tar cvf /dev/fd1 a b c
```

To copy a directory named *recipes*:

```
tar cvf /dev/fd1 recipes
```

To copy the contents of the current directory, and all subdirectories, enter this command and press RETURN:

```
tar cvf /dev/fd1 .
```

Note that files are restored according to the pathnames given when they are first copied. Therefore, never use full, absolute pathnames when making **tar** backups, because you might accidentally erase an existing version of backed up files. For example, **never do this**:

```
tar cvf /usr/allan/examplefile
```

6.5.2 Restoring Files With tar

To restore files from a backup made with the **tar** command, use this command:

tar xvf *specialfile*

The *specialfile* must be the name of the special file corresponding to the disk drive containing the **tar** disk.

Files are copied from the backup media to your current directory, and all pathnames are preserved. If there are directories on the backup, they are recreated relative to the current directory.

For example, to restore files from the disk in the drive */dev/fd1*, enter:

```
tar xvf /dev/fd1
```

and press **RETURN**.

Since the **tar** command copies files only to the current directory, make sure you are in the desired directory before you invoke the command.

Replace this Page
with Tab Marked:

Using Peripheral Devices

Chapter 7

Using Peripheral Devices

7.1 Introduction

One important task of the system manager is to add peripheral devices such as terminals, hard disks, tape drives and lineprinters to the system. Adding these devices lets more users access the system, provides extra storage space for users' files and directories, and adds to overall system capabilities.

To add a peripheral device, the system manager must:

- connect the physical device to the computer, then
- use the correct system commands to enable the device for operation.

This chapter explains how to do this and how to maintain the devices once they are added.

Note that physical connections between a device and the system vary depending upon hardware. This chapter provids some information about making the necessary physical connections, but for more information about these connections, see the hardware manuals provided with the device and your computer.

7.2 Using Multiscreen™

With multiscreens, the video display and keyboard act as several separate terminals. You switch between screens by holding down the **alt** key and pressing **f1-f10**. The tty01 is the "alt-f1" terminal, tty02 is the "alt-f2" terminal, tty03 is "alt-f3", etc. The multiple screen feature uses the */dev/tty[01...12]* device files. These device files provide character I/O between the system and the computer screen and keyboard. The actual number of multiscreens available depends upon the amount of memory in your personal computer and is displayed when the system boots. There are generally between two and six multiscreens enabled, up to a maximum of twelve.

Each multiscreen (device file) is independent of the rest, so you can log in and run programs on each screen. Output from programs is saved in a screen buffer, and when you switch back to a screen, you see the latest screenful of output. If you stop output to a screen, for example by entering Ctl-S, this affects only that screen. Output continues on all the other screens.

Although all of the multiscreens may be open and active at the same time, you can see only one screen at a time. The selected multiscreen is the terminal currently "connected" to the keyboard.

To select any active screen, press **alt-F*n***, where **F*n*** is one of the function keys on the far left side of the keyboard. For example, this keystroke switches you to screen 6, corresponding to */dev/tty06*:

```
alt-F6
```

You can also rotate through the screens by pressing **Ctrl-PrtSc** (using the **Ctrl** key and the **PrtSc** key. Use this feature to access screens for which you do not have function keys. For example, if you have twelve multiscreens enabled, but the computer keyboard only has ten function keys, you can display screen eleven by first going to screen 10, then pressing Ctrl-PrtScr. To access screen 12, press Ctrl-PrtScr again. If you press Ctrl-PrtScr a third time, you rotate back to the first multiscreen, *tty01*.

Refer to the XENIX *Reference* for more on **multiscreen**(M) and **screen**(HW).

The system displays the number of multiscreens enabled on boot. This number is determined by the amount of memory in your computer. Do not try to use more than the auto-configured number of screens.

7.3 Using a Cartridge Tape Drive

A tape cartridge drive is a mass storage device that uses 1/4 inch tape cartridges to store data. It is also referred to as a QIC (quarter inch cartridge) tape drive. 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 under XENIX. are listed in the Operating System *Release Notes*. For hardware-specific information, refer to the manual for your drive and **tape**(HW) in the XENIX *Reference*.

7.3.1 Installation and Configuration

Refer to the your tape drive hardware manual for physical installation instructions. Once it is connected to your computer, enter this command to configure the drive:

```
mkdev tape
```

mkdev runs an interactive program called **tapeinit** which explains the configuration process and prompts you for necessary input.

See **mkdev**(C) for more information.

7.3.2 Accessing the Drive

You use a QIC tape drive much like a floppy. You can use the standard commands such as **tar**(C), **dd**(C), **cpio**(C), **backup**(C), **dump**(C), and **restore**(C) commands are used to access a cartridge tape drive. This section summarizes tape operations and how to perform simple backups and restores. For instructions on how to perform system backups and related filesystem maintenance, see the chapter on "Backing Up File Systems" in this guide.

7.3.3 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*

The *devicefile* is the file name that corresponds to the cartridge drive. *files* are the names of the files or directories to be copied. For example, to copy all the files in the directory */u/bogart* to the cartridge drive **/dev/rct0**, enter:

```
tar cvf /dev/rct0 /u/bogart
```

To restore files stored on tape, insert the cartridge containing the files or directories you wish to restore and enter the following command:

tar xvf *devicefile*

tar restores all the files on the tape to the present directory.

For detailed information on backup operations and other methods of copying files, refer to **tar**(C), **cpio**(C), **backup**(C), and **restore**(C) in the XENIX *Reference*.

7.3.4 Tape Drive Maintenance

The **tape**(C) utility is performs various tape maintenance operations. It sends commands and receives status from the tape drive. The basic form of the command is:

tape *command* [devicefile]

For example, to rewind the cartridge tape device */dev/rct0*, enter:

```
tape rewind /dev/rct0
```

Other commands are:

erase Erase tape cartridge. Also retensions.

reset Reset tape controller and tape drive. Clears error conditions and returns tape subsystem to power up state.

reten Retension tape cartridge. Should be used periodically to remedy slack tape problems, generally resulting in an unusually large number of tape errors.

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 retension cartridges to keep everything operating error-free.

7.3.5 /etc/default Entries

Depending on the utilities used, you will need to create entries in certain files in the */etc/default* directory. These entries are the default information that utilities such as **tar** use when accessing the given device. Using **tar** and a QIC tape drive (configured as */dev/rct0*) as an example, the device would require the following entry in */etc/default/tar*:

```
archive0=/dev/rct0 20 300 y
```

In this case, */dev/rct0* is the default device (archive0), with a blocking factor of 20, and a volume size of 300 feet. *y* indicates that (yes) the device is a tape drive. If you are configuring archive0 to access an rctmini tapedrive, the entry would be:

```
archive0=/dev/rctmini 18 0 y
```

The utilities **backup** and **restore** have similar files and entries. For more information on default files, see **default**(M) in the XENIX *Reference* and the manual entry for the particular command.

7.3.6 Tape Formatting

Tape cartridges used with the mini tape drive (ctmini) must be formatted before use. Use the **format**(C) utility to format a cartridge tape. For example, this formats a ctmini tape cartridge:

```
format /dev/rctmini
```

The **-e** option erases servo information:

```
format -e /dev/rctmini
```

See also **tape**(HW) and **format**(C) for more information.

7.4 Configuring Serial Ports

If you wish to add a multi-port expansion card, you also need to create new device files for the additional ports. Use **mkdev** command with the **serial** option. This calls an interactive program, **serinit**, to initialize the serial ports on the installed card.

To configure the additional ports, follow these steps:

1. Install the expansion card. See the instructions furnished with the serial card and with your computer.

   Boot the system and enter system maintenance mode.

2. When you are in system maintenance mode, enter:

   ```
   /etc/mkdev serial
   ```

3. This invokes **serinit**, which begins with the following display:

```
You would like to install a:

        1. 1 port card
        2. 4 port card
        3. 8 port card

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

Enter the appropriate number and press RETURN. The program will respond with the following menu (if your system has two slots, only COM1 and COM2 appear):

```
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", a table is displayed containing ports, card types, I/O and status addresses.

Enter the appropriate number and press RETURN. After accepting the selected port, **serinit** will display messages indicating the new ports that have been configured 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. Do not attempt to access both the modem control and non-modem control ports of the same serial line simultaneously.

You have now configured your serial ports. Make sure the serial ports are also defined in the system configuration.

Check your hardware manual to determine how your system is configured, via a CMOS database or by switch settings on the main system board. If your system is configured using a CMOS database, the ports are defined in the database (see **cmos**(HW)).

Otherwise, define the ports by setting the proper switches on the main system board. Refer to your hardware manual for the settings. Note: You will get an error message if an attempt is made to to access a serial port that has not been installed and defined.

7.5 Setting Up a Serial Console

You can configure a serial device as the system console rather than the display adapter. The **boot** progrem sets the default console at boot time in the following manner:

1. **boot** reads */etc/default/boot* and looks for the entry **SYSTTY=***x* where *x* specifies the system console device.

2. If **SYSTTY** is not found or */etc/default/boot* is unreadable, **boot** checks for a display adapter and designates it as the 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.

If you wish to use a serial console, simply create an entry in */etc/default/boot* as follows:

SYSTTY=*x*

where *x* is "0" or "1", for the display adapter or COM1 serial port, respectively. You can also use the command **systty=***x* at the boot prompt to change the system console device. Note that this command does not create or change an **SYSTTY** entry in */etc/default/boot*.

7.6 Adding a Terminal

Adding extra terminals gives more users simultaneous access to the system. You can add a terminal by connecting it to an RS-232 serial line on the system and enabling it with the **enable**(C) command.

Many different terminals work well with the XENIX operating system. A short list of recommended models is given in **terminals**(M) in the XENIX *Reference*. The file */etc/termcap* contains a more comprehensive list of terminals that work with XENIX . For a description of */etc/termcap*, see **termcap**(M) in the XENIX *Reference*.

Before you add a terminal, you must know how to connect the terminal to a serial line on the computer. You also need to know the name of the serial line. Physical connections for the terminal are usually explained in the terminal hardware manual. The names of the standard available serial lines are given in Appendix A, "XENIX Special Device Files." If you add a serial card, the possible names of the additional device files are listed in the manual section **serial**(HW).

The following steps show how to install a terminal with the standard "COM"serial lines, or with serial expansion cards:

1. This step is for serial expansion cards. If you are adding a terminal directly to a COM port, skip to the next step.

   If you are using a supported 4 or 8-port 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 XENIX *Release Notes* for the correct addresses.

This only applies to boards that are listed as supported in the *Release Notes*. Vendor supplied drivers may not print a recognition message at boot time.

Configure your board for the correct interrupts for the two standard COM ports. COM1 should be configured as interrupt 4 and COM2 as interrupt 3. Most serial cards use one interrupt per board, so two four-port boards could use COM1 and COM2. Be aware of the requirements of other products and hardware to avoid interrupt conflicts. See **serial**(HW) in the XENIX *Reference* for more information on COM1 and COM2.

When a supported card is correctly configured for the desired COM port and recognized at bootup, run this command:

/etc/mkdev serial

This creates device files for your extra serial ports.

2. Make sure you are in multi-user mode. You are if you typed Ctrl-D, at the startup prompt "Type CONTROL-D to proceed with Normal Startup (or give root password for system maintenance):".

   Make sure your terminal is plugged in, turned on, and set for 9600 baud, 8 data bits, 1 stop bit, and no parity. If your terminal does not work in this mode, see the section "Changing the gettydefs File" later in this chapter, and **stty**(C) in the XENIX *Reference* for alternate ways to configure your terminal.

   Some terminals are designed to connect directly to a computer with a straight cable. Others are designed to connect to a modem instead. In the latter case, you need a cable that has pins 2 and 3 crossed. This is called a "null modem" or "modem connecter". The terminal should be connected with a cable so that Transmit Data on the serial port is connected to Read Data on the terminal, and Transmit Data on the terminal is connected to Read Data on the serial port. Signal Ground should be connected to Signal Ground. Other pins probably do not need to be connected.

   For more information on your specific terminal, see 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 /dev/*ttyname*

In the above command, *ttyname* is the device special name of the port in question. Make sure you are using a non-modem control. For example, */dev/tty1a*, not */dev/tty1A*. For more information on serial port names, see **serial**(HW), in the XENIX *Reference* and the section "Configuring Serial Ports" in this chapter.

4. Check the */etc/ttys* file so that the entry for this serial port looks like this:

   0m*ttyname*

   In this example, *ttyname* is the name of the device file, for example */dev/tty1a*. If the entry does not look like this, edit the file with a text editor to correct it.

   Also, if */etc/ttys* contains entries for *tty11*, *tty12*, *tty13* and replace them with *tty1a*, *tty2a*, *tty1A* and respectively. Make sure you remove duplicate entries for the same serial port.

5. From the console, as *root*, see if you can redirect output to the terminal by typing:

   date > /dev/*ttyname*

   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, there may be a problem with the hardware.

   Verify the following:

   - The terminal is plugged in.
   - The cable is configured correctly. If the serial port you are using has a 25-pin connector (DB-25), insure that pins 2, 3 and 7 are being used as specified in step (2) above. Again, note that other pins probably do not have to be used.

     However, if your system or expansion card has a 9-pin connector (DB-9), then you must have a 9-pin to 25-pin connector. Consult your hardware manual for information on 9-pin to 25-pin connections.
   - Re-check your terminal setup configuration. See step (2).
   - Re-check the switches on your serial port. If you are using a multi-port card, try other lines on that card.
   - Try attaching the terminal to a standard serial port, COM1 or COM2, to see if the terminal and cabling are working

correctly. If you are already using a COM port, try switching to another one.

If you have already installed another terminal successfully, try switching hardware one piece at a time, with the working terminal. This may help you isolate a hardware problem.

6. Once you are able to get the date printed on your terminal, enable the port and look for the "login:" prompt:

   enable /dev/*ttyname*

   If you do not see the "login:" prompt, verify that **getty** is actually running on the port and that the software is configured properly with this command:

   ps -t *ttyname*

   The output should look similar to the following, where "login" or "getty" is listed in the "command" column:

PID	TTY	TIME	COMMAND
2557	1a	0:06	getty

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, shutdown the system, reboot, go into multiuser mode and try again.

7.7 Setting Terminal Lines

Your XENIX system can automatically adapt to several different terminal baud rates and settings. The same program that displays the login message, **getty**(C), 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 terminals lines.

On your XENIX system, **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.

7.7.1 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:

label — Identifies the *gettydefs* entry to **getty**. This could be a number or a letter. *label* corresponds to the line mode field in */etc/ttys*. **init** passes the line mode to **getty** as an argument.

initial-flags — Sets terminal line characteristics when **getty** first establishes the connection. **getty** recognizes the flags listed in *tty*(M), in the XENIX *Reference*. Often the only flag in this field is the one setting the baud rate. For example, B300 would set the speed to 300.

final-flags — Sets the terminal line characteristics just before **getty** executes **login**. 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.

login-prompt — Contains the login prompt message that greets users. This field is printed exactly as it is entered including spaces and tabs. The '@' in the login-prompt field is expanded to the first line in the file */etc/systemid* (unless the '@' is preceded by a '\').

Several character sequences are recognized, including:

\n	Line feed	\t	Tab
\r	Carriage return	\f	Form feed
\v	Vertical tab	\b	Backspace
nnn	(3 octal digits) The specified ASCII characters		

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 XENIX. The default program is */etc/login.*

If preceded by the keyword "AUTO," **getty** does not prompt for a username, but instead uses the devicename (i.e. *dev/tty03*) as the username and immediately executes the *login-program*

Each field is separated by pound sign (#), and each entry in *gettydefs* is separated by a blank line.

An entry in *gettydefs* might look like this:

```
4# B1200 # B1200 SANE TAB3 HUPCL #Login: #2#AUTO
```

The number 4 identifies this entry to **getty**. In the next field, the baud rate is set to 1200. The final settings in the third field include the baud rate (B1200), SANE (a composite flag for a number of characteristics), and HUPCL (hangs up line on final close). The login prompt appears as "Login:", and if this setting is not successful, **getty** proceeds to label 2 in *gettydefs.* AUTO attempts to log in the user "*/dev/console*
(or other tty)" executing */etc/login.*

If the last entry also contains a *filename,* that *login program* is executed. (Note that the *filename* and the corresponding *login program* are user created.) For example, including a file such as */etc/dial_login* for a line connected to a modem can be used. It would set the user ID, acquire a password, validate the user, and then become the user. It could possibly require a password for the system in addition to an account password and even have a special set of login environment variables included in */etc/default/dial_login.*

7.7.2 Changing the gettydefs File

The file */etc/gettydefs* already exists on your XENIX system and has sets of entries for the dial-up lines, and terminal lines. These different sets correspond to line mode settings in */etc/ttys.* The **init** program passes the line mode 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 on your XENIX system might look like this:

```
1# B2400 # B2400 SANE IXANY TAB3 #@!login: #2

2# B4800 #B4800 SANE IXANY TAB3 #@!login: #3

3# B9600 # B9600 SANE IXANY TAB3 #@!login: #1
```

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 second and third fields from B2400 to B1200.

3. Exit the text editor, saving *gettydefs*.

The sample file should look like this:

```
1# B1200 # B1200 SANE IXANY TAB3 #@!login: #2

2# B4800 #B4800 SANE IXANY TAB3 #@!login: #3

3# B9600 # B9600 SANE IXANY TAB3 #@!login: #1
```

You can also add additional terminal line settings to *gettydefs*. Flags and permissible values for terminal settings are listed in **tty** (M), 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 will be 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:

   ```
   4# B300 # B300 SANE IXANY TAB3 #@!login: #1
   ```

4. To incorporate label 4 into the set of labels, change the *next label* field for entry 3 to 4:

   ```
   3# B4800 # B4800 SANE IXANY TAB3 #@!login: #4
   ```

   getty is now directed from label 3 to 4, and then back to 1.

5. Exit the text editor, saving the revised *gettydefs* file.

The new *gettydefs* looks like this:

```
1# B1200 # B1200 SANE IXANY TAB3 #@!login: #2

2# B4800 #B4800 SANE IXANY TAB3 #@!login: #3

3# B9600 # B9600 SANE IXANY TAB3 #@!login: #4

4# B300 # B300 SANE IXANY TAB3 #@!login: #1
```

7.7.3 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 that 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
```

If any of the values and settings in *gettydefs* are not permitted, **getty -c** displays them on your terminal screen.

For more information on **getty** and *gettydefs*, see **getty**(M) and **gettydefs**(F), in the XENIX *Reference*.

7.8 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 and change the operating characteristics of a serial line with the **stty** (for "set tty") command.

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 < *specialfile*

In the above command, *specialfile* is the name of the device special file corresponding to the serial line (see Appendix A, "XENIX Special Device Files"). 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. The meaning of this information is explained in **stty**(C) in the XENIX *Reference*.

One common change to a serial line is changing the baud rate. This is usually done from a terminal connected to another serial line since changing the rate disrupts communication between the terminal and 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* < *specialfile*

where *baud-rate* is the current baud rate of the terminal, and *specialfile* 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, the command:

```
stty 9600 < /dev/tty1a
```

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 changing 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, the command:

```
stty tabs
```

causes the system to expand tabs with spaces (used with terminals which do not expand tabs on their own), and the command:

```
stty echoe
```

causes the system to remove a deleted character from the terminal screen when you back over it with the **BACKSPACE** 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 which requires parity generation and detection, or special input and output processing.

For a full description of this command, see **stty**(C) in the XENIX *Reference*.

7.9 Setting the Terminal Type

The XENIX system requires 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.

If you are using the Bourne shell (*sh*), 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

and press the **RETURN** key. From C–shell, enter at the shell prompt ("%"):

setenv TERM ansi

and press **RETURN**.

If you are not sure which name you may use for *termtype*, you can view the names by displaying the */etc/termcap* file. To display the file, enter:

cat /etc/termcap

and press the **RETURN** key. Since the file is very large, you will need to use the Ctrl-s key to stop the display at every full screen. You may view more of the file by pressing the Ctrl-q key or the Spacebar.

You can let the system define the terminal type automatically whenever you log in by including the TERM assignment in your *.profile* file (see Chapter 8 "Solving System Problems" under "Changing the *.profile* File").

For an alternate method of setting your terminal type see **tset**(C) in the XENIX *Reference*

If you do let the system set the terminal type be careful when logging in on terminals that are not the same as your normal terminal. The XENIX 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.

7.10 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.

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 *specialfile*

   where *specialfile* is the name of the serial line to which the terminal is attached. For example, the command:

   disable /dev/tty1a

   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.

When using the **disable** command, make sure that you wait a full minute between each use of the command. Failure to do so can cause a system crash.

7.11 Modem Usage under XENIX

7.11.1 Serial Lines

XENIX supports modem control on serial ports. The following device names refer to the serial ports with and without modem control.

Device:	Function:
/dev/tty1a	main serial adapter **without** modem control.
/dev/tty1A	main serial adapter **with** modem control.
/dev/tty2a	alternate serial adapter **without** modem control.
/dev/tty2A	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"

For systems including multi-port serial cards, the devices */dev/tty*[*1,2*][*a-m*] refer to use *without* modem control, and the devices */dev/tty*[*1,2*][*A-M*] refer to use *with* modem control.

7.11.2 Dialing Out From Your Computer

The **cu**(C) and **uucp**(C) utilities are used to call remote systems and transfer data under XENIX . The file */usr/lib/uucp/L-devices* (referred to as *L-devices*) contains information used by these programs to determine the characteristics of a particular serial line. The *L-devices* file comes set for the most common uses of **cu** and **uucp**.

The *L-devices* file contains lines which specify the device for the line, the call-unit associated with the line (0 for direct lines), and the baud rate, that are to be used by **uucp**. The following table shows example *L-devices* lines for various types of connections (direct or modem).

For Outgoing Calls					
Serial Line	**Connection**	***L-devices* line**			
main serial adapter	(direct)	DIR	tty1a	0	1200*
main serial adapter	(modem)	ACU	tty1A	tty1A	1200*
alternate serial adapter	(direct)	DIR	tty2a	0	9600*
alternate serial adapter	(modem)	ACU	tty2A	tty2A	300*
* example baud rate. Substitute the baud rate for your machine or modem. Direct hookups depend on the baud rate setting of the login on the remote machine. Modem hookups depend on the capability of your modem and the modem you are dialing.					

Modem control devices should be used with lines connected to modems.

Note that these *L-devices* lines are compatible alternatives; they can exist in your *L-devices* file at the same time.

It is best to disable a line login before a dial-out program uses it. (See the section on "Shared Dial-In/Dial-Out".) To disable a line, enter:

```
disable tty
```

This command disables */dev/tty1a* for dialing out:

```
disable /dev/tty1a
```

To invoke **cu** for a direct line, enter:

```
cu dir
```

cu will select the first device from the *L-devices* file that matches the connection type, (DIR or ACU) and the speed. The default speed is 1200 baud. To select a specific line use the **-l** option. For example:

```
cu -l tty2a dir
```

This makes specific use of tty2a. To request a speed other than the default (1200), request use the **-s** option:

```
cu -s 9600 dir
```

This will find the first direct, 9600 baud line, in this case tty2a.

For dialing, both **cu** and **uucp** use the file */usr/lib/uucp/dial*, an executable C program. This program dials a Hayes® Smartmodem 1200 or Smartmodem 1200B (a plugin modem card).

If you are using a Hayes® Smartmodem 2400, the dialer program you will need to use is in **/usr/lib/uucp/dialHA24**. To replace the default dialer with the 2400 version, use the following commands:

```
cd /usr/lib/uucp

ln dialHA24 dial
```

The source for the default **dial** and a makefile for recompiling the source program. are included in the directory */usr/lib/uucp*. If you have any other kind of modem, you can modify either of the two source files and create your own **dial** program. Note that you must have the XENIX *Development System* installed to compile a new *dial* program.

To replace the **dial** program with another 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

   and change the name from *dialHA12* to the name of the dialing program you wish to use. When this is done, exit the file, saving the changes you have made.

3. Next, enter the command:

   make

   to your shell prompt and press **RETURN**.

4. When the **make** command is finished, link the file *dial* to the new dialing program you have compiled. Use the command:

   ln *dialing.program* dial

5. You now have a new */usr/lib/uucp/dial* program.

The dialer program can also be a **sh**(C) script.

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. Unplug a modem wire at the computer. Three-wire cables are not sufficient when using modems. Several other pins must be connected for the modem to operate properly. If you are unsure as to what to use, a ribbon cable that connects all pins will work correctly.

7.11.3 Dialing Into Your Computer Under XENIX

To allow dialing into your computer, you must enable a serial line that recognizes modem control signals, with the **enable**(C) command. When using the **enable** command, make sure that you wait a full minute between each use of the command. Failure to do so may send too many signals to the **init**(C) program, which will then terminate. If **init** terminates, no new logins are possible.

To use the main serial adapter, enter:

```
disable tty1a
enable tty1A
```

Or, for the the alternate serial adapter, 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 use the same line in both its modem and non-modem invocations at the same time as this will cause an error.

7.11.4 Shared Dial-In/Dial-Out

XENIX supports the use of dial-in and dial-out on the same modem line, without having to disable the login.

When a dial-out program is using the line, the login will be disabled. If someone is logged in on a line when a dial-out program attempts to use it, the dial-out program will fail to open the device.

For this feature to work correctly, the modem control device must be used, and carrier detection must be enabled in the modem. (For information on using dial-in/dial-out in conjunction with **uucp**, refer to Chapter 6 in the XENIX *User's Guide* "Building a Communications System.")

7.11.5 Hayes Modem Settings

Proper modem configuration is necessary when using **cu** and **uucp**. Modem settings differ for each modem. Consult your modem manual for the proper switch settings.

Smartmodem 1200

If you have a Hayes Smartmodem 1200, 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 will be sent to, (echoed by), the modem to the terminal or computer. When switch 8 is down, the modem is able to interpret the command being issued. This allows both the XENIX and DOS communications systems to work.

Smartmodem 2400

The Hayes 2400 Smartmodem requires on-line configuration if it is 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. You must configure the modem by issuing set up commands via **cu**(C). The form of the **cu** command is:

cu -s2400 -l tty*nn* dir

nn is the "tty" number of the serial line. To configure a modem on tty1a, enter this command and press <RETURN>:

cu -s2400 -l tty1a dir

Next, enter the following commands to configure the modem. They will be saved in the modem's non-volatile memory. If you do not want to save the settings, do not enter the last command (at&w). Commands are in the left column and short descriptions of what they do are in the right column. Follow each command with a RETURN:

at&f	Fetch factory configuration.
att	Tone dialing.
atl0	Low speaker volume.
at&d2	Set dtr "2": go on hook when dtr drops.
at&c1	Set dcd "1": dcd tracks remote carrier.
ats0=1	Answer phone after 1 ring (AA light should come on).

ate0	No echo (modem will no longer echo what is sent to it).
atq1	Quiet mode (modem will not respond with "OK" after this command or any that follow).
at&w	Saves settings in non-volatile memory.

Exit from **cu** by entering a "tilde" and a "period", followed by a RETURN:

~.

The modem is now configured and ready for use.

/usr/lib/uucp/L.sys is the file containing information about systems to which **uucp** can connect. In order to prefix a number in *L.sys* with a non-numeric character, that character must be defined in the file */usr/lib/uucp/L-dialcodes*. In some cases this is needed to pass special codes on to modems. For example, a modem might require the string "P" to pulse dial (to dial on a line with pulse dial, instead of touch tone dial). A sample *L-dialcodes* line is:

PULSE P

The number would be listed in *L.sys* as "PULSE4085551234". For more information on *L-dialcodes* and *L.sys*, refer to Chapter 6 in the XENIX *User's Guide* "Building a Communications System"

7.11.6 Modem Control

uucp must be used with modem control serial lines.

The modem must be set to respond to DTR (Data Terminal Ready). Check your modem manual for instructions on setting DTR. Note that the dialer port should be owned by *uucp*.

7.12 Adding a Lineprinter

This section explains how to add new lineprinters to your XENIX system and how to use the lineprinter commands to organize and control your printing.

The XENIX lineprinter spooling system is a collection of commands that help you, as system manager, to efficiently install, monitor, and control the lineprinters serving your system. When a user requests a file to be printed using the **lp**(C) command, the lineprinter system responds with a "request ID." This consists of the name of the printer on which the file will be 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 easily control printer output. For more information on **lp**, see "Using the Lineprinter," in the *Introduction to XENIX*.

There are several terms used to describe the line printer system:

device — The target for **lp** output. It can be a hard-wired printer, a terminal that is sometimes used as a printer, or a regular file. A device can be represented by a full XENIX pathname.

printer — The name assigned by the system manager to represent a device. This name can have up to 14 characters. At different times, a printer may be associated with different devices.

class — An ordered list of printers. Print requests sent to a class of printers are printed by the first available member of that class.

destination — A *destination* is where print requests are sent. A destination can be a class or a printer.

Consult your computer and lineprinter hardware manuals for information on making the connection between your system and printing devices.

7.12.1 Installing a Printer: lpinit

To install new printing devices on your XENIX system, use the **lpinit**(C) command. Before you use **lpinit** you should first know the port to which the lineprinter is connected or the XENIX pathname of the device (for example, */dev/tty1a*) and the lineprinter interface program. A model interface program is supplied with your XENIX system. For more information on printer interface programs, see "Printer Interface Programs."

When you give the **lpinit** command, you are prompted with a series of questions. For most of them you need to supply your own answers, entering the information as you are prompted. When you are prompted for the printer name, however, you are given a default name option. If you wish to choose the default answer, simply press the **RETURN** key. If you make a mistake while responding to the questions, just press the **DELETE** key or the **INTERRUPT** key and start again.

Example

The following example shows how to add a line printer to your system. The printer name is *printer1*, the device pathname is */dev/tty1a*, and it will be the default printer for your system:

Before you give the **lpinit** command, you must make sure that the line to the printer is working correctly. To check this, you should connect the printer to your system, turn it on, and send text from the system to the port that is connected to the printer. For example, the command:

```
cat /etc/motd > /dev/tty1a
```

should cause a copy of the */etc/motd* file to print on your printer. If this test fails, you should check and make sure that:

- The printer is correctly connected to the computer.
- The connecting cable is in good condition.
- Your printer is powered on.
- The port is recognized by XENIX.

lpinit will not configure the printer correctly if the line is not working. When you are sure that the computer can communicate with the printer, perform the following steps:

1. Enter the command:

   ```
   /etc/lpinit
   ```

2. 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.

3. 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.

4. Next you are prompted:

```
Are you adding a (P)arallel or (S)erial printer?
```

   Enter "P" or "S" and press RETURN.

5. *lpinit* displays a list of available parallel or serial ports.

   Enter your choice and press RETURN.

6. **lpinit** displays the following message:

```
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, *printer1*, and press the **RETURN** key.

7. **lpinit** displays the following message on your screen:

```
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.
```

8. 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:
```

Enter one of the options, then press **RETURN**.

9. After you have finished responding to these questions, **lpinit** displays the following message:

```
destination "printer1" now accepting requests
printer "printer1" now enabled
```

Then, **lpinit** automatically gives the instructions to enable *printer1* and allow it to accept print requests.

10. After enabling *printer1*, **lpinit** prompts you 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 *printer1* or not.

11. 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/printer1.
```

After you have responded to these questions, **lpinit** starts **lpsched**, and users can begin printing files on the new printer.

You can also add printers to your system using the **lpadmin** command discussed in "Installing a Printer." 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 "Stopping the Print Spooling Daemon," "Accepting and Rejecting Print Requests," and "Enabling and Disabling Printers."

7.12.2 Stopping the Print Spooling Daemon: lpsched

The spooling daemon, **lpsched**, routes print requests through the correct printer interface program and then to the lineprinter. No printing can be done on your system unless **lpsched** is running. The program **lpsched** starts automatically each time your XENIX system is restarted. 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 (**lpinit** automatically stops and restarts **lpsched**).

This section explains how to find out whether or not **lpsched** is running, how to stop and restart it, and how to recreate it if necessary.

To find out whether or not **lpsched** is running, enter:

```
lpstat -r
```

The system responds with a message indicating whether **lpsched** is either running or not.

To shutdown the scheduler, **lpsched**, enter:

```
/usr/lib/lpshut
```

lpsched stops running and all printing stops as well. Printing requests stopped in the middle of printing reprint when **lpsched** starts again.

After you have finished configuring the printers, you should restart **lpsched**. To do this, enter:

```
/usr/lib/lpsched
```

lpstat -r should confirm that **lpsched** is running.

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.

For more information on **lpsched**, see **lpsched**(C), XENIX *Reference*.

7.12.3 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 boots. This is done on the first open only so that printers with large text buffers will not be 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. Login 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. Use *lpinit* to associate one of the parallel *init* devices (lp01, 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 doesn't have a large internal text buffer) you can use the */dev/lp0i* device all the time.

The **lp**(C) command now sends an **init** every time a file is sent to the printer.

Some printers do not have newline/carriage return mapping. If your printer needs to have newlines mapped to newline/carriage returns, specify the *crnlmap* filter when you set up the printer using the **lpinit**(C) command. When you are prompted for the type of printer (dumb printer, Imagen laser printer (parallel interface), Imagen laser printer (serial interface), or other), enter "other". You are then prompted for the pathname of the interface program. The printer filter file is found in */usr/spool/lp/model/crnlmap*.

7.12.4 Moving Requests Between Printers: lpmove

You can move print requests between printing destinations by using the command **lpmove**. This command does not move print requests while the scheduler, **lpsched**, is running. To stop **lpsched**, see "Stopping the Print Spooling Daemon." **lpmove** will move individual print requests by request ID, or all requests waiting to be printed on a particular printer.

For example, to move a request with a request ID of *quick-532* to a printer named *slow*, enter:

```
/usr/lib/lpmove  quick-532  slow
```

The print request now has a new request ID: *slow-532*.

To move all requests on a printer named *quick* to *slow*, enter:

```
/usr/lib/lpmove  quick  slow
```

For more information on **lpmove**, see **lpsched** (C), in the XENIX *Reference*.

7.12.5 Accepting and Rejecting Print Requests: accept

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. The printer, however, will not begin printing the requests until the **enable** command is given. If you added a printer to your system using the **lpinit** command, these steps were automatically performed. For information on **enable**, see "Enabling and Disabling Printers."

For example, to have print requests accepted for a class of printers named *class1*, enter:

```
/usr/lib/accept  class1
```

If you want to prevent requests from being routed to a printer, you can use the **reject** command. The **-r** options allows you to send users a message explaining why a printer is out of service.

For example, to prevent printing requests, from being routed to a printer, *printer4*, because of repairs, enter:

```
/usr/lib/reject -r"printer4 needs repair" printer4
```

A user who requests a file to be printed on *printer4* will receive the following message:

```
lp:can not accept requests for destination "printer4"
  --printer printer4 needs repair
```

To find out the acceptance status of printing destinations, enter:

```
lpstat -a
```

For more information on **lpstat**, see "Using the Lineprinter" in the *Introduction to XENIX* . For more information on **accept/reject**, see **accept**(C), in the XENIX *Reference*.

7.12.6 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 in order for the files to be printed, the **enable** command must be given as well.

For example, to enable a printer named *daisy*, enter:

```
enable daisy
```

You can disable printers with the **disable** command. The scheduler, **lpsched**, will not send printing requests to disabled printers regardless of their acceptance status. The **-r** options allows you to send a message to users explaining why a printer has been 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** will receive the following message:

```
printer "laser" disabled since Dec 5 10:15
paper jam
```

For more information on these two commands, see **enable**(C) and **disable**(C) in the XENIX *Reference*.

7.12.7 Printer Interface Programs

Each printer on your system must have a printer interface program. This can be a shell script, C program, or any other executable program. Your XENIX system provides a model interface program. It is written as a shell script and can be found in */usr/spool/lp/model*. You can use this program as is, modify it, or write your own interface program.

If you want to write or modify a printer interface program, the following information may be helpful.

When **lpsched** routes a printing request to a printer *P*, */usr/spool/lp* invokes the interface program for *P* as follows:

interface *P id user title copies options file*

These flags are:

interface	the directory which contains executable copies of interface programs
P	the interface program being executed
id	the request id returned by **lp**
user	the login name of the user who made the request
title	an optional title given by the user
copies	the number of copies requested
options	a list of printer dependent options separated by blanks
file	the full pathname of a file to be printed

When the interface program is started, its standard input comes from */dev/null* and both standard output and standard error output are directed to the printer's device. 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 may 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 that printer's device must be open for reading -- take the standard input for the **stty** command from the device.

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. Interface programs should examine this file and behave accordingly.

After printing is completed, the interface program should exit with a code showing that the print job was successful. Exit codes are interpreted by the printer scheduler, **lpsched**, as follows:

Exit Code	Meaning to **lpsched**
0	Print job was successful
1 to 127	**lpsched** found a problem while printing this particular request, for example, too many unprintable characters. This problem will not affect future printing requests. **lpsched** notifies users by mail that there was an error in printing the request.
greater than 127	These codes are reserved for internal use by **lpsched**. 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.

For more information on printer interface programs, see **lpadmin**(C), in the XENIX *Reference*.

7.13 Adding Additional 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.

2. Install additional memory according to the manufacturer's instructions. Make sure you have set all switches as noted in the instructions.

3. Boot XENIX . The boot screen details how the additional memory has affected your system.

4. Some features of XENIX 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. Since 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 XENIX allows.

The number of buffers may also be unchanged. Since 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 XENIX allows.

If the maximum user process size is unchanged, then it is now limited by the size of the *swap* file system instead of the amount of internal memory. You can:

- Reinstall XENIX and increase the size of the *swap* file system. See Chapter 2, "Installation Procedure," of the *Installation Guide* for details on reinstallation.
- Change the process so that it runs without being swapped. Refer to **proctl** (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 to XENIX , XENIX displays the message:

```
panic: parity
```

You then see the software reboot message:

```
** Normal System Shutdown **

      ** Safe to Power Off **
                - or -
  ** Press Any Key to Reboot **
```

If the system repeatedly panics from parity errors, consider replacing the memory chips.

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.

7.14 Adding a Hard Disk

You can give the system extra room for storing users' files and directories by adding a hard disk to the system. This is often the only remedy for a system that has one hard disk and suffers from chronic lack of space. See the *Release Notes* for a list of hard disks compatible with the current XENIX release.

You can only have one disk controller card. Software support is now provided for hard disks that do not have matching entries in the ROM tables. Switch settings on the disk controller card may need to be changed. Check your hardware manual for the hard disk drive and the computer for this information.

Before adding the new disk, you must know how to connect it to the computer. Connecting the hard disk is explained in the hardware manual provided with the disk.

Make sure the second drive passes the manufacturers diagnostics before running XENIX . If it does not pass the diagnostic tests, you cannot use it with XENIX.

This is an outline of the procedure to add another hard disk:

- Connect the hard disk, then boot the system and enter system maintenance mode.

- Use the **/etc/mkdev** program. **mkdev** executes **hdinit** which, in turn, runs four interactive programs: **dkinit** to inititalize the hard disk characteristics, **fdisk**(C), to separate the disk into MS-DOS and XENIX areas (the DOS area is optional), **badtrk**(M), to identify and map any bad areas on your disk, and **divvy**(C), to partition your disk into file systems.

Note

In the steps outlined below, you are prompted to respond to a variety of prompts. Although it is not always designated in this documentation, remember to press **RETURN** (or **ENTER**) after you have typed each response. XENIX waits indefinitely until this is done.

These are the steps to add another hard disk with one XENIX file system and no DOS area:

1. Connect the hard disk, then boot the system and enter system maintenance mode.

2. When you are in system maintenance mode, enter:

   /etc/mkdev hd

3. The first utility invoked by **hdinit** is **dkinit**, which sets parameters for non-standard hard disks. You see the menu

   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:

   If you have a standard hard disk, type "q" followed by <RETURN>. This selects the default parameters for your hard disk. Unless you know that your disk is non-standard, assume that it is standard and continue.

   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	*value*
2. Heads	*value*
3. Write Reduce	*value*
4. Write Precomp	*value*
5. Ecc	*value*
6. Control	*value*
7. Landing Zone	*value*
8. Sectors/track	*value*

When you see the display, *"value"* is replaced with the default value for that variable.

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:

If you enter "q", you see the first menu again. If you enter any of "1" - "8", you see the prompt:

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

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

After the table values are displayed or modified, the initial menu is again displayed and you must type "q" to exit **dkinit.** Exiting **dkinit** by typing "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 before typing "q" to exit **dkinit.**

dkinit is used to support non-standard hard disks. The normal user, while adding a second hard disk, will simply enter "q" when confronted with the **dkinit** initial menu. When the **dkinit** program exits, the **hdinit** program invokes **fdisk**(C), which partitions the hard disk.

4. When **fdisk** starts, you see this menu:

    Select one of the following options or 'q' to exit the program

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

    Enter your choice or 'q' to quit:

    If you want to use your whole disk for XENIX , enter '2', then press **RETURN**. If your hard disk already contains valid partitions, **fdisk** displays the current partition table followed by the message:

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

If you would like XENIX to occupy the whole disk, enter 'y' and press **RETURN.**

If no partitions were previously installed on your hard disk, **fdisk** will allocate the entire disk for XENIX without displaying the above warning.

Note that **fdisk** reserves the first track for **masterboot** and the last cylinder of the hard disk for disk diagnostics. Thus your partition begins on track 1 instead of track 0 and it ends on track 1219 instead of track 1223.

You see the partition table again, with the following changes:

Current Hard Disk Drive: /dev/rhd01

Partition	Status	Type	Start	End	Size
1	Active	XENIX	1	1219	1218

Total disk size: 1224 (5 tracks reserved for masterboot and diagnostics)

Press <RETURN> to continue.

After you press **RETURN**, you see the main **fdisk** menu, shown above. Note that the XENIX partition must be active before exiting **fdisk.** Type 'q' to exit **fdisk** and continue with the installation.

For more information on having DOS and XENIX on your hard disk, see **fdisk**(C).

If you had a large portion of the disk already allocated to DOS, you must run DOS to deallocate this area. See **fdisk**(C) for more information on sharing disks between DOS and XENIX. No matter what configuration you produce with **fdisk**, the active partition must be the XENIX partition when you are through.

5. The next menu is from the bad track mapping program, **badtrk**. It displays this menu:

   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**. If **badtrk** thinks that the table may have been changed, you are prompted if you want to update this device with a new table. You should answer 'y'. Next, you are prompted if you want to salvage any data on the bad tracks, and warned that this may take a long time. You should answer 'n' unless you have data on the bad tracks that you need to restore and there is no other way (i.e. backups) to recover the information, in which case you should respond 'y'.

You see the following submenu:

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

Enter the number corresponding to the type of scan you wish to perform. If this is a *fresh installation*, (that is you are installing XENIX for the first time or performing a complete reinstallation), you probably want to select option '1'.

If you choose option '2' or '3', you are then prompted to specify the area you want **badtrk** to scan.

After you have selected the area you want scanned, you are given the choice:

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

Option '2' should be selected if this is a fresh installation. Once you respond to the above prompt, you are prompted:

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

If you respond 'n', a non-destructive, readonly scan begins. However, for your fresh installation, respond 'y'. You are warned:

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

Respond 'y' for your fresh installation and you see the following message:

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

After you have responded to the above prompts, the program scans the active partition of the hard 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 "q" at any time interrupts the scan. You will then be 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/head conventions.

If your disk comes with a flaw map, cross-check your map against the results of the disk scanning. Because most disk flaws are marginal or intermittent, any flaw map will almost certainly contain more flaws than the scanning process will reveal. Depending on the format of your flaw map, select either option '3', "Add Entries to Current Bad Track Table by Cylinder/Head Number", or option '4', "Add Entries to Current Bad Track Table by Sector Number", and enter the flaws, one per line. Enter 'q' when you are finished.

Exit **badtrk** by entering 'q' at the main menu.

The program now displays the number of identified bad tracks. You are prompted to enter the number of tracks to be allocated as replacements for bad tracks. You will see a recommended number of tracks to allocate as space for replacement. This number is based on the number of known bad tracks and an allowance for tracks that will go bad in the future. You should choose to allocate at least as many tracks as recommended. Please make this choice carefully, because if you want to change this amount later, you will have to reinstall XENIX.

Next, **badtrk** prompts:

> Do you want to update this device with the new table?

To save the changes made to the bad track table, enter 'y'. If an irrevocable mistake has been made in the bad track table at some time during this process, enter 'n', modify the bad track table to contain the desired entries, and then enter 'q' at the main menu to return to the prompt displayed above.

You are prompted:

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

Generally you should respond 'n' to this prompt. However, if you have not made a backup of needed data on the bad tracks and there is no other way of recovering this information, you may want to respond 'y', keeping in mind that this process may take a long time.

6. Next, you see a prompt from **divvy**. The **divvy** program divides a partition into file systems. You can create up to seven divisions on a single partition, and name them anything you like.

   If the hard disk you are installing is 20 megabytes or larger, you will be prompted for the number of file systems you want to create. Press **RETURN** to use the default value of one file system. Smaller hard disks automatically default to creating one file system.

   You are next prompted for block control of your hard disk. You see:

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

   Enter 'y' and press RETURN. This allows you to create up to seven file systems on a single XENIX partition, and assign specific names to whatever file systems you create. You must enter 'y' if you prompted to create more than 1 file system in the previous prompt.

   You see the main **divvy** menu and a display that shows you how your disk is divided:

Name	New File System?	#	First Block	Last Block
root	no, exists	0	0	24344
swap	no, exists	1	24345	25345
	no	2	-	-
	no	3	-	-
	no	4	-	-
	no	5	-	-
recover	no,exists	6	25346	25546
hd0a	no, exists	7	0	25546

n[ame]	Name or rename a division.
c[reate]	Create a new filesystem on this division.
p[revent]	Prevent a new file system from being created on this division.
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:

Each row in the **divvy** table corresponds to a file system. For example, in this table, the name of file system 0 is "root".

Use option 'n' to change the name of a filesystem. You could name it *u* (for "user"), for example.

Do not change the configuration of file system 7. It is reserved for internal use by XENIX.

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

If you have a large file system, you may be prompted if you want a scratch device to be created for you. You should answer yes.

For more information on **divvy**(C) see the XENIX *Reference*.

7.14.1 Mounting Another File System

To use an additional disk, or create a second, mounted file system, enter this command:

```
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 (used by **fsck** to recover files if the file system is corrupted).
- Mounts the device (*/dev/u*) on */u*, cleans it, and unmounts it.
- Removes write permissions on the directory */u* for group and all other users except root.
- Creates files in the */u/lost+found* directory, then removes them. This allocates inodes for the directory, so that if the file system is corrupted and runs out of inodes, **fsck**(C) is still able to recover files.
- Adds the following line to */etc/checklist*:

  ```
  /dev/u
  ```

- Modifies */etc/default/filesys* so that the new file system is automatically mounted, checked and cleaned when appropriate.

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

```
mount fs /dev/u /u
umount fs /dev/u /u
```

7.14.2 Using A Second File System

You can access the files on your new file system by first mounting it in the appropriate directory (we used */u* as an example in the preceding section). After you mount the file system, all directories and files on it are usable just as any others on the system.

If you want to have user accounts on your mountable file system, change the variable "HOME" in the file */etc/default/mkuser*. The "HOME" variable defines the directory where user accounts are placed. This variable is used by the **mkuser**(C) program, which adds new users to the system.

Edit the file */etc/default/mkuser*. There is a line which may look like this:

```
HOME=/usr
```

Change */usr* to the name of the directory where you want to place user accounts. In our example, this is */u*. The line should now read:

```
HOME=/u
```

Now, whenever you run the **mkuser**(C) script to add a new user, that user's account will be in */u*.

If there are already users on the system, and you want to move their accounts to the new file system, you can use the **copy**(C) command to copy their accounts to the new file system. 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 file system to another:

1. Make sure the new file system is mounted. Also, be sure you are either in system maintenance mode, or 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:

   lc

   You see a list of account names, for example:

alisonb	dean	jerrys	sams
blf	gregt	lost+found	tammyr
buckm	jeffj	pj	vicki

4. Enter:

   copy -orm /usr /u

   and press **RETURN**.

5. When the **copy** command has 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 file system.

7. Change the home directory for each user as listed in */etc/passwd*. An example entry in */etc/passwd* might be:

   alisonb:CoHiKNs.:271:104:Alison Berry:/usr/alisonb:/bin/csh

You see one such line for every user on your system. Change the field:

```
:/usr/alisonb:
```

to match the user's new home directory:

```
:/u/alisonb:
```

Do this for every user whose home directory has changed.

7.15 Creating Boot and Root Floppy Disks

mkdev(C) provides a utility to create bootable and root system floppy disks. **mkdev** invokes *fdinit*, a menu-driven program which allows the user to select the disk format and filesystem type. There are three basic types generated: boot and root on a single disk (96 tpi only), boot and root pair (48 tpi), or filesystem only (which simply invokes **mkfs**(C)). The formats supported are: 48 tpi, 96 tpi-15 sectors/track, in 5 1/4 and 3 1/2 inch formats. To create the floppies, enter:

```
/etc/mkdev fd
```

and you will see the following display:

```
Choices for type of floppy filesystem.

        1. 48 tpi, double sided, 9 sectors per track
        2. 96 tpi, double sided, 15 sectors per track

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

Enter the correct type and press RETURN. You are then prompted for the disk:

```
Insert a xx tpi floppy into drive 0. Press Return to continue.
```

Insert your floppy and press RETURN. The program responds with another menu:

```
Choices for contents of floppy filesystem.

        1. Filesystem only
        2. Bootable only
        3. Root filesystem only
        4. Root and Boot (only available for 96tpi floppy)

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

Select the appropriate filesystem and press RETURN. **fdinit** generates the filesystem and will display the following message when complete:

```
xx tpi filesystem floppy complete.
```

Replace this Page
with Tab Marked:

System Problems

Chapter 8

Solving System Problems

8.1 Introduction

This chapter explains how to solve problems that affect the operation of the system. The problems range in complexity from how to fix a nonechoing terminal, to how to restore lost system files.

8.2 Restoring a Nonechoing Terminal

A nonechoing terminal is any terminal that does not display characters entered at the keyboard. This abnormal operation can occur whenever a program stops prematurely as a result of an error, or when the user presses the **BREAK** key.

To restore the terminal to normal operation, follow these steps:

1. Press the Ctrl-j key. The system may display an error message. If it does, ignore the message.

2. Enter:

   ```
   stty sane
   ```

   and press the Ctrl-j key. The terminal does not display what you enter, so enter accurately.

After pressing Ctrl-j, the terminal should be restored and you may continue your work.

8.3 Resolving Dialer Difficulties

If the dialer program has aborted abnormally, the serial line you are using will remain enabled for dial out. (See **dial**(M) in the XENIX *Reference Manual.*) To check to see if a line is still in a dial out state, use this command:

```
who -a
```

If the serial line is still in a dial-out state, (the entry for the line will read: **DIALOUT**) use the **ungetty** command. For example, to correct *tty1a*, enter this command and press **RETURN**:

```
/usr/lib/ungetty -r tty1a
```

The line should be now be enabled as a dial-in line.

8.4 Solving Lineprinter Problems

No printing can be done on the lineprinter spooling system unless the print scheduler, **lpsched**, is running. To check the status of **lpsched**, enter:

```
lpstat -r
```

To restart **lpsched**, enter the following on two lines:

```
/usr/lib/lpshut
/usr/lib/lpsched
```

Calling the file */ust/lib/lpshut* cleans the system and */usr/lib/lpsched* start it up again.

Access to files and directories in */usr/spool/lp* by **lp** can be another source of spooling problems. You can check the **lpsched** log file, */usr/spool/lp/log*. This is a record of the print scheduler's activity and errors. If **lpsched** refuses to run or a printer refuses to print, check to make sure that:

- The printer is enabled; see **lp**(C) in the XENIX *Reference Manual.*

- The files and directories in */usr/spool/lp* are readable and writable by **lp**.

For more information on the lineprinter spooling system, see the section "Adding A Lineprinter" in Chapter 7.

8.5 Stopping a Runaway Process

A runaway process is a program that cannot be stopped from the terminal at which it was invoked. This occurs whenever an error in the program "locks up" the terminal, that is, prevents anything you enter from reaching the system.

To stop a runaway process, follow these steps:

1. Go to a terminal that is not locked up.

2. Log in as the super-user.

3. Enter:

   ps -a

   and press the **RETURN** key. The system displays all current processes and their process identification numbers (PIDs). Find the PID of the runaway program.

4. Enter:

   kill *PID*

   and press the **RETURN** key. The *PID* is the process identification number of the runaway program. The program should stop in a few seconds. If the process does not stop, enter:

   kill -9 *PID*

   and press the **RETURN** key.

The last step is sure to stop the process, but may leave temporary files or a nonechoing terminal. To restore the terminal to normal operation, follow the instructions in the section "Restoring a Nonechoing Terminal" in this chapter.

8.6 Replacing a Forgotten Password

The XENIX operating system does not provide a way to decipher an existing password. If a user forgets his password, the system manager must change the password to a new one. To change an ordinary user password, follow the instructions in the section "Changing a User's Password" in Chapter 3.

8.7 Removing Hidden Files

A hidden file is any file whose name begins with a dot (.). You can list the hidden files in a directory by entering:

lc -a

and pressing the **RETURN** key.

You can remove most hidden files from a directory by entering:

```
rm .[a-z]*
```

and pressing the **RETURN** key. Remaining files can be removed individually.

8.8 Restoring Free Space

The system displays an "out of space" message whenever the root directory has little or no space left to work. To restore system operation, you must delete one or more files from the root directory. To delete files, follow the steps outlined in the section "Maintaining Free Space" in Chapter 5.

8.9 Restoring Lost System Files

If a system program or data file is accidentally modified or removed from the file system, you can recover the file from the periodic backup disk with the **sysadmin** program. To restore the files, follow the instructions in the section "Restoring a Backup File" in Chapter 6.

8.10 Restoring an Inoperable System

On very rare occasions, one or more of the critical XENIX system files may be accidentally modified or removed, preventing the system from operating. In such a case, you must reinstall the XENIX system, and restore user program and data files from backup disks. To reinstall the system, follow the instructions in the XENIX *Installation Guide*. To restore files from backup disks, follow the instructions in the section "Restoring a Backup File" in Chapter 6.

8.11 Recovering from a System Crash

A system crash is a sudden and dramatic disruption of system operation that stops all work on the computer. System crashes occur very rarely. They are usually the result of hardware errors or damage to the root file system which the operating system cannot correct by itself. When a system crash occurs, the system usually displays a message explaining the cause of the error, then stops. This gives the system manager the chance to recover from the crash by correcting the error (if possible), and restarting the system.

A system crash has occurred if the system displays a message beginning with "panic:" on the system console, or the system refuses to process all input (including INTERRUPT and QUIT keys) from the system console and all other terminals.

To recover from a system crash, follow these steps:

1. Use the error message(s) displayed on the system console to determine the error that caused the crash. If there is no message, skip to step 3.

2. Correct the error, if possible. A complete list of error messages and descriptions for correcting the errors is given in **messages**(M) in the XENIX *Reference Manual.* (Even if the problem cannot be located or corrected, it is generally worthwhile to try to restart the system at least once by completing the remaining steps in this procedure.)

3. Turn off the computer and follow the steps described in Chapter 2, "Starting the System," to restart the system.

4. If the system will not restart, or crashes each time it is started, the operating system is inoperable and must be reinstalled. Follow the procedures described in the XENIX *Installation Guide* to reinstall the system and in Chapter 6, "Backing Up File Systems," to restore user's files.

5. If the system cannot be started from the "Boot" disk in the distribution set for installation, the computer has a serious hardware malfunction. Contact a hardware service representative for help.

8.12 Mapping a Bad Track

Bad tracks on the hard disk are mapped during the XENIX installation procedure. This allows XENIX to avoid those areas of the disk that cannot be read or written. However, hard disks can develop bad tracks after XENIX is installed and running. If this occurs, the **badtrk**(M) utility should be run by the super-user (root) enabling XENIX to avoid the new bad track(s). **badtrk** must be run in single-user mode. Use **/etc/shutdown su** to enter single-user mode from multi-user mode (see **shutdown**(C)).

badtrk is a menu-driven utility for viewing, adding, or deleting entries to the bad track table. See **badtrk**(M) in the XENIX *Reference Manual* for more on its options and their use.

8.13 Changing XENIX Initialization

One common problem is adapting the system initialization to suit your system environment. This problem occurs whenever you have added new devices such as terminals or disk drives to the system, and wish these devices to be automatically enabled or mounted whenever you start normal system operation. You can adapt system initialization by modifying the system initialization files.

The XENIX initialization files contain XENIX commands and/or data which the system reads at system startup or whenever a user logs in. The files typically mount file systems, start programs, and set home directories and terminal types. The initialization files are named */etc/rc*, *.profile*, and */etc/motd*.

The system manager may modify these files to create any desired initial environment. The files are ordinary text files and may be modified using a text editor such as **ed** (see the XENIX *User's Guide*). Note, however, that the */etc/rc* and *files contain XENIX commands and comments, and have the command file format described in Chapter 4, "The Shell," in the XENIX User's Guide.*

8.13.1 Changing the /etc/rc File

The */etc/rc* file contains XENIX system initialization commands. The system executes the commands at system startup. The commands display a startup message, start various system daemons, and mount file systems. You can display the contents of the file with the **more** command. Enter:

```
more /etc/rc
```

and press the **RETURN** key.

You may change the contents of the file so that the system executes any set of commands you wish. For example, if you want the system to automatically mount a new file system, simply append the appropriate **mount** command in the file. The system will execute the command on each startup.

To append a command to the file, follow these steps:

1. Log in as the super-user.

2. Invoke a text editor and specify the */etc/rc* as the file to be edited.

3. Locate the place in the file you wish to insert the command (e.g., if the command mounts a file system, insert it with other mounting commands).

4. Insert the command on a new line. Make sure you enter the command correctly. The system rejects any incorrect commands and the commands that follow it when the file is read at system startup.

5. Exit the editor.

No other changes to the file are required. Be careful not to delete any commands already in the file unless you are sure they are not needed.

8.13.2 Changing the .profile Files

The *.profile* files contain commands that initialize the environment for each user. The commands in the file are executed whenever the user logs in. The file usually contains commands that set and export various system variables (e.g., TERM, PATH, MAIL). These variables give the system information such as what terminal type is being used, where to look for programs the user runs, where to look for the user's mailbox, what keys to expect for the "kill" and "backspace" functions, and so on (see Chapter 4, "The Shell," in the XENIX *User's Guide*).

There is one *.profile* file for each user account on the system. The files are placed in the user's home directory when the account is created. An ordinary user may modify his own *.profile* file or allow the system manager to make modifications. In either case, the file can be edited like the */etc/rc* file by using a text editor. Commands can be added or removed as desired.

8.13.3 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 XENIX system. It can be modified to include messages such as a reminder to clean up directories, a notice of the next periodic backup, and so on.

The */etc/motd* file is an ordinary text file, so you can change the message by editing the file with a text editor. One common change is to include a reminder to delete unused files in order to preserve disk space. In general, you should limit the size of the file to include no more than a screenful of information.

Replace this Page
with Tab Marked:

Micnet

Chapter 9

Building a Micnet Network

9.1 Introduction

A Micnet network allows communications between two or more independent XENIX systems. The network consists of computers connected by serial communication 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**, **rcp**, and **remote** commands. These commands pass information such as mail, files, and even other commands, from one computer to another.

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 choose machine names and aliases
- How to draw the network topology map
- How to assign serial lines
- How to create the Micnet files
- How to distribute the Micnet files
- How to test the Micnet network

9.2 Planning a Network

To build a Micnet network, the **netutil** program requires that you provide the names of the computers that will 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 will use the network, and what aliases (if any) they will be 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 will be required to supply. To help you make these lists, the following sections suggest ways to plan a network.

9.2.1 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.

9.2.2 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 "serial."

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 serial 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 may be strictly star, strictly serial, or a combination of star and serial topologies. The only restriction is that no network may form a ring. For example, you cannot close up a serial 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 serial topology distributes the communication 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 serial 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 may change the network at any time.

9.2.3 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 three computers might look like this:

```
a ---------- b ---------- c
```

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.

9.2.4 Assigning Lines and Speeds

Once you have made the topology map, you can decide which serial lines to use. Since every connection between computers in the network requires exactly two serial lines (one on each computer), you need to be very careful about assigning the lines. Follow these steps:

1. Make a list of the serial lines (TTY lines) available for use on each computer in the network. You can display a list of the serial lines on a computer by displaying the file */etc/ttys*. A line is available if it is not connected to any device such as a terminal or modem.

2. Using the topology map, first pick a computer, then assign one (and only one) serial line to each connection shown for that computer. The serial lines must be from the list of available lines for that computer. No line may be assigned more than once. For example, if computer *a* has only one available serial line (tty1a), then the topology map should look like this:

```
a ---------- b ---------- c
tty1a
```

3. Repeat step 2 for all computers in the topology map. Make sure that each connection is assigned a line and that no two connections on any given computer have the same line. When finished, the map should look like this:

```
          a ---------- b ---------- c
       tty1a        tty2a  tty3a  tty4a
```

   If a computer does not have enough available serial lines 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.

4. Using the topology map, assign a serial line transmission speed for each computer pair. The speed must be within the normal range for XENIX 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 may then increase the speed if you find the hardware can support it.

5. 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 will use this list when installing the network.

9.2.5 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.

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.

9.3 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 may start the network and use it to communicate between computers.

The following sections describe how to build the network.

9.3.1 Creating the Micnet 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

   and press the **RETURN** key. The program displays the network utility menu. The **install** option is the first item in the menu.

3. Enter the number **1**, and press the **RETURN** key. The program displays 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 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 may 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 a:
   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 may 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 5 for all remaining computers. As the program prompts for each new set of connections, it will show 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 a:
   ```

9. Using the list of serial line assignments you created when planning the network, enter the serial line name or number (for example, tty03 or 3) for the first computer in the pair and press the **RETURN** key. The program displays the message:

   ```
   Tty on b:
   ```

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 step 8 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 move to the next step.

The program displays the message:

```
Enter the names of users on each machine:

For machine a:
  Users on a:
```

14. Enter the login name of a user on the given computer, then press the **RETURN** key. You may 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 13 and 14 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 17 and 18 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.

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 may use the **save** option to copy the Micnet files to floppy disk.

9.3.2 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 Chapter 4). 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 the number 2, and press the **RETURN** key. The program displays the message:

   Save to /dev/fd*x* (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, 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 (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.

9.3.3 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 the number 3, and press the **RETURN** key. The program displays the message:

   Restore from /dev/fd*x* (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 have been copied, you may start the network with the **start** option.

9.4 Starting the Network

Once the Micnet files have been 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 may 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 command:

netutil start

in the system initialization file, */etc/rc*, of each computer. To add this command, use a text editor as described in the section "Changing the /etc/rc File" in Chapter 8. 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 log in and log out message to the system *LOG* file each time you start and stop the network. This means you will need to periodically clear the file. See the section "Clearing Log Files" in Chapter 5.

9.5 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 will need to know how to use the **mail** command (see Chapter 3, "Mail," 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.

9.5.1 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. Follow these steps:

1. Choose a computer.

2. Log in as the super-user.

3. Use the **mail** command (see the XENIX *User's Guide*) and the **all** alias to mail the message:

   Micnet test

   to all users in the network.

4. Check the mailboxes of each user in the network to see if the message was received. To check the mailboxes, log in as the super-user at each computer and use the **cat** command to display the contents of each user's mailbox.

   The name of each user's mailbox has the form:

   /usr/spool/mail/*login-name*

   where *login-name* is the user's login name.

The network is properly installed when all users have received the message. If the users at one or more computers fail to receive the message, the computers are not properly connected to the network. To fix the problem, you need to locate the computer which has failed to make a connection. The next section explains how to do this.

9.5.2 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

and pressing 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" which 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/tty02
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 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 will contain a "handshake" message if the connection between the computer pair has been established. The message:

first handshake complete

is added to the file on a successful connection. If the message is not present, make sure that the network has been 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 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 which detected the error. For example, the message:

rsync: bad Probe resp: 68

shows that the *rsync* subroutine received a bad response (character 68 hexadecimal) from the other computer. You may 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, elasped time, and time of day of the transmission. For example, the entry:

rx: 0c 01 22:33:49

shows that 12 characters (0c hexadecimal) were received (*rx*) at 22:33:49. The elasped time for the transmission was 1 second. You can use the records to see if messages are actually being transmitted.

9.5.3 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.

9.5.4 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 an alias), you can modify the Micnet files directly with a text editor. The files and their contents are described in detail in the M section of the XENIX *Reference Manual*.

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 have been 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.

Note that changes to the *aliases* file will not be incorporated into the system until the **aliashash** program is executed. This program produces the *aliases.hash* file needed by the network to resolve aliases. See **aliashash** (M) in the XENIX *Reference Manual* for a description of this command.

9.6 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 XENIX 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 XENIX *User's Guide* for complete details.

2. Add the entry:

   uucp:

   to the *maliases* 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 *maliases* file. The *machine-name* must be the name of the computer on which the **uucp** system is installed. One may 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

will send 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.

Replace this Page
with Tab Marked:

Appendices

Appendix A
XENIX Special Device Files

A.1 Introduction

This appendix contains information needed to create file systems and add terminals to the XENIX system. For a full description of the special files mentioned here, see section HW of the XENIX *Reference Manual* found at the back of this volume.

A.2 File System Requirements

Many of the file system 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.

A.3 Special Filenames

A special filename is the name of 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**, **mount**, and **df** to specify the device containing the file system to be created, mounted, or searched.

The following table lists the special filenames and corresponding devices, for hard and floppy disk drives on a typical computer.

Device Special Filenames – Disks

Filename	Disk Drive
/dev/fd0	Floppy Drive 0
/dev/fd1	Floppy Drive 1
/dev/hd00	Entire hard disk
/dev/root	Root file system
/dev/usr	User file system

A.4 Block Sizes

The block size of a disk is the number of blocks of storage space available on the disk, where a block is typically 512 or 1024 bytes of storage. Refer to the **machine**(HW) manual page or use **cmchk**(C) to determine the size of blocks on your system. Many commands require input that defines the number of blocks to be operated on. Other commands report disk space in terms of 512 byte blocks, in particular **df**, **du**, **ls**, **lc**, and **find**. A 500 byte file on a 1024 byte block file system is reported as using 2 blocks by these utilities, as the file uses one system block which is equivalent to two 512 byte blocks. The size of a 10 megabyte hard disk in 1024 byte blocks is 9792. 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.

A.5 Gap and Block Numbers

The gap and block numbers are used by the **mkfs**, and possibly **fsck**, commands to describe how the blocks are to be arranged on a disk. The following table lists the gap and block numbers for the floppy and hard disks used with a typical computer.

Disks	Gap	Block
Floppy Disk, 48ds9	1	9
Floppy Disk, 96ds15	1	15
Hard Disk	1	34

The number of blocks can also be determined by multiplying the number of sectors per track (usually 17) by the number of heads on the hard disk, dividing by 2 (since there are 2 blocks per sector), and rounding off to the nearest integer.

A.6 Terminal and Network Requirements

The **enable** and **disable** commands are used to add and remove terminals on a system. The install option of the **netutil** program is used to build a network. The preceding commands and option require the names of the serial lines through which a terminal or network is to be connected. The following table 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 correpsonding to these serial lines can be found in the */dev* directory. Note that the files */dev/console* and */dev/tty02* through */dev/tty10* represent "hardwired" devices and are not available for connection to terminals or hardware. Also, refer to **serial**(HW) for more information on serial lines.

Filename	Line
/dev/tty1a	main serial line (without modem control)
/dev/tty2a	alternate serial line (without modem control)
/dev/tty1A	main serial line (with modem control)
/dev/tty2A	alternate serial line (with modem control)

Appendix B

XENIX Directories

B.1 Introduction

This appendix lists the most frequently used files and directories in the XENIX system. Many of these files and directories are required for proper XENIX operation and must not be removed or modified. The following sections briefly describe each directory.

B.2 The Root Directory

The root directory (/) contains the following system directories:

/bin	XENIX command directory
/dev	Device special directory
/etc	Additional program and data file directory
/lib	C program library directory
/mnt	Mount directory (reserved for mounted file systems)
/usr	User service routines (may contain user home directories)
/tmp	Temporary directory (reserved for temporary files created by 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.

B.3 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 following is a list of a few of the commands:

basename	echo	passwd	su
cp	expr	rm	sync
date	fsck	sh	tar
dump	login	sleep	restor
dumpdir	mv	stty	test

These commands and all others in the /bin directory are required.

B.4 The */dev* Directory

The */dev* directory contains special device files which control access to peripheral devices. All files in this directory are required, and must not be removed.

The following is a list of the files:

/dev/console	System console
/dev/lp	Lineprinter
/dev/mem	Physical memory
/dev/null	Null device (used to redirect unwanted output)
/dev/rXX	Unbuffered interface to corresponding device name
/dev/root	Root file structure
/dev/swap	Swap area
/dev/ttyXX	Terminals
/dev/tty	The terminal you are using

B.5 The *letc* Directory

The */etc* directory contains miscellaneous system program and data files. All files are required, but many may be modified.

The following program and data files must not be removed or modified:

/etc/mnttab	Mounted device table
/etc/mount	For mounting a file structure
/etc/mkfs	For creating a file structure
/etc/init	First process after boot

The following data files may be modified, if desired. No files may be removed.

/etc/passwd	Password file
/etc/rc	Bootup shell script
/etc/ttys	Terminal set up
/etc/termcap	Terminal capability map
/etc/motd	Message of the day

The data files in the directory */etc/default* contain default information which is used by system commands (see **default**(M)). The following data files may be modified. No files may be removed.

/etc/default/archive	**sysadmin**(C) default information
/etc/default/backup	**backup**(C) default information
/etc/default/boot	**autoboot**(M) information
/etc/default/cron	**cron**(C) default logging information
/etc/default/dumpdir	**dumpdir**(C) default information
/etc/default/filesys	**sysadmin**(C) default information
/etc/default/login	**login**(M) default information
/etc/default/lpd	**lp**(C) default information
/etc/default/micnet	**micnet**(M) default information
/etc/default/mkuser	**mkuser**(C) default information
/etc/default/msdos	**Location** of DOS disks (A:, B:,...)
/etc/default/passwd	**passwd**(C) default information
/etc/default/quot	**quot**(C) default information
/etc/default/restor	**restore**(C) default information
/etc/default/su	**su**(C) default information (Note that you must create this file yourself.)
/etc/default/tar	**tar**(C) default information

B.6 The */lib* Directory

The */lib* directory contains runtime library files for C and other language programs. The directory is required.

B.7 The */mnt* Directory

The */mnt* directory is an empty directory reserved for mounting removable file systems.

B.8 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 may remove any temporary file that does not belong to a running program.

B.9 The */usr* Directory

The */usr* directory contains the home directories of all users on the system. It also contains several other directories which provide additional XENIX commands and data files.

The */usr/bin* directory contains more XENIX commands. These commands are less frequently used or 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. Since messages in the file can accumulate rapidly, the file must be deleted periodically.

B.10 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 "Clearing Log Files" in Chapter 5). The following table lists the files (by full pathname) and their contents.

Filename **Description**

/etc/ddate
  Records date of each backup.

/usr/adm/pacct
  Records accounting information; grows rapidly when process accounting is on.

/usr/adm/messages
  Records error messages generated by the system when started.

/etc/wtmp

Records user logins and logouts.

/usr/adm/sulog

Records each use of the **su** command; grows only if option is set in the */etc/default/su* file. You must create */etc/default/su*. See **su**(C) in the XENIX *Reference*.

/usr/lib/cron/cronlog

Records each use of the *at* and *cron* commands.

/usr/spool/micnet/remote/*/LOG

Records transmissions between machines in a Micnet network. The * must be the name of a remote machine connected to the current machine.

For more information on these files, see the appropriate sections in the XENIX *Reference*.

Index

A

B

C

D

E

F

N

O

P

Q

R

T

U

W

X

Contents

Hardware Dependent **(HW)**

Name

intro – Introduction to machine related miscellaneous features and files.

Description

The hardware-dependent section (HW) contains information useful in maintaining the system. Included are descriptions of files, devices, tables and programs that are important in maintaining the entire system that are directly related to the kind of computer on which the system runs. This section is intended for use with the 86 family of Intel CPUs, specifically 8086, 8088, 80186, 80286 and 80386 based computers.

Name

8087

Syntax

8087
80287

Description

The 8087 is the INTEL math co-processor for the 8086. The 80287 is the INTEL math co-processor for the 80286. The kernel tests for the presence of an 8087 or 80287 at startup.

If your system has an 8087 or 80287, you must turn off a switch main system board in order to enable 8087 interrupts. Check you hardware manual to determine the proper switch and setting. If your system does not have an 8087, or the switch is on, the kernel will run a set of emulator routines which are much slower.

The C compiler available with the program development package generates the appropriate 8087 (or 80287) opcodes. C routines compiled with this compiler have run as much as 200 times as fast as the emulated code. In particular, the standard math library routines run considerably faster if you have an 8087 (or 80287).

The overflow, division by zero, and invalid operand exceptions return a SIGFPE signal. This signal can be caught. The rest of the 8087 and 80287 floating point exceptions (underflow, denormalized operand, and precision error) are masked.

Notes

The emulator returns meaningless information on divide by zero.

There is no obvious way to tell which 8087 (or 80287) exception generated the SIGFPE.

Name

boot – XENIX boot program.

Description

boot is an interactive program used to load and execute standalone XENIX programs. It is used primarily for loading and executing the XENIX kernel, but can load and execute any other programs that are linked for standalone execution. *boot* is a required part of the XENIX Operating System and must be present in the root directory of the root file system to ensure successful loading of the XENIX kernel.

The *boot* program is invoked by the system each time the computer is started. To restart the system without going through lengthy shutdown procedures, you can use the *reboot* command. This causes the system to reboot after shutting down without waiting for keyboard input. See *haltsys (C)* for more information.

For diskette boot, the procedure has three stages:

1. The ROMs load the boot block from sector 0 of the floppy, where sector 0 of the disk is the same as sector 0 of the filesystem.

2. The boot block-loads **/boot** from the floppy filesystem.

3. **/boot** executes and prompts the user.

For fixed disk boot, the procedure has five stages:

0 of the filesystem.
The ROMs load in the *masterboot* block from sector 0 on the hard disk.

2. The *masterboot* block then loads the partition boot block from sector 0 of the active partition (see *fdisk*(C)).

3. Then, boot1 is loaded from the next four tracks.

4. boot1 loads **/boot** from the XENIX file system.

5. **/boot** executes and prompts the user.

/boot and **/xenix** may lie on tracks that have been mapped by *badtrk(M)* .

The fixed disk boot procedure is invoked if the diskette drive is empty.

When first invoked, *boot* prompts for the location of a program to load by displaying the message:

XENIX System V

Boot
:

To specify the location of a program, a device and filename must be given. The filename must include the full pathname of the file containing the standalone program. You can display a list of the current allowable device names by typing a question mark (?).

The format for the device and pathname is as follows:

xx(m,o)filename
  or
xx(m)filename

where:
- xx = device name
  ('hd' for the hard disk or 'fd' for diskette device)
- m = minor device number
  (40 for the **root** filesystem on the hard disk)
- o = offset in the partition (usually 0). This is optional.
- filename = standard XENIX pathname. Must start with a slash if the program is not in the root directory.

All numbers are in decimal. See the manual pages for *hd*(HW) and *fd*(HW) for minor device numbers of these devices. Specifying the offset is optional. The location of the program to be loaded must always be entered first on the command line and be present if other *boot* options are specified either on the command line or in **/etc/default/boot.**

If you want *boot* to pause and wait for a <RETURN> before executing the program that it loads, enter the word "prompt" on the command line. For example, if you enter "prompt" and press <RETURN>, *boot* prints the following message and waits for you to press the return key again:

Loaded, press <RETURN>.

The prompt can be changed to another string as in this example:

prompt="change diskettes now"

boot loads **xenix** from the diskette, prints the message "change diskettes now", and waits for <RETURN> to be pressed. No other characters can appear between *prompt*, the "=" sign and the prompt string, although *string* may contain spaces. When you press <RETURN>, **xenix** will begin execution. "Prompt" can be set either on the command line or in **/etc/default/boot.** If a prompt is not specified, *boot* executes the loaded program without pausing.

If you have just loaded the *boot* program from the distribution diskette, simply press <RETURN> and *boot* defaults to the correct values.

To load XENIX from a hard disk, enter:

hd(40,0)xenix

To use the default boot string specified in **/etc/default/boot**, simply press <RETURN> when the system displays the boot prompt, and *boot* uses the values specified by DEFBOOTSTR in **/etc/default/boot**.

If nothing is typed after a short while and LOADXENIX is set to YES in the default *root* file system's **/etc/default/boot** file, *boot* times out and behaves as though a <RETURN> had been pressed, except that an "auto" is added to the boot string. *boot* proceeds through the boot procedure, and *init*(M) is passed a **-a** flag with no "prompt".

It is recommended that you install DOS on the hard disk before XENIX. See the manual page for *dos*(C). However, once you install DOS you can boot it at the XENIX "Boot" prompt by entering "dos".

During XENIX installation, a custom *masterboot* is placed on the hard disk. If a non-standard disk is specified, its parameters are stored and enabled in this *masterboot*.

Configuring The Kernel

boot passes any boot string typed at the boot prompt to the kernel, except for the "prompt" string.

The kernel reads the boot string to determine which peripherals are the root, pipe and swap devices. If no devices are specified in either the **/etc/default/boot** description or on the command line, the default devices compiled into the kernel are used.

Additional arguments in the boot string can alter this default action. These arguments have the form:

dev=xx(m,o)
  or
dev=xx(m)
where:
  dev = The desired system device (**root[dev]**, **pipe[dev]**,
    or **swap[dev]**)
  xx, m, o = same as for the boot device

If any combination of **root**, **pipe** or **swap** is specified, then those system devices will reside on that device, with the unspecified system devices using the defaults compiled in the kernel. Setting one device does not affect the default values for the other system devices.

Selecting The System Console

You can select the system console at boot time either by entering the command **systty=***x* at the boot prompt, or by placing the keywords **SYSTTY=***x* in the file **/etc/default/boot**. The letter *x* represents either a number or a string parameter.

If you use the **stty=***x* command at boot time, *boot* uses the string parameter *x* to pass the selected console device to the kernel. The values of the boot string parameter **systty** are:

sio Serial port COM1
scrn Display adapter

For example, to assign the system console to the serial port at COM1, enter this command at the boot prompt:

systty=sio

If you do not specifically set the system console at boot time, the *boot* program follows these steps to determine the system console:

- *boot* reads **/etc/default/boot** and looks for the keywords **SYSTTY=***x* where *x* is a number that specifies the system console device.

  0 indicates the serial adapter at COM1.
  1 indicates the display adapter.

- If **SYSTTY** is not found or **/etc/default/boot** is unreadable, *boot* checks for a display adapter and assigns it as the system console.

– If no display adapter is found, *boot* looks for COM1, sets the serial port to 9600 baud, 8 data bits, 1 stop bit, and no parity, and uses it as the system console.

Thus, to have *boot* automatically set the system console to the serial port at COM1, enter this line in **/etc/default/boot:**

STTY=0

Aliasing

A set of system devices can be aliased to a single keyword by defining the keyword in the file **/etc/default/boot.** This keyword can then be entered on the "Boot" command line and the boot program then reads the corresponding system devices from **/etc/default/boot** and pass them to the kernel. An alias has the following form:

key=file [root=xx(m) pipe=xx(m) swap=xx(m) prompt[="string"]]

In all cases, the device specification can also have the format dev=xx(m,o), where o is the offset.

For example, if you have a root file system on a second hard disk and want to use it, but want to boot using the **xenix** located on the first hard disk, enter the following line into the **/etc/default/boot** description:

disk2=hd(40,0)xenix root=hd(104,0) prompt="Using second disk"

The next time you boot the system from the first hard disk, enter "disk2" in response to the "Boot" prompt. **xenix** will be loaded from the first hard disk, and when you see the message, "Using second disk", press <RETURN> . **xenix** will now boot and use the root file system on the second hard disk. Note that you must edit the **/etc/default/boot** file in the root file system on the device from which *boot* will be read, in this case the first hard disk.

Boot options

Boot options can be changed via keywords in **/etc/default/boot.** The following keywords are recognized by *boot*:

LOADXENIX=YES	If YES, *boot* automatically loads XENIX after a moderate delay.

DEFBOOTSTR=*string*	*string* is used as the default boot string for timeouts and for no input on the command line. There can be no white space between DEFBOOTSTR, the "=" sign and *string*.
SYSTTY=*x*	If *x* is zero (0), the system console device is set to the serial adapter at COM1. If *x* is one (1), the system console is set to the main display adapter.
RONLYROOT=NO	Whether or not the root filesystem is to be mounted *readonly*. This should only be set to "yes" during installation.

Diagnostics

If an error occurs, *masterboot* displays an error message, and locks the system. The following is a list of the most common messages and their meanings:

IO ERR
An error occurred when *masterboot* tried to read in the partition boot of the active operating system.

BAD TBL
The bootable partition indicator of at least one of the operating systems in the fdisk table contains an unrecognizable code.

NO OS
There was an unrecoverable error that prevented the active operating system's partition boot from executing.

When *boot* displays error messages, it returns to the "Boot" prompt. The following is a list of the most common messages and their meanings:

bad magic number
The given file is not an executable program.

can't open *<pathname>*
The supplied pathname does not correspond to an existing file, or the device is unknown.

Stage 1 boot failure
The bootstrap loader cannot find or read the **boot** file. You must restart the computer and supply a file system disk with the **boot** file in the root directory.

not a directory
The specified area on the device does not contain a valid XENIX filesystem.

zero length directory
Although an otherwise valid filesystem was found, it contains a directory of apparently zero length. This most often occurs when a pre- System V XENIX filesystem (with incorrect, or incompatible word ordering) is in the specified area.

fload:read(*x*)=*y*
An attempted read of *x* bytes of the file returned only *y* bytes. This is probably due to a premature end-of-file. It could also be caused by a corrupted file, or incorrect word ordering in the header.

Files

/boot
/etc/default/boot
/etc/masterboot
/etc/hdboot0
/etc/hdboot1

See Also

autoboot(M), badtrk(M), fd(HW), fdisk(M), haltsys(C), hd(HW), init(M), sulogin(M)

Notes

The computer tries to boot off any diskette in the drive. If the diskette does not contain a valid bootstrap program, errors occur.

The *boot* program cannot be used to load programs that have not been linked for standalone execution. To create standalone programs, the **-A** option of the XENIX linker (*ld*(CP)) and special standalone libraries must be used.

Standalone programs can operate in real or protected mode, but they must not be large or huge models. Programs in real mode can use the input/output routines of the computer's startup ROM.

RONLYROOT should only be set to "yes" for installation. If it is set to "yes" during day-to-day operations, it will prevent you making changes to the root filesystem. You will then be required to boot from the floppy drive, edit the **/etc/default/boot file and reboot.**

Name

clockrate – Changes clock rate.

Syntax

/etc/clockrate frequency

Description

/etc/clockrate alters the interrupt timer clock frequency (different from the CPU clock frequency) to bring the system clock in sync with the computer's clock. This frequency is expressed in Megahertz (megaHZ) and can be found in the computer's hardware reference manual.

To set a new clockrate to 1.22878, for example, type:

```
# /etc/clockrate 1.22878
```

/etc/clockrate is a compiled "C" program, which modifies the kernel found in */xenix*.

/etc/clockrate only needs to be run once, unless you reinstall the XENIX distribution floppies.

Files

/etc/clockrate

Notes

Your computer may not be able to change the clockrate with this utility.

Name

cmos – Displays and sets the configuration data base.

Syntax

cmos [address [value]]

Description

The *cmos* command displays and/or sets the values in the CMOS configuration data base. This battery-powered data base stores configuration information about the computer that is used at power up to define the system hardware configuration and to direct boot procedures. The data base is 64 bytes long and is reserved for system operation. Refer to your computer hardware manual for more information.

The *cmos* command is typically used to alter the current hardware configuration when new devices are added to the system. When only *address* is given, the command displays the value at that address. If both *address* and a *value* are given, the command assigns the value to that address. If no arguments are given, the command displays the entire contents of the data base.

The CMOS configuration data base may also be examined and modified by reading from and writing to **/dev/cmos** file. Because successful system operation depends on correct configuration information, the data base should be modified by experienced system administrators only.

The computer manufacturer's diagnostic diskette should be run before setting the CMOS data base.

Files

/etc/cmos
/dev/cmos

Notes

Not all computers have a CMOS configuration data base. Some computers use switches on the main system board to configure the system. Refer to your computer hardware reference manual to determine whether you have a configuration data base.

Name

fd - floppy devices

Description

The **fd** devices implement the XENIX interface with floppy disk drives. Typically, the *tar*(C), *cpio*(C) or *dd*(C) commands are used to read or write floppy disks. For instance,

```
tar tvf /dev/fd0
```

tabulates the contents of the floppy disk in drive 0 (zero).

The block special **fd** devices are also block-buffered. The floppy driver can read or write 512 bytes at a time using raw i/o. Note that block transfers are always a multiple of the 1K disk block size.

The floppy devices are named **/dev/fd0** and **/dev/fd1** (see Notes, below, for more information about device naming procedure).

The corresponding character special (raw) devices, **/dev/rfd0** and **/dev/rfd1**, afford direct, unbuffered transmission between the floppy and the user's read or write transfer address in the user's program.

For information about formatting, see *format*(C).

The minor device number determines what kind of physical device is attached to each device file (see Notes).

Files

/dev/fd0	/dev/rfd048ds8	/dev/rfd096ds15
/dev/fd1	/dev/rfd148ds8	/dev/rfd196ds15
/dev/rfd0	/dev/rfd048ds9	/dev/rfd096ds8
/dev/rfd1	/dev/rfd148ds9	/dev/rfd196ds8
		/dev/rfd048ss8
		/dev/rfd148ss9

Notes

When accessing the character special floppy devices, the user's buffer must begin on a word boundary. The count in a *read*(S), *write*(S), or *lseek*(S) call to a character special floppy device must be a multiple of 512 bytes.

Device names determine the particular drive and media configuration. The device names have the form: fd048ds9 Where: fd0 = drive number (0, 1, 2 or 3) 48 = number of disk tracks per inch (48 or 96) ds = single or double sided floppy (ss or ds) 9 = number of sectors on the floppy (8 or 9)

For instance, /dev/fd048ss9 indicates a 48 track per inch, single sided, 9 sector floppy disk device in drive 0.

The minor device numbers for floppy drives depend on the drive and media configuration. The most common are:

	48tpi				96tpi	
	ds/8	ds/9	ss/8	ss/9	ds/15	ds/8
Drive	Minor Device Number					
0	12	4	8	0	52	44
1	13	5	9	1	53	45
2	14	6	10	2	54	46
3*						

* reserved for special, non-floppy devices connected to the floppy controller as unit #3.

The scheme for creating minor device numbers is as follows. When interpreted as a binary number, each bit of the minor device number represents some aspect of the device/media configuration.

For example, the minor device number for **/dev/fd048ss8** is "8." Interpreted as a binary number, 8 is:

00001000

This is how each bit, or binary digit, is significant:

48tpi - 0 96tpi - 1	Sectors per Track		ss - 0 ds - 1	Drive	
32	16	8	4	2	1
0	0	1	0	0	0

Only the last six digits of the number are used in minor device identification. The first significant digit is the third from the left. In this example, the third digit from the left is zero, thus the device is 48tpi. The next two digits mean:

Bits		Sectors per Track
16	8	
0	0	9
0	1	8
1	0	15

The fourth digit tells whether the floppy is single sided (ss - 0) or double sided (ds - 1). The last two signify the drive number:

Bits		Drive Number
2	1	
0	0	0
0	1	1
1	0	2
1	1	3*

* reserved for special, non-floppy devices connected to the floppy controller as unit #3.

Using this information, you can construct any minor device numbers you need.

It is not advisable to format a low density (48tpi) diskette on a high density (96tpi) floppy drive. Low density diskettes written on a high density drive should be read on high density drives. They may or may not be readable on a low density drive.

Use error-free floppy disks for best results on reading and writing.

Name

hd – Internal hard disk drive

Description

The files **hd00, hd01** through **hd04, hd0a** through **hd0d, root**, and **swap** provide block-buffered access to the primary hard disk. The corresponding files for a second hard disk are listed below.

root refers to the root file system; **swap** refers to the swap area; **hd00** is the entire disk; **hd01** through **hd04** are the four partitions recognized by ROM-BIOS. Partition one is generally the XENIX partition. These special device files access the disks via the system's normal buffering mechanism and may be read and written without regard to the size of physical disk records.

The following are the names of the fixed disk partitions. Each partition can be accessed through a block interface, for example **/dev/hd01**, or through a character (raw) interface, for example **/dev/rhd01.**

Device File Names for Fixed Disks		
Disk 1	Disk 2	Partition
/dev/hd00 /dev/rhd00	/dev/hd10 /dev/rhd10	entire disk
/dev/hd01 /dev/rhd01	/dev/hd11 /dev/rhd11	first partition
/dev/hd02 /dev/rhd02	/dev/hd12 /dev/rhd12	second partition
/dev/hd03 /dev/rhd03	/dev/hd13 /dev/rhd13	third partition
/dev/hd04 /dev/rhd04	/dev/hd14 /dev/rhd14	fourth partition
/dev/hd0a /dev/rhd0a	/dev/hd1a /dev/rhd1a	active partition
/dev/hd0d /dev/rhd0d	/dev/hd1d /dev/rhd1d	DOS partition
/dev/u /dev/ru		
/dev/root /dev/rroot		root file system
/dev/swap /dev/rswap		swap area

Note that the last three file names do not exist for a second disk.

The device file names for DOS partitions function similarly to **/dev/hd?a** .

To access DOS partitions, specify letters such as "C:" or "D:" to indicate first or second partitions. The file **/etc/default/msdos** contains lines that assign a letter abbreviation for the DOS device name. Refer to *dos*(C).

The following table lists the minor device numbers for possible disk partitions. The minor device names for the raw devices are identical.

Minor Device Numbers			
Partition:	Minor Device Number:	Partition:	Minor Device Number:
hd00	0	hd10	64
hd01	15	hd11	79
hd02	23	hd12	87
hd03	31	hd13	95
hd04	39	hd14	103
hd0a	47	hd1a	111
hd0d	55	hd1d	119
root	40	u (user choice)	104
swap	41	u1	105
u (on 1st disk)	42	u2	106
recover	46		

Files

/dev/hd0a
/dev/rhd0a
/dev/hd0?
/dev/rhd0?
/dev/hd0d
/dev/rhd0d

/dev/hd1a
/dev/rhd1a
/dev/hd1?
/dev/rhd1?
/dev/hd1d
/dev/rhd1d

/dev/u
/dev/ru
/dev/root
/dev/rroot
/dev/swap
/dev/rswap

See Also

badtrk(C), divvy(C), dos(C), mkdev(C)

Diagnostics

The following messages may be printed on the console:

invalid fixed disk parameter table

and:

Error on Fixed Disk (minor *n*), blkno = *nnnnn*,
cmd=*nnnnn*, status=*nnnn*,
sector = *nnnnn*, Cylinder/head = *nnnnn*

Possible reasons for the first error include:

- The kernel is unable to get drive specifications, such as number of heads, cylinders, and sectors per track, from the disk controller ROM.
- Improper configuration.
- The disk is not turned on.
- The disk is not supported.

The second error specifies the following information:

- *blkno :* The XENIX block number within the device.
- *cmd :* The last command sent to the disk controller.
- *status :* The first byte of error status from the disk controller.
- *sector* and *Cylinder/head* specify the location of a possible flaw. This information is used with *badtrk*(M).

Notes

On the first disk, **hd00** denotes the entire disk and is used to access the master boot record and partition table. For the second disk, **hd10** denotes the entire disk and is used to access its partition table. Do not write to **hd10** and **hd00**.

Name

keyboard – The PC keyboard.

Description

The PC keyboard is used to enter data, switch screens, and send certain control signals to the computer. XENIX performs terminal emulation on the PC screen and keyboard, and, in doing so, makes use of several particular keys and key combinations. These keys and key combinations have special names that are unique to the XENIX system, and may or may not correspond to the keytop labels on your keyboard. These keys are described later.

When you press a key, one of the following happens:

- An ASCII value is entered
- A string is sent to the computer.
- A function is initiated.
- The meaning of another key, or keys, is changed.

When a key is pressed (a keystroke), the keyboard sends a scan-code to the computer, it is interpreted by the keyboard driver. The interpretation of key codes may be modified so that keys can function differently from their default actions.

There are three special occurrences, or keystrokes:

- Switch screens.
- Send signals.
- Change the value of previous character, characters or string.

Switching Screens (Multiscreen)

To get to the next consecutive screen, enter **Ctrl–PrtSc** using the **Ctrl** key, and the **PrtSc** key. Any active screen may be selected by entering **alt–F***n*, where **Fn** is one of the function keys. **F1** refers to the PC display (**/dev/tty01**).

Signals

A signal affects some process or processes. Examples of signals are **Ctrl−d** (end of input, exits from shell), **Ctrl−** (quits a process), **Ctrl−s** (stop output to the screen), and **Ctrl−q** (resume sending output).

Typically, characters are mapped to signals using *stty*(C). The only way to map signals is using *stty*.

Altering Values

The actual code sent to the keyboard driver can be changed by using certain keys in combination. For example, the SHIFT key changes the ASCII values of the alphanumeric keys. Holding down the **Ctrl** key while pressing another key sends a control code (**Ctrl−d**, **Ctrl−s**, **Ctrl−q**, etc.).

Special Keys

To help you find the special keys, the following table shows which keys on a typical console correspond to XENIX system keys. In this table, a hyphen (-) between keys means 'hold down the first key while pressing the second.'

XENIX Name	**Keytop**	**Action**
INTR	Del	Stops current action and returns to the shell. This key is also called the RUB OUT or INTERRUPT key.
BACKSPACE	←	Deletes the first character to the left of the cursor. Note that the "cursor left" key also has a left arrow (←) on its keytop, but you cannot backspace using that key.
Ctrl-d	Ctrl-d	Signals the end of input from the keyboard; also exits current shell.
Ctrl-h	Ctrl-h	Deletes the first character to the left of the cursor. Also called the ERASE key.
Ctrl-q	Ctrl-q	Restarts printing after it has been stopped with Ctrl-s.

Ctrl-s	Ctrl-s	Suspends printing on the screen (does not stop the program).
Ctrl-u	Ctrl-u	Deletes all characters on the current line. Also called the KILL key.
Ctrl-\	Ctrl-\	Quits current command and creates a *core* file, if allowed. (Recommended for debugging only.)
ESCAPE	Esc	Special code for some programs. For example, changes from insert mode to command mode in the *vi*(C) text editor.
RETURN	(down-left arrow or ENTER)	Terminates a command line and initiates an action from the shell.
F*n*	F*n*	Function key *n*. F1-F12 are unshifted, F13-F14 are shifted F1-F12, F25-F36 are Ctrl-F1 through F12, and F37-F48 are Ctrl-Shift-F1 through F12. The next F*n* keys (F49-F60) are on the number pad (unshifted): F49 - '7' F55 - '6' F50 - '8' F56 - '+' F51 - '9' F57 - '1' F52 - '-' F58 - '2' F53 - '4' F59 - '3' F54 - '5' F60 - '0' For keys F61 through F96, see **/usr/lib/keyboard/strings**. These function keys are not available on all keyboards, but you can map other keys to represent them.

The keyboard mapping is performed through a structure defined in **/usr/include/sys/keyboard.h**. Each key can have ten states. The first eight are:

– Base	– Ctrl-Shift
– Shift	– Alt-Shift
– Ctrl	– Alt-Ctrl
– Alt	– Alt-Ctrl-Shift

There are two additional states indicated by two special bytes. The first is a "special state" byte whose bits indicate whether the key is "special" in one or more of the first eight states.

The second is one of four characters (C, N, B, O) which indicate how the lock keys affect the particular key. This is discussed further in the next section, "Scan Codes."

Scan Codes

The following table describes the default contents of **/usr/lib/keyboard/keys**. The column headings are:

SCAN CODE – The scan code generated by the keyboard hardware when a key is pressed. There is no user access to the scan code generated by releasing a key.

BASE – The normal value of a key press.

SHIFT – The value of a key press when the SHIFT is also being held down.

LOCK – Indicates which lock keys affect that particular key:

- C indicates Capslock
- N indicates Numlock
- B indicates both
- O indicates locking is off

Keys affected by the lock keys C, B, or N, send the shifted value (scan code) of current state when that lock key is on. When the shift key is depressed while a lock key is also on, the key reverts (toggles) to its original state.

The other columns are the values of key presses when combinations of the CTRL, ALT and SHIFT keys are also held down.

All values, except for keywords, are ASCII character values. The keywords refer to the special function keys.

SCAN CODE	BASE	SHIFT	CTRL	CTRL SHIFT	ALT	ALT SHIFT	ALT CTRL	ALT CTRL SHIFT	LOCK
0	nop	nop	nop	nop	nop	nop	nop	nop	O
1	esc	esc	nop	nop	esc	esc	nop	nop	O
2	'1'	'!'	nop	nop	'1'	'!'	nop	nop	O
3	'2'	'@'	nop	nop	'2'	'@'	nop	nop	O
4	'3'	'#'	nop	nop	'3'	'#'	nop	nop	O
5	'4'	'$'	nop	nop	'4'	'$'	nop	nop	O
6	'5'	'%'	nop	nop	'5'	'%'	nop	nop	O
7	'6'	'^'	rs	rs	'6'	'^'	rs	rs	O
8	'7'	'&'	nop	nop	'7'	'&'	nop	nop	O
9	'8'	'*'	nop	nop	'8'	'*'	nop	nop	O
10	'9'	'('	nop	nop	'9'	'('	nop	nop	O
11	'0'	')'	nop	nop	'0'	')'	nop	nop	O
12	'-'	'_'	ns	ns	'-'	'_'	ns	ns	O
13	'='	'+'	nop	nop	'='	'+'	nop	nop	O
14	bs	bs	del	del	bs	bs	del	del	O
15	ht	btab	nop	nop	ht	btab	nop	nop	O
16	'q'	'Q'	dc1	dc1	'q'	'Q'	dc1	dc1	C
17	'w'	'W'	etb	etb	'w'	'W'	etb	etb	C
18	'e'	'E'	enq	enq	'e'	'E'	enq	enq	C
19	'r'	'R'	dc2	dc2	'r'	'R'	dc2	dc2	C
20	't'	'T'	dc4	dc4	't'	'T'	dc4	dc4	C
21	'y'	'Y'	em	em	'y'	'Y'	em	em	C
22	'u'	'U'	nak	nak	'u'	'U'	nak	nak	C
23	'i'	'I'	ht	ht	'i'	'I'	ht	ht	C
24	'o'	'O'	si	si	'o'	'O'	si	si	C
25	'p'	'P'	dle	dle	'p'	'P'	dle	dle	C
26	'['	'{'	esc	esc	'['	'{'	esc	esc	O
27	']'	'}'	gs	gs	']'	'}'	gs	gs	O
28	cr	cr	nl	nl	cr	cr	nl	nl	O
29	ctrl	ctrl	ctrl	ctrl	ctrl	ctrl	ctrl	ctrl	O
30	'a'	'A'	soh	soh	'a'	'A'	soh	soh	C
31	's'	'S'	dc3	dc3	's'	'S'	dc3	dc3	C
32	'd'	'D'	eot	eot	'd'	'D'	eot	eot	C
33	'f'	'F'	ack	ack	'f'	'F'	ack	ack	C
34	'g'	'G'	bel	bel	'g'	'G'	bel	bel	C
35	'h'	'H'	bs	bs	'h'	'H'	bs	bs	C
36	'j'	'J'	nl	nl	'j'	'J'	nl	nl	C
37	'k'	'K'	vt	vt	'k'	'K'	vt	vt	C
38	'l'	'L'	np	np	'l'	'L'	np	np	C
39	';'	':'	nop	nop	';'	':'	nop	nop	O
40	'\''	'"'	nop	nop	'\''	'"'	nop	nop	O
41	'`'	'~'	nop	nop	'`'	'~'	nop	nop	O
42	lshift	lshift	lshift	lshift	lshift	lshift	lshift	lshift	O
43	'\\'	'|'	fs	fs	'\\'	'|'	fs	fs	O
44	'z'	'Z'	sub	sub	'z'	'Z'	sub	sub	C
45	'x'	'X'	can	can	'x'	'X'	can	can	C
46	'c'	'C'	etx	etx	'c'	'C'	etx	etx	C
47	'v'	'V'	syn	syn	'v'	'V'	syn	syn	C

48	'b'	'B'	stx	stx	'b'	'B'	stx	stx	C
49	'n'	'N'	so	so	'n'	'N'	so	so	C
50	'm'	'M'	cr	cr	'm'	'M'	cr	cr	C
51	','	'<'	nop	nop	','	'<'	nop	nop	O
52	'.'	'>'	nop	nop	'.'	'>'	nop	nop	O
53	'/'	'?'	nop	nop	'/'	'?'	nop	nop	O
54	rshift	rshift	rshift	rshift	rshift	rshift	rshift	rshift	O
55	'*'	'*'	nscr	nscr	'*'	'*'	nscr	nscr	O
56	alt	alt	alt	alt	alt	alt	alt	alt	O
57	' '	' '	' '	' '	' '	' '	' '	' '	O
58	clock	clock	clock	clock	clock	clock	clock	clock	O
59	fkey1	fkey13	fkey25	fkey37	scr1	scr11	scr1	scr11	O
60	fkey2	fkey14	fkey26	fkey38	scr2	scr12	scr2	scr12	O
61	fkey3	fkey15	fkey27	fkey39	scr3	scr13	scr3	scr13	O
62	fkey4	fkey16	fkey28	fkey40	scr4	scr14	scr4	scr14	O
63	fkey5	fkey17	fkey29	fkey41	scr5	scr15	scr5	scr15	O
64	fkey6	fkey18	fkey30	fkey42	scr6	scr16	scr6	scr16	O
65	fkey7	fkey19	fkey31	fkey43	scr7	scr7	scr7	scr7	O
66	fkey8	fkey20	fkey32	fkey44	scr8	scr8	scr8	scr8	O
67	fkey9	fkey21	fkey33	fkey45	scr9	scr9	scr9	scr9	O
68	fkey10	fkey22	fkey34	fkey46	scr10	scr10	scr10	scr10	O
69	nlock	nlock	dc3	dc3	nlock	nlock	dc3	dc3	O
70	slock	slock	del	del	slock	slock	del	del	O
71	fkey49	'7'	'7'	'7'	'7'	'7'	'7'	'7'	N
72	fkey50	'8'	'8'	'8'	'8'	'8'	'8'	'8'	N
73	fkey51	'9'	'9'	'9'	'9'	'9'	'9'	'9'	N
74	fkey52	'-'	'-'	'-'	'-'	'-'	'-'	'-'	N
75	fkey53	'4'	'4'	'4'	'4'	'4'	'4'	'4'	N
76	fkey54	'5'	'5'	'5'	'5'	'5'	'5'	'5'	N
77	fkey55	'6'	'6'	'6'	'6'	'6'	'6'	'6'	N
78	fkey56	'+'	'+'	'+'	'+'	'+'	'+'	'+'	N
79	fkey57	'1'	'1'	'1'	'1'	'1'	'1'	'1'	N
80	fkey58	'2'	'2'	'2'	'2'	'2'	'2'	'2'	N
81	fkey59	'3'	'3'	'3'	'3'	'3'	'3'	'3'	N
82	fkey60	'0'	'0'	'0'	'0'	'0'	'0'	'0'	N
83	del	'.'	del	del	del	del	del	del	N
84	nop	nop	nop	nop	nop	nop	nop	nop	O
85	fkey11	fkey23	fkey35	fkey47	scr11	scr11	scr11	scr11	O
86	fkey12	fkey24	fkey36	fkey48	scr12	scr12	scr12	scr12	O

The next table lists the "value" of each of the special keywords used in **/usr/lib/keyboard/keys** (and the preceding table). *mapkey*(M) places a "value" in the *ioctl* buffer during key mapping. The keywords are only used in the scan code file (**/usr/lib/keyboard/keys**) for readability.

Name	Value	Meaning
nop	0	No operation - no action from keypress
lshift	2	Left hand shift
rshift	3	Right hand shift
clock	4	Caps lock
nlock	5	Numeric lock
slock	6	Scroll lock
alt	7	Alt key
btab	8	Back tab key - generates fixed sequence (esc [Z)
ctrl	9	Control key
nscr	10	Switch to the next screen
scr1	11	Switch to screen #1
...		...
scr16	26	Switch to screen #16
fkey1	27	Function key #1
...		...
fkey96	122	Function key #96

This table lists names and decimal values that are interchangeable in the *mapkey* file. Names are used in place of numeric constants to make it easier to read the scan code table. Again, only the decimal values are placed in the *ioctl* buffer. These are taken from **ascii**(M).

Name	Value	Name	Value
nul	0	dc1	17
soh	1	dc2	18
stx	2	dc3	19
etx	3	dc4	20
eot	4	nak	21
enq	5	syn	22
ack	6	etb	23
bel	7	can	24
bs	8	em	25
ht	9	sub	26
nl	10	esc	27
vt	11	fs	28
np	12	gs	29
cr	13	rs	30
so	14	ns	31
si	15	del	127
dle	16		

Keyboard Mapping

The PC keyboard is mapped as part of terminal emulation. This kind of mapping is performed only on the computer keyboard, not on remote terminals. Use *mapkey* to change keyboard mapping. To change the mapping for individual channels (multiscreens), use *mapchan*(M).

Keyboard mapping can also be performed using *ioctl*. The syntax is the same as for string key mapping (see previous section).

For keyboard mapping, *cmd* is GIO_KEYMAP to display the current map, and PIO_KEYMAP puts the prepared buffer into place.

String Key Mapping

To map string (function) keys, use the *mapstr* (see *mapkey*(M)) utility. *mapstr* modifies the string mapping table where function keys are defined.

The string mapping table is an array of 512 bytes (typedef *strmap_t*) containing null terminated strings that redefine the function keys. The first null terminated string is assigned to the first string key, the second string to the second string key, and so on.

There is no limit to the length of any particular string as long as the whole table does not exceed 512 bytes, including nulls. Strings are made null by the introduction of extra null characters.

The following is a list of default function key values:

Default Function Key Values

Key #	Function	Shift Function	Ctrl Function	Ctrl Shift Function
1	ESC [M	ESC [Y	ESC [k	ESC [w
2	ESC [N	ESC [Z	ESC [l	ESC [x
3	ESC [O	ESC [a	ESC [m	ESC [y
4	ESC [P	ESC [b	ESC [n	ESC [z
5	ESC [Q	ESC [c	ESC [o	ESC [@
6	ESC [R	ESC [d	ESC [p	ESC [[
7	ESC [S	ESC [e	ESC [q	ESC [\
8	ESC [T	ESC [f	ESC [r	ESC []
9	ESC [U	ESC [g	ESC [s	ESC [^
10	ESC [V	ESC [h	ESC [t	ESC [_
11	ESC [W	ESC [i	ESC [u	ESC [`
12	ESC [X	ESC [j	ESC [v	ESC [{

Home	ESC [H	End	ESC [F
Up arrow	ESC [A	Down arrow	ESC [B
Page up	ESC [I	Page down	ESC [G
Left arrow	ESC [D	5	ESC [E
Right arrow	ESC [C	Insert	ESC [L

You can also map string keys using *ioctl*(S). The syntax is:

```
#include <sys/keyboard.h>
ioctl(fd,cmd,buf)
int fd, cmd;
char *buf;
...
```

For string key mapping where *cmd* is GIO_STRMAP to display the string mapping table and PIO_STRMAP to put the new string mapping table in place.

Files

/usr/lib/keyboard/keys
/usr/lib/keyboard/strings

See Also

mapchan(F), mapchan(M), mapkey(M), multiscreen(M), screen(HW), setkey(C), stty(C)

Name

lp, lp0, lp1, lp2 – Line printer device interfaces.

Description

The **lp0** , **lp1**, and **lp2** files provide access to the optional parallel ports of the computer. The **lp0** and **lp2** files provide access to parallel ports 1 and 2, respectively. The **lp1** file provides access to the parallel port on the monochrome adaptor.

Only one of **lp0** and **lp1** may be used on a given system. To access two parallel printers on a system, use either **lp0** or **lp1**, and **lp2**.

Files

/dev/lp0
/dev/lp1
/dev/lp2

See Also

lp(C), lpadmin(C), lpsched(C), lpinit(C)

Notes

The standard **lp** ports, **lp0**, **lp1**, and **lp2** send a printer initialization string the first time the file is opened after the system is *booted*.

Not all computers have an alternate parallel port slot.

Name

Machine – Description of host machine.

Description

This page lists the internal characteristics of personal computers which use the Intel 8086 processor family and its associated hardware. The information is intended for software developers who wish to transfer relocatable object or executable files from other XENIX machines to a personal computer then prepare the files for execution on the personal computer.

Central Processing Unit	Intel 8086, 8088, 80186, 80286, 80386
Disk Block Size (BSIZE)	1024 bytes
Memory Management Scheme	Unmapped (8086, 8088, 80186) Segmented (80286) Segmented and paged (80386)
Split Instruction and Data	Supported
Variable Stack Size	Supported (8086, 80386 only) (8086, 80386 default configuration)
Fixed Stack Size	Supported (80286 default configuration)
Clock Ticks	.05 second (8086, 8088, 80186) .02 second (80286, 80386)

Binary Compatibility

The small and middle model binary programs created by the C compiler *cc*(CP) run on many processors. The following chart shows which XENIX systems running on which processors produce code executable on other machines. It is assumed that system specific system calls are not used to produce portable code. *cc*(CP) produces code by default, but can also be used as a cross development compiler, by using the appropriate flags.

SCO-*nn* is XENIX distributed by The Santa Cruz Operation, Inc. MS-*nn* is XENIX distributed by Microsoft Corporation. Intel XENIX is distributed by Intel Corporation. Altos XENIX is distributed by Altos Computer Systems. *nn* designates the machine processor. System designates the version of XENIX, either 2.3, 3.0, or System V.

Binary Compatibility			
Your System Processor	Default compiler produces programs which run on System/Processor	Runs default programs created on System/Processor	Compiles (cross development) programs for System/Processor
SCO-86 3.0	SCO-86 [3.0, Sys V] SCO-186 [3.0, Sys V] SCO-286 SysV	SCO-86 3.0 SCO-186 3.0 Intel, Altos-86 2.3, 3.0	DOS*
SCO-86 SystemV	SCO-86 Sys V SCO-186 Sys V SCO-286 Sys V MS-286 Sys V	SCO-86 [3.0, Sys V] SCO-186 [3.0, Sys V] SCO-286 Sys V Intel, Altos-86 2.3, 3.0	MS-286 3.0† DOS*
SCO-186 3.0	SCO-86 [3.0, Sys V] SCO-186 [3.0, Sys V] SCO-286 Sys V	SCO-86 3.0 SCO-186 3.0 Intel, Altos-86 2.3, 3.0	DOS*
SCO-186 System V	SCO-86 Sys V SCO-186 Sys V SCO-286 Sys V MS-286 Sys V	SCO-86 [3.0, Sys V] SCO-186 [3.0, Sys V] SCO-286 Sys V Intel, Altos-86 2.3, 3.0	MS-286 3.0† DOS*
SCO-286 3.0	SCO-286 [3.0, Sys V] MS-286 [3.0†, Sys V]	SCO-286 3.0 MS-286 3.0†	DOS*
SCO-286 System V	SCO-86 Sys V SCO-186 Sys V SCO-286 Sys V MS-286 Sys V	SCO-86 [3.0, Sys V] SCO-186 [3.0, Sys V] SCO-286 [3.0, Sys V] MS-286 [3.0†, Sys V]	SCO-286 3.0 MS-286 3.0† DOS*
SCO-386 System V	SCO-86 Sys V SCO-186 Sys V SCO-286 Sys V SCO-386 Sys V MS-286 Sys V MS-386 Sys V	SCO-86 [3.0, Sys V] SCO-186 [3.0, Sys V] SCO-286 [3.0, Sys V] SCO-386 [Sys V] MS-286 [3.0†, Sys V] MS-386 [Sys V]	SCO-286 3.0 MS-286 3.0† DOS*
MS-286 3.0†	MS-286 [3.0†, Sys V] SCO-286 Sys V	SCO-286 3.0	DOS*
MS-286 System V	MS-286 Sys V SCO-286 Sys V	SCO-86 [3.0, Sys V]‡ SCO-186 [3.0, Sys V]‡ SCO-286 [3.0, Sys V]‡	DOS*
MS-386 System V	MS-386 Sys V SCO-386 Sys V	SCO-86 [3.0, Sys V]‡ SCO-186 [3.0, Sys V]‡ SCO-286 [3.0, Sys V]‡ SCO-386 [Sys V]‡	DOS*

* MS-DOS for i8086/8088, i80186 and i80286 processors.
† MS-286 3.0 XENIX is equivalent to Intel 286 3.0 XENIX.
‡ untested, pending release of this product.

See also

clockrate(HW), cc(CP), ld(CP), a.out(F).

Name

parallel - Parallel interface devices.

Description

There are several parallel devices:

/dev/lp0 Main parallel adapter.

/dev/lp1 Adapter on monochrome video card.

/dev/lp2 Alternate parallel adapter (on appropriate machines).

It is not possible to have all three parallel devices on one system. XT computers only allow the use of **/dev/lp0**. Some AT computers allow the use of two parallel devices, **/dev/lp0**, and either **/dev/lp1**, or **/dev/lp2**.

If a parallel device fails to interrupt properly, the parallel driver enters "poll mode." Once interrupts are received from the device, the driver returns to its original mode.

The parallel driver delays a certain amount of time when a parallel device is closed. The amount of delay can affect printer performance, but is necessary to compensate for different sizes of printer buffers and printer speeds. For example, this command sets the delay on close to 1 second, specified in 10ths of a second by Ctrl-J (ASCII value 10):

```
stty eol "^J" < /dev/lp0
```

The default delay value is 2 seconds (^T).

Notes

Parallel adapters on add-on cards will function, but switches must be set correctly. Some compatable computers have ports lp0 and lp1 reversed.

The *stty*(C) command for output processing is supported on a parallel device. *stty* options that have no effect on a parallel device are ignored and no error messages are displayed.

Usage

Usually invoked by through *lp*(C), but can be written to directly.

Files

/dev/lp0
/dev/lp1
/dev/lp2

See Also

lp(C), lp(HW), lpadmin(C), lpinit(C), lpsched(C), serial(HW)

Name

ramdisk - Memory block device

Description

The *ramdisk* device driver provides a block interface to memory. A *ramdisk* can be used like any other block device, including making it into a file systems using *mkfs*(C). There are eight *ramdisk*s available.

The characteristics of a *ramdisk* file are determined by its minor device number. The bits in the minor device number encode its size, longevity, and which of the eight possible *ramdisk*s it is.

The three low-order bits of the minor device number determine which of the eight *ramdisk*s is being accessed.

The next four bits of the minor device number determine the size of the *ramdisk*. The size of a *ramdisk* must be a power of 2, and must be at least 16K. Since 4 bits are available, there are 16 possible sizes, starting at 16K and doubling every time the size indicator is incremented, to produce possible sizes of 16K, 32K, 64K, and up.

The high-order bit is a longevity indicator. If set, memory is permanently allocated to that *ramdisk*, and can be deallocated only by rebooting the system. Permanent *ramdisk*s can only be allocated by the superuser. However, once a permanent *ramdisk* is allocated (by opening it), it can be read and written by anyone having the appropriate permissions on the *ramdisk* inode.

If clear, the *ramdisk* is deallocated when no processes have it open. To create an easily removable, but semi-permanent *ramdisk*, use a separate process to keep the device open for as long as necessary.

Since a complete set of *ramdisks* (8) would consume 256 inodes, only one example 16K *ramdisk* (**/dev/ram00**) is created when the system is installed. The system administrator can use this example to determine the major device numbers for any other required *ramdisks*. The following table shows how the minor device number is constructed:

Example Minor Device Number Construction									
Description	Longe-vity	Size (see next table)				Ram Disk No.			Minor Device Number
16K (#1) (Temporary)	0	0	0	0	0	0	0	1	1
16K (#1) (Permanent)	1	0	0	0	0	0	0	1	129
64K (#0) (Temporary)	0	0	0	1	0	0	0	0	16
512K (#7) (Permanent)	1	0	1	0	1	1	1	1	175

The contents of the size field and the corresponding *ramdisk* size is shown in the next table.

Size Bits				Ramdisk Size
0	0	0	0	16K
0	0	0	1	32K
0	0	1	0	64K
0	0	1	1	128K
0	1	0	0	256K
0	1	0	1	512K
0	1	1	0	1M
0	1	1	1	2M
1	0	0	0	4M
1	0	0	1	8M
1	0	1	0	16M
1	0	1	1	32M
1	1	0	0	64M
1	1	0	1	128M
1	1	1	0	256M
1	1	1	1	512M

Once you determine the major and minor device numbers of a new *ramdisk*, make the appropriate device node using *mknod*(C).

To make a file system on a non-permanent *ramdisk*, the device file must be held open between the *mkfs* and the *mount*(C) operations. Otherwise, the *ramdisk* is allocated at the start of the *mkfs* command, and deallocated at its end. Once the *ramdisk* is mounted, it is open until it is unmounted.

The following shell fragment shows one way to use *mkfs* on a non-permanent 512K *ramdisk*, then mount it:

```
( /etc/mkfs /dev/ram40 512
  /etc/mount /dev/ram40 /dev/mnt
) > /dev/ram40
```

Notes

*ramdisk*s must occupy contiguous memory. If free memory is fragmented, opening a *ramdisk* may fail even though there is enough total memory available. Ideally, all *ramdisk*s should be allocated at system startup. This helps prevent the *ramdisk*s themselves from fragmenting memory.

ramdisks are geared towards use in specialized applications. In many cases, you will notice a *decrease in system performance* when *ramdisks* are used, because XENIX can generally put the memory to better use elsewhere.

Files

/dev/ram00

See Also

mkfs(C), mount(C), mknod(C)

Name

screen – tty[01–*n*], color, monochrome, ega, pga
– Display adapter and video monitor.

Description

The **tty[01–*n*]** device files provide character I/O between the system and the video display monitor and keyboard. Each file corresponds to a separate teletype device. Although there is a maximum of 10 screens, the exact number available (*n*) depends upon the amount of memory in the computer.

Each device consists of two files: a readable file from the keyboard and a writable file to one of five display devices: **color, monochrome, ega,** and **pga.** The **color** file provides access to a color/graphics adapter; the **monochrome** file to the monochrome adapter; the **ega** file to the enhanced graphics adapter; the **pga** file to the professional graphics adapter. The screens are modeled after a 24 line, 80 column ASCII terminal, unless specified otherwise.

tty01 is the default output device for keyboard input. If the **/dev/console** is the default output device for system error messages, and the display being used is switched to graphics mode, console messages are not displayed. They are, instead, redirected to the kernel error handler where they can then be retrieved by the application which took over the screen.

Although all **tty**[01-*n*] devices may be open concurrently, only one of the corresponding devices can be active at any given time. The active device displays its own screen and takes sole possession of the keyboard. It is an error to attempt to access the **color, monochrome, ega** or **pga** file when no corresponding adapter exists.

To get to the next consecutive screen, enter **Ctrl–PrtSc** using the **Ctrl** key, and the **PrtSc** key. Any active screen may be selected by entering **alt–Fn**, where **Fn** is one of the function keys. **F1** refers to the **tty01** device.

Control Modes

To change the display mode used for the video monitor, open the device file associated with that mode. For example, to switch to display output on an installed CGA, a program should open **/dev/color** and use the selector mapping *ioctl s* on the file descriptor that is returned from the *open* call. Using device files helps ensure future compatibility.

Programs can also use the *stty*(C) or *stty*(HW) commands, or an *ioctl*(S) with the following requests:

SWAPMONO	Selects the monochrome display as the output device for the video monitor.
SWAPCGA	Selects the regular color display as the output device for the video monitor.
SWAPPGA	Selects the professional color display as the output device for the video monitor.
SWAPEGA	Selects the enhanced color display as the output device for the video monitor.

To obtain information about the display adapter type currently attached to the video monitor, you can use *ioctl*(S) with the following request:

CONS_CURRENT	Returns the display adapter type currently attached to the video monitor. The return value can be one of: MONO, CGA, EGA, or PGA.

Display Modes

To switch display modes on the various display devices, you can use the *stty*(C) or *stty*(HW) commands, or you can use *ioctl*(S) with the following requests:

SW_B80x25	Selects 80x25 black and white text display mode.
SW_C80x25	Selects 80x25 color text display mode.

SW_BG320	Selects 320x200 black and white graphics display mode.
SW_CG320	Selects 320x200 color graphics display mode.
SW_BG640	Selects 640x200 black and white graphics display mode.
SW_EGAMONO80x25	Selects EGA (Enhanced Graphics Adapter) mode 7 - emulates support provided by the monochrome display.
SW_EGAMONOAPA	Selects EGA support for 640x350 graphics display mode (EGA mode F).
SW_ENHMONOAPA2	Selects EGA mode F*.
SW_ENHB40x25	Selects enhanced EGA support for 40x25 black and white text display mode.
SW_ENHC40x25	Selects enhanced EGA support for the 40x25 color text display mode.
SW_ENHB80x25	Selects enhanced EGA support for 80x25 black and white text display mode.
SW_ENHC80x25	Selects enhanced EGA support for 80x25 color text display mode.
SW_CG320_D	Selects EGA support for 320x200 graphics display mode. (EGA mode D.)
SW_CG640_E	Selects EGA support for 640x200 graphics display mode (EGA mode E).
SW_CG640x350	Selects EGA support for 640x350 graphics display mode (EGA mode 10).
SW_ENH_CG640	Selects EGA mode 10*.
SW_MCAMODE	Reinitializes the monochrome adapter.

Switching to an invalid display mode for a display device will result in an error.

Getting Display Modes

To obtain information about the current display modes, you can use *ioctl*(S) with the requests that follow. These apply to the color graphics adapter, the enhanced graphics adapter or the professional graphics adapter:

CGA_GET	Returns the current display mode setting of the color graphics adapter.
PGA_GET	Returns the current display mode setting of the professional graphics adapter.
EGA_GET	Returns the current display mode setting of the enhanced graphics adapter.
MCA_GET	Returns the current display mode setting of the monochrome adapter.
CONS_GET	Returns the current display mode setting for whatever display adapter is being used.

Memory Mapping Modes

The *ioctl*(S) routine is used to map the display memory of the various devices into the user's data space. It returns a selector for the display memory. The macro **sotofar** is used to create a far pointer from this selector so the display memory can be accessed. The **sotofar** macro is located in */usr/include/sys/param.h*. Refer to your hardware manual for details on various displays, adapters, and controllers.

You can use *ioctl*(S) requests to map the display memory as follows:

MAPCONS	Maps the display memory of the adaptor currently being used into the user's data space. Not meant to work with MAPPGA1K.
MAPMONO	Maps the monochrome adapter's display memory into the user's data space.
MAPCGA	Maps the color adapter's display memory into the user's data space.
MAPPGA	Maps the professional graphics controller's display memory into the user's data space.
MAPEGA	Maps the enhanced graphics adapter's display memory into the user's data space.

MAPPGA1K — Maps the professional graphics controller 1K communications fifos (etc.) into the user's data space.

For example, the following code can be used to acquire a pointer to the start of the user data space associated with the color graphics adapter display memory:

```
char far  *dp;
int selector;
        .
        .
        .
fd=open ("/dev/color", O_WRONLY);
selector = ioctl (fd, MAPCGA,0);
dp = sotofar (selector, 0);
        .
        .
        .
```

Note that when the display memory is mapped into the user space, the adapter's start address registers are not set. The start address can be reset in two ways, so that the start address of the display memory corresponds to the upper left hand corner of the screen:

1. Switch modes with an *ioctl*() (the "switch" can be to the present mode). See the "Display Modes" section of this manual page.

2. Change the start address high and low address with the *in−on−port/out−on−port ioctl*().

The *in−on−port/out−on−port* ioctl()'s can also be used to determine the current value in the start address register, and then set up a pointer to point to the offset in the mapped−in data space.

Graphics Adapter Port I/O

You can use *ioctl*(S) to read or write a byte from or to the graphics adapter port. The *arg* parameter of the *ioctl* call uses the *io_arg* data structure:

```
struct port_io_arg {
    struct port_io_struct args[4];
};
```

As shown above, the *io_arg* structure points to an array of four *port_io* data structures. The *port_io* structure has the following format:

```
struct port_io_struct {
      char            dir;   /* direction flag (in vs. out) */
      unsigned_int    port;  /* port address */
      char            data;  /* byte of data */
};
```

You may specify one, two, three, or four of the *port_io_struct* structures in the array for one *ioctl* call. The value of *dir* can be either IN_ON_PORT to specify a byte being input to the graphics adapter port or OUT_ON_PORT to specify a byte being output to the graphics adapter port. *Port* is an integer specifying the port address of the desired graphics adapter port. *Data* is the byte of data being input or output as specified by the call.

If you are not using any of the *port_io* structures, load the *port* with 0, and leave the unused structures at the end of the array. Refer to the hardware manuals for port addresses and functions for the various adapters.

You can use the following *ioctl(S)* commands to input or output a byte on the graphics adapter port:

MGAIO	Inputs or outputs a byte on the monochrome adapter port as specified.
CGAIO	Inputs or outputs a byte on the color graphics adapter port as specified.
EGAIO	Inputs or outputs a byte on the enhanced graphics adapter port as specified.
PGAIO	Inputs or outputs a byte on the professional graphics adapter port as specified.

To input a byte on any of the graphics adapter ports, load *dir* with IN_ON_PORT and load with the port address of the graphics adapter. The byte input from the graphics adapter port will be returned in *data*.

To output a byte, load *dir* with OUT_ON_PORT, load *port* with the port address of the graphics adapter, and load *data* with the byte you want output to the graphics adapter port.

Function Keys

ioctl(S) can be used to define or obtain the current definition of a function key. The **arg** parameter of the **ioctl** call uses the **fkeyarg** data structure:

```
struct fkeyarg {
    unasigned int keynum;
    char keydef [MAXFK];
        /*Comes from
    char flen; ioctl.h via comcrt.h */
}
```

You can use the following **ioctl**(S) requests to obtain or assign function key defintions:

GETFKEY	Obtains the current definition of a function key. The function key number must be passed in **keynum**. The string currently assigned to the key will be returned in **keydef** and the length of the string will be returned in **flen** when the **ioctl** is performed.
SETFKEY	Assigns a given string to a function key. The function key number must be passed in **keydef** and the length of the string (number of characters) must be passed in **flen**.

Screen Attribute Sequences

The following character sequences are defined by ANSI X3.64-1979 and may be used to control and modify the screen display. Each **Pn** is replaced by the appropriate ASCII number (decimal) to produce the desired effect. The last column is for *termcap*(M) codes, where "n/a" means not applicable.

The use of 7 or 8 bit characters in the escape sequence is a valid invocation for each action defined. For example the ANSI ED command can be invoked via the "ESC [Pn S" (0x1b-0x5b-Pn-0x4a, 7 bit chars) sequence or the "CSI Pn J" (0x9b-Pn-0x4n, 8 bit chars) sequence.

ANSI	Sequence	Action	Termcap Code
ED (Erase in Display)	ESC [Pn J CSI Pn J	Erases all or part of a display. **Pn=0**: erases from active position to end of display. **Pn=1**: erases from the beginning of display to active position. **Pn=2**: erases entire display.	cd
EL (Erase in Line)	ESC [Pn K CSI Pn K	Erases all or part of a line. **Pn=0**: erases from active position to end of line. **Pn=1**: erases from beginning of line to active position. **Pn=2**: erases entire line.	ce
ECH (Erase Character)	ESC [Pn X CSI Pn X	Erases **Pn** characters	n/a
CBT (Cursor Backward Tabulation)	ESC [Pn Z CSI Pn Z	Moves active position back **Pn** tab stops.	bt
SU (Scroll Up)	ESC [Pn S CSI Pn S	Scroll screen up **Pn** lines, introducing new blank lines at bottom.	sf
SD (Scroll Down)	ESC [Pn T CSI Pn T	Scrolls screen down Pn lines, introducing new blank lines at top.	sr
CUP (Cursor Position)	ESC [P1 ; P2 H CSI P1 ; P2 H	Moves active position to location **P1** (vertical) and **P2** (horizontal).	cm

HVP (Horizontal & Vertical Position)	ESC [P1 ; P2 f CSI P1 ; P2 f	Moves active position to location **P1** (vertical) and **P2** (horizontal).	n/a
CUU (Cursor Up)	ESC [Pn A CSI Pn A	Moves active position up **Pn** number of lines.	up (ku)
CUD (Cursor Down)	ESC [Pn B CSI Pn B	Moves active position down **Pn** number of lines.	do (kd)
CUF (Cursor Forward)	ESC [Pn C CSI Pn C	Moves active position **Pn** spaces to the right.	nd (kr)
CUB (Cursor Backward)	ESC [Pn D CSI Pn D	Moves active position **Pn** spaces backward.	bs (kl)
HPA (Horizontal Position Absolute)	ESC [Pn ` CSI Pn `	Moves active position to column given by **Pn**.	n/a
HPR (Horizontal Position Relative)	ESC [Pn a CSI Pn a	Moves active position **Pn** characters to the right.	n/a
VPA (Vertical Position Absolute)	ESC [Pn d CSI Pn d	Moves active position to line given by **Pn**.	n/a
VPR (Vertical Position Relative)	ESC [Pn e CSI Pn e	Moves active position down **Pn** number of lines.	n/a
IL (Insert Line)	ESC [Pn L CSI Pn L	Inserts **Pn** new, blank lines.	al
ICH (Insert Character)	ESC [Pn @ CSI Pn @	Inserts **Pn** blank places for **Pn** characters.	ic

DL (Delete Line)	ESC [Pn M CSI Pn M	Deletes Pn lines.	dl
DCH (Delete Character)	ESC [Pn P CSI Pn P	Deletes **Pn** number of characters.	dc
CPL (Cursor to Previous Line)	ESC [Pn F CSI Pn F	Moves active position to beginning of line, Pn lines up.	n/a
CNL (Cursor Next Line)	ESC [Pn E CSI Pn E	Moves active position to beginning of line, Pn lines down.	n/a
SGR (Select Graphic Rendition)	ESC [0 m CSI 0 m	Resets bold, blink, blank, underscore, and reverse. **Color:** Restores normal selected colors.	n/a
	ESC [1 m CSI 1 m	Sets bold. **Color:** Sets intensity (changes *color* to *lt_color*).	n/a
	ESC [2h	Lock keyboard. Ignores keyboard input until unlocked. Characters are not saved.	n/a
	ESC [2i	Send screen to host. Current screen contents are sent to the application.	n/a
	ESC [2l	Unlock keyboard. Re-enable keyboard input.	n/a
	ESC [4 m CSI 4 m	Sets underscore. **Color**: No effect.	n/a
	ESC [5 m CSI 5 m	Sets blink. **Color:** Changes background *lt_color* to *color*; foreground blinks.	n/a

ESC [7 m CSI 7 m	Sets reverse video. **Color:** Uses reverse selected colors.	so
ESC [10 m CSI 10 m	Select primary font.	GE
ESC [11 m CSI 11 m	Select first alternate font. Allows ASCII characters less than 32 to be displayed as ROM characters.	n/a
ESC [12 m CSI 12 m	Select second alternate font. Toggles high bit of extended ASCII code before displaying as ROM characters.	GS

The following color codes and sequences are defined by International Organization for Standardization ISO DP 6429.

C	Color
0	Black
1	Red
2	Green
3	Yellow
4	Blue
5	Magenta
6	Cyan
7	White

ISO	Sequence	Action	Termcap Code
SGR (Select Graphic Rendition)	ESC [3Cm	**Color:** Selects foreground color *C*.	n/a
	ESC [4Cm	**Color:** Selects background color *C*.	n/a
	ESC [8m	Sets blank (non-display).	n/a

The following color codes and sequences are additional control sequences.

Cn	Color	Cn	Color
0	Black	8	Grey
1	Blue	9	Lt. Blue
2	Green	10	Lt. Green
3	Cyan	11	Lt. Cyan
4	Red	12	Lt. Red
5	Magenta	13	Lt. Magenta
6	Brown	14	Yellow
7	White	15	Lt. White

Name	Sequence	Action	Termcap Code
SGR	ESC [2;C1;C2 m	Color only. Sets foreground (C1) and background (C2) colors.	n/a
SGR	ESC [7;C1;C2 m	Reverse video. **Color:** Sets foreground (C1) and background (C2) reverse video colors.	n/a
SGR	ESC [3;0 m	Color only. Clears blink bit.	n/a
SGR	ESC [3;1 m	Color only. Sets blink bit.	n/a

SGR	ESC[4 m	Underscores. **Color:** No effect	n/a
n/a	ESC[=*Ss*;*Se* C	Sets cursor to start at scan line *Ss*, and ends at scan line *Se*. Default underline cursor is 6,7 for color adaptor and 11,12 for monochrome.	n/a
n/a	ESC[=*C* A	where *C* is 0-15. Sets border color on color adaptors. Color codes are same as ESC[2;C1;C2m.	n/a
n/a	ESC[Pn g	Accesses alternate graphics set. Not the same as "graphics mode." Refer to your owner's manual for decimal/character codes **(Pn)** and possible output characters.	n/a
n/a	ESC Q Fn '*string* '	Define function key **Fn** with *string*. String delimiters ' and ' may be any character not in *string*. Function keys are numbered 0 through 9 (**F1** = 0, **F2** = 1, etc.).	n/a
n/a	ESC[Pn z CSI Pn z	Pn should be equal to the number of the screen to switch to. Will only work if screen was configured for at boot, else no action will take place.	n/a

n/a	ESC [= *F* ; *T* B	Sets bell frequency and duration. *F* is the value loaded into the timer-counter, and *T* is the bell duration in 1/10ths of a second (initial settings are 1500 and 2/10ths of a second).	n/a

Files

/dev/console

/dev/tty[02 −*n*]

/dev/color

/dev/monochrome

/dev/ega

/dev/pga

See Also

console(M), ioctl(S), keyboard(HW), keymap(M), mapkey(M), mapchan(M), multiscreen(M), setcolor(C), stty(C), systty(M), termcap(M), tty(M)

Name

tty1[a-h] , tty1[A-H] , tty2[a-h] , tty2[A-H] – Interface to serial ports

Description

The **tty1[a–h]**, **tty1[A–H]**, **tty2[a–h]** and **tty2[A–H]** files provide access to the standard and optional serial ports of the computer. Each file corresponds to one of the serial ports (with or without modem control). Files are named according to the following conventions:

- The first number in the file name corresponds to the COM expansion slot.

- Lower case letters indicate no modem control.

- Upper case letters indicate the line has modem control.

tty1a and **tty1A** both refer to COM 1, whereas **tty2a** and **tty2A** both refer to COM 2.

For example, with a four port expansion board installed at COM 1 and a single port board installed at COM 2, you can access:

tty1a	tty1A
tty1b	tty1B
tty1c	tty1C
tty1d	tty1D
tty2a	tty2A

Each serial port has modem and non-modem invocations. The device names in the following table refer to the serial ports, with and without modem control. The first section of the table describes boards at COM 1 and the second section describes boards installed at COM 2. "Minor" is the minor device number for the port (see *mknod*(C)).

Serial Lines					
Board Type		Non-Modem Control		Modem Control	
		Minor	*Name*	*Minor*	*Name*
8 Port	4 Port / 1 Port	0	tty1a	128	tty1A
	4 Port	1	tty1b	129	tty1B
		2	tty1c	130	tty1C
		3	tty1d	131	tty1D
		4	tty1e	132	tty1E
		5	tty1f	133	tty1F
		6	tty1g	134	tty1G
		7	tty1h	135	tty1H
8 Port	4 Port / 1 Port	8	tty2a	136	tty2A
	4 Port	9	tty2b	137	tty2B
		10	tty2c	138	tty2C
		11	tty2d	139	tty2D
		12	tty2e	140	tty2E
		13	tty2f	141	tty2F
		14	tty2g	142	tty2G
		15	tty2h	143	tty2H

Interrupt Vectors:

All board(s) installed at COM 1 - 4
All board(s) installed at COM 2 - 3

For a list of I/O addresses, see the *Release Notes* furnished with your distribution.

Access

The files may only be accessed if the corresponding serial interface card is installed and its jumper I/O address correctly set. Also, for multi-port expansion cards, you must use the *mkdev*(C) program to create more than the default number of files. See *mkdev (C)* in the XENIX *Reference*.

The serial ports must also be defined in the system configuration. Check your hardware manual to determine how your system is configured, via a CMOS database or by switch settings on the main system board. If your system is configured using a CMOS database, the ports are defined in the database (see *cmos*(HW)). Otherwise, define the ports by setting the proper switches on the main system board. Refer to your computer hardware manual for switch settings.

It is an error to attempt to access a serial port that has not been installed and defined.

The serial ports can be used for a variety of serial communication purposes such as connecting login terminals to the computer, attaching printers, or forming a serial network with other computers. Note that a serial port may operate at most of the standard XENIX baud rates, and that the ports (on most computers) have a DTE (Data Terminal Equipment) configuration. The following table defines how each pin is used.

Pin	Description
2	Transmit Data
3	Receive Data
6	Request to Send
7	Signal Ground
8	Carrier Detect (Data Set Ready)
20	Data Terminal Ready

Only pins 2, 3, and 7 are necessary for a terminal (or direct) connection.

See *tty*(M) and *termio*(M) for the details of serial line operation in the XENIX system.

Files

/dev/tty1[a-h]
/dev/tty1[A-H]
/dev/tty2[a-h]
/dev/tty2[A-H]

See Also

cmos(HW), csh(C), cu(C), getty(M), mkdev(C), mknod(C) nohup(C), open(S), termio(M), tty(M), uucp(C)

Notes

If you login via a modem control serial line, hanging up logs that line out and kills your background processes. See *nohup*(C) and *csh*(C)).

You cannot use the same serial port with both modem and non-modem control at the same time. For example, you cannot use tty1a and tty1A simultaneously.

Use a modem cable to connect your modem to a computer.

Name

stty – Sets the options for the video monitor.

Syntax

stty [**−a**] [**−g**] [options]

Description

stty sets certain terminal I/O options for the device that is the current standard input; without arguments, it reports the settings of certain options. With the **−a** option, *stty* reports all of the option settings; with the **−g** option, it reports current settings in a form that can be used as an argument to another *stty* command. The following options describe control modes for the video monitor and other display devices attached to a display adapter. This information is hardware specific; refer to *stty*(M) for options generic to this utility. Additional details on the modes described here can be found in *screen*(HW) and *keyboard*(HW).

Control Modes for the Video Monitor

mono Selects the monochrome display as the output device for the console screen. This mode is valid if a standard monochrome adapter is present or if a standard enhanced graphics adapter (EGA) is present and the EGA is currently in one of the monochrome display modes.

color Selects a standard regular color display as the output device for the console screen. This mode is valid if a color graphics adapter is present or if a standard EGA is present and is currently in one of the color graphics compatibility modes.

enhanced
Selects the enhanced color display as the output device for the console screen. This mode is valid if an EGA is present and is currently in a non-monochrome display mode.

herc Selects the monochrome display as the output device for the console screen. This mode is valid if a Hercules monochrome graphics adapter is present and is currently in a full monochrome graphics mode.

pro Selects the professional color display for the output device for the console screen. This mode is valid if a standard professional graphics adapter is present.

Note that on systems with more than one display adapter, the keyboard multiscreen switchings only work on the current primary screen display adapter.

Control Modes for the Attached Display Devices

B80x25 Selects 80x25 black and white text display mode.

C80x25 Selects 80x25 color display text mode.

BG320 Selects 320x200 black and white graphics display mode.

CG320 Selects 320x200 color graphics display mode.

BG640 Selects 640x200 black and white graphics display mode.

The keyboard and display control modes above are valid for the following configurations: standard color graphics adapter (CGA) attached to an standard regular color display, standard enhanced graphics adapter (EGA) (modes 0-6) attached to a standard regular color display or an standard enhanced color display, and an standard professional graphics adapter (PGA) attached to a standard professional graphics display.

CG320_D
Selects EGA support for 320x200 graphics display mode (EGA mode D).

CG640_E
Selects EGA support for 640x200 graphics display mode (EGA mode E).

The two options above are only valid when an EGA is attached to a standard regular color display or an enhanced color display.

EGAMONO80x25
Selects EGA Mode 7 as the display mode. Emulates the support provided by the standard monochrome display adapter.

EGAMONOAPA
Selects EGA support for 640x350 graphics display mode (EGA mode F).

ENHMONOAPA2
Selects EGA mode F*.

The three options above are only valid when a standard EGA is attached to an IBM monochrome display.

ENH_B80x25
Selects enhanced EGA support for 80x25 black and white text display mode (EGA mode 2*).

ENH_C80x25
Selects enhanced EGA support for 80x25 color text display mode (EGA mode 3*).

CG640x350
Selects EGA support for 640x350 graphics display mode (EGA mode 10).

ENH_CG640
Selects EGA mode 10*.

The six options above are only valid when a standard EGA is attached to a standard enhanced color display.

MCAMODE
Reinitializes the monochrome graphics adapter.

Warning: If invalid options are used for any of these display modes, the behavior of the attached device(s) may be erratic.

See Also

console(M), ioctl(S), keyboard(HW), screen(HW), stty(C), tty(M)

Notes

Many combinations of options make no sense, but no checking is performed.

Name

tape - Cartridge tape device.

Description

The *tape* device implements the XENIX interface with a tape drive. Typically, the *tar*(C), *cpio*(C), *dd*(C), *backup*(C), *dump*(C), or *restore*(C) commands are used to access a tape drive.

A single tape drive with a raw (character, non-blocking) interface is supported. There are two standard tape device types. Devices beginning with the "r" prefix, (for "raw device"), should be used for most normal tape work, while devices with the "n" prefix, ("for no rewind on hold"), should be used for storing and restoring multiple files.

The following table summarizes the base naming conventions for the tape drives supported:

ct0,1	QIC24 unit 0,1
ct2,3	QIC11 unit 0,1
ctmini	floppy controller-based cartridge tape
mt0,1	reel to reel unit 0,1 1600 bpi
mt2,3	reel to reel unit 0,1 800 bpi
mt4,5	reel to reel unit 0,1 6250 bpi

tape(C) describes the commands used to access tape drives.

Files

/dev/rct0
/dev/nrct0
/dev/rct2
/dev/nrct2
/dev/rctmini

Notes

After certain tape operations are executed, the system returns a prompt before the tape controller has finished its operation. If the user enters another tape command too quickly, a "device busy" error is returned until the tape device is finished with its previous operation.

Periodic tape cartridge retensioning and tape head cleaning are necessary for continued error-free operation of the tape subsystem. Use *tape*(C), to retension the tape.

See Also

backup (C), cpio (C), dd (C), dump (C), format (C), tape (C), tar (C) restore (C),

Name

terminal – Login terminal.

Description

A *terminal* is any device used to enter and display data. It may be connected to the computer:

- By a serial wire, either direct or dialup
- As a virtual terminal, for example with emulator software
- Through a display adapter

A terminal has an associated device file **/dev/tty***.

Files

/dev/tty*

See Also

console(M), disable(C), enable(C), mkdev(C), serial(HW), stty(C), stty(HW), termcap(M), term(F), terminals(M)

04-01-87
SCO-518-210-014