The XENIX™ Operating System

for the IBM PC™ Operations Guide

The Santa Cruz Operation, Inc.

The Santa Cruz Operation, Inc.
500 Chestnut Street
P.O. Box 1900
Santa Cruz, California 95061
(408) 425-7222 · TWX: 910-598-4510 SCO SACZ

Release: 86-6-25-84-1.1/1.5

Contents

4 Using File Systems

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 all the 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.

1.2 The System Manager

Every XENIX system should have at least 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 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
BREAK	Delete	Stops current action and returns to the shell. This key is also called the INTERRUPT key.
BACKSPACE	Backspace	Deletes the first character to the left of the cursor.
CNTRL-D	Ctrl-D	Signals the end of input from the keyboard; also exits current shell.
CNTRL-H	Erase	Deletes the first character to the left of the cursor. Also called the ERASE key.
CNTRL-Q	Ctrl-Q	Restarts printing after it has been stopped with CNTRL-S.
CNTRL-S	Ctrl-S	Stops printing at the standard output device (does not stop the program).
CNTRL-U	Ctrl-U	Deletes all characters on the current line. Also called the KILL key.
CNTRL-\	Ctrl-\	Quits current command and creates a *core* file (Recommended for debugging only).

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.

1.5 Using This Guide

The tasks presented in this guide range from very simple tasks 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 introduces this guide.

Chapter 2 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 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 explains how to create and mount file systems, how to set permissions, and how to keep the system secure.

Chapter 5 explains how to maintain free space on the root file system and other file systems.

Chapter 6 explains how to create backup copies of the root file system and other file systems.

Chapter 7 explains how to add terminals.

Chapter 8 explains how to solve system problems such as a jammed lineprinter or a forgotten password.

Chapter 9 explains how to create a multiple system mailing network with Micnet.

Appendix A presents a list of the XENIX system special files, and explains how to use these files when creating and maintaining file systems.

Appendix B presents a list of commonly used XENIX directories and log files.

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:

   ```
   Xenix 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, type *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 typing *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 **
```

If it displays this message, you must reload the system. You can do this by resetting the computer and repeating the steps given in the previous section. For instruction 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 CNTRL-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 have the finished, the system displays the "login:" prompt. You may then log in as a normal user, as decribed in the XENIX *User's Guide*, or as the super-user, as described in the next section.

To choose system maintenance mode, type 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 CNTRL-D key until it appears.

To log in as the super-user, follow these steps:

1. When you see the "login:" message, type the super-user's login name:

   ```
   root
   ```

   and press the RETURN key. The system asks for the super-user's password.

2. Type the super-user's password and press the RETURN key. The system does not display the password as you type it, so type 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 CNTRL-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. Type:

   ```
   /etc/shutdown
   ```

   and press the RETURN key. The system loads the command which in turn asks for the number of minutes you wish to elapse before the computer stops:

   ```
   Minutes till shutdown? (0-15):
   ```

3. Type 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, displays the message:

   ```
   ** Normal System Shutdown **
   ```

   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. Type:

   ```
   /etc/haltsys
   ```

   and press the RETURN key. The system displays the message:

   ```
   ** Normal System Shutdown **
   ```

   and stops.

You may now turn off the computer.

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* 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. Type

```
mkuser
```

and press the RETURN key. The system displays the following message:

```
            Newuser
            -------
        Add a user to the system

    Do you require detailed instructions? (y/n/q):
```

3. Type the letter *y* (for "yes"), if you want information about the program, otherwise type the letter *n* (for "no"). (Type *q* (for "quit") only if you wish to stop the program and return to the system.)

   When the program continues, it asks you 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 acceptable for the user John Doe.

4. Type the new name, and press the RETURN key. The program asks you 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 his account for the first time. Once in the account, the user should create a new password for himself, one that is hard to guess. (See the section "Changing Your Password" in the XENIX *User's Guide*.)

5. Type the password carefully, and press the RETURN key. After you have entered the password, the program asks for the group name:

   ```
   Enter group:
   ```

   The group name is the name of the group of users to which the user will belong when he logs in. Users in a group have access to a common set of files and directories. The group name is optional. If not given, the the XENIX system's common group "group" (with group ID 50) is used.

6. Type the group name, and press the RETURN key. If you do not wish to enter a group name, just press the RETURN key. After you have entered the group, the program asks for a comment:

   ```
   Please enter Comment      >--------------------
                             >
   ```

   A comment is information about the new user, such as his department 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.

7. Type 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 shows what you have typed 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:

   ```
   Do you want to change anything? (y/n/q):
   ```

8. Type the letter *y* (for "yes") and press the RETURN key, if you wish to change something. Type *n* (for "no") and skip to step 10 if you wish to complete the new account. (Type *q*, for "quit", only if you wish to leave the program and abort the new account.)

9. If you type *y* in step 8, the program asks for the item you wish to change:

   ```
   username
   password
   group
   comment
   ```

   Type the name of the item you wish to change, then type the new item. The program changes the item and returns to step 8.

10. Once you type *n* in step 8, 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.

11. Type *y* if you wish to add another user. Otherwise, type *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. See the XENIX *User's Guide* for details.

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 XENIX *User's Guide*). 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. Type

    ```
    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. Type the new password and press the RETURN key. The command does not display the password as you type it, so type carefully. The command asks you to type the password again:

    ```
    Retype new password:
    ```

4. Type 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 RETURN *User's Guide*.

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 also need to choose a minimum number of weeks that a user must wait before changing his password and a maximum number of weeks that a user may go without changing the password. A common pair of minimum and maximum values is 2 and 8.

To set the minimum and maximum dates, type:

```
pwadim -min num -max num  login-name
```

where *num* is a number in the range 1 to 63, and *login-name* is simply the login name of the user whose password you are administering.

If you are unsure of the current minimum and maximum values for a password, you can display them by typing:

```
pwadmin -d login-name
```

This command does not change the current values.

If you wish to force a user to change his password immediately, type:

```
pwadmin -f login-name
```

The user is asked 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 typing:

```
pwadmin -n login-name
```

The system will no longer prompt for changes.

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 0 to 65535. Both the group name and ID must be unique, i.e., they must be 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 typing:

   ```
   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", type:

```
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 XENIX *User's Guide* 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. Type:

   ```
   cd /etc
   ```

3. Use the **cp** command to make a copy of the */etc/passwd* file. Type:

   ```
   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. Type:

   ```
   mv passwd passwd-
   ```

10. Use the mv command to make the edited file the new */etc/password* file. Type:

    ```
    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, then 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. Type:

```
cd /etc
```

3. Use the cp command to make a copy of the */etc/passwd* file. Type:

   ```
   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 and substitute 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 user ID "205".

7. Exit the text editor.

8. Use the mv command to save the old */etc/passwd* file. Type:

   ```
   mv passwd passwd-
   ```

9. Use the mv command to make the edited file the new */etc/passwd* file. Type:

   ```
   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, type:

```
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. Type:

   ```
   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. Type:

   ```
   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, then type:

   ```
   cat /dev/null >/usr/spool/mail/login-name
   ```

   and press the RETURN key.

6. When the user's home directory and mailbox are empty, type:

   ```
   cd /usr
   ```

   and press the RETURN key. The user's home directory cannot be removed until you have moved to another directory.

7. Type:

   ```
   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 asks for the login name of the user you wish to remove:

   ```
   Enter name of id to be removed.
   ```

8. Type the user's login name. You should now see the message:

   ```
   Removing user name from the system. CONFIRM? (y/n/q):
   ```

9. Type *y* (for "yes") to remove the user from the system. Otherwise type *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. Type *y* to remove another user. Otherwise, type *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", or "uucp", or a system ID (user ID below 200). Also, the program cannot remove a user account if the user's mailbox still has mail in it, or if the user's home directory contains files other than *.profile*.

Chapter 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 files systems on hard and on floppy disks. 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 the file system, 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. Type

   /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, type *y* and press the RETURN key to overwrite the data and continue creating the file system. Otherwise, type *n*. If you type *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
```

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 filename of disk drives are given 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. Type 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 may 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. If a user is in the directory, you must wait for him to leave that 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, type:

```
/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 message:

```
umount: Device busy
```

if you or another user is currently in the directory containing the file system.

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. Type

   ```
   /etc/format
   ```

   and press the RETURN key. The program formats the disk.

If for some reason you have not properly inserted the disk 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, an ordinary user may access those files and directories for which he has 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 the 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 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-------", type:

```
chmod go-r receivables
```

and 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--
```

and 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 belongs when he creates the file.

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 Encrypting Text Files

You can usually ensure both the privacy and safety of files by setting the appropriate permissions and maintaining system security. However, these methods cannot protect text files from unauthorized individuals who have logged in as the super-user. You can protect files from an unauthorized super-user by using the **crypt** command to encrypt the file. Encryption changes the contents of the file into meaningless characters. The encryption is carried out by means of a key which you supply. The process can be reversed, and the file returned to meaningful text, by giving the same key.

For example, to encrypt the contents of the file *projects.june* and store the encrypted file in the file *projects.secret*, type:

```
crypt <projects.june >projects.secret
```

and press the RETURN key. The command asks for the key with the message:

```
Enter key:
```

Type a string of characters (it may be up to eight characters long) and press the RETURN key. The program encrypts the file.

To restore the encrypted file *projects.secret* and display it on the screen, type:

```
crypt <projects.secret
```

and press the RETURN key. The command asks for the key. Type the same key you used to encrypt the data and press the RETURN key. The program displays the restored data.

4.5.4 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.

4.5.5 Copying Floppy Disks

To ensure against the loss of data stored on floppy disks, you can use the **dd** command to make copies of floppy disks on new, formatted disks.

To make a copy of a disk, follow these steps:

1. Insert the disk to be copied into floppy drive 0.

2. Insert an empty, 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. Type:

```
dd if=/dev/fd0 of=/dev/fd1 count=blkcount
```

and press the RETURN key. The *blkcount* must be the number of blocks on the disk to be copied (see Appendix A for details).

The 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) invoked 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. Type:

```
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.

Note that when you start the system, the contents of the */usr/adm/pacct* file is usually saved in the file */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 with which to select other information to display.

To display the average size of each process, type

```
acctcom
```

The command displays the basic information plus the average size of each process.

To display basic accounting information about a specific command, type:

```
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,

```
acctcom -n units
```

displays each entry for the system command **units**.

To display information about the number and size of input and output counts, type:

```
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, type

```
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 systen, 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*, type:

```
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 type at your terminal to the terminals of all users currently logged in.

To send a message, type:

```
wall
```

and press the RETURN key. Type the message, pressing the RETURN key to start a new line if necessary. After you have typed the message, press the CNTRL-D key. The command displays the message on all terminals in the system. To leave the wall command, press the CNTRL-D key. 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*, type:

```
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*, type:

```
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, type

```
mail login-name
```

and press the RETURN key. The *login-name* must be the login name of the recipient. To send a message, type the message, press the RETURN key, and then press the CNTRL-D key. 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*, type:

```
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, type:

```
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 typing:

```
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*, type

```
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. Type *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 floppy disk, of the files in the root directory and other regularly mounted file systems. A backup allows the system manager 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 Strategies for Backups

The system manager should back up the root directory (and any other mounted file systems) on a regular basis. In particular, he should make daily copies of all files modified during the day, and 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* program and the **tar** command.

The *sysadmin* program is a formal maintenance program for systems that require a rigorous schedule of file system backups. Such systems usually have many users and a large number of files that are modified daily. The program automatically locates modified files, copies them to disk, and optionally produces a list of the files.

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. 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 once a day 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.

6.3 Using the *sysadmin* Program

The *sysadmin* program performs daily and periodic backups, lists backup files, and restores individual files from backup disks. 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.

6.3.1 Creating Backups

To create backups with the *sysadmin* program, you need several formatted floppy disks. The exact number depends on the number of files to be copied; for example, some periodic backups require as many as nine disks. For details on how to format a floppy disk, see the section "Formatting Floppy Disks" in Chapter 4.

To create a backup, follow these steps:

1. Log in as the super-user.

2. Type:

   ```
   sysadmin
   ```

   and press the RETURN key. The program displays a file system maintenance menu.

   ```
                File System Maintenance
                -----------------------
   Type         1 to do daily backup
                2 to do a periodic backup
                3 to get a backup listing
                4 to restore a file
                5 to quit
   ```

3. Type *1* for a daily backup or *2* for a periodic backup. Then press the RETURN key. Note that if the system has never had a periodic backup, it automatically performs one, even if you have chosen a

daily backup.

4. Insert a floppy disk in drive 1, wait for the drive to accept the disk (all drive noise should stop), and press the RETURN key. 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 floppy disks. If the disk runs out of space, the program displays the message:

   ```
   Change volumes
   ```

5. Remove the first disk and insert a new disk. Wait for the drive to accept the disk, then press the RETURN key. The program continues to copy files to the new disk. 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 disk as you remove it from the disk drive. For example, label the first disk "Volume 1", the second "Volume 2", and so on.

6.3.2 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*. This listing is especially convenient if you keep detailed records of the files copied in each backup. The backup listing is available after every daily or periodic backup.

To get the listing, follow these steps:

1. Log in as the super-user.

2. Type

   ```
   sysadmin
   ```

   and press the RETURN key. The program displays the system maintenance menu.

3. Type *3* and press the RETURN key. The program asks you to reinsert the backup disks in the same order that you inserted them during the backup.

4. Insert the first disk, wait until the drive accepts the disk, then press the RETURN key. The program automatically reads the filenames

from the backup disk and places them in the list file. When the program has read all the names, it asks for the next disk.

5. Remove the first disk and insert the next. Wait for the drive to accept the disk and press the RETURN key. Repeat this step until all disks have been read.

You may produce a printed copy of the backup list by printing the list at the lineprinter. Type

```
lpr /tmp/backup.list
```

and press the RETURN key. After printing the file, you should remove it from the */tmp* directory with the **rm** command to save space.

6.3.3 Restoring a Backup File

You can restore files from the backup disks by invoking the *sysadmin* program and selecting the fourth item in the menu. You will need the complete set of backup disks containing the latest version of the file you wish to restore. You will also need the "full pathname" of the file you wish to restore. This is the name given for the file in the backup listing.

To restore a file, follow these steps:

1. Log in as the super-user.

2. Type

   ```
   sysadmin
   ```

   and press the RETURN key. The program displays the file system maintenance menu.

3. Type *4* and press the RETURN key. The program asks you to type the full pathname of the file you wish to restore.

4. Type the pathname and press the RETURN key. The program asks for another pathname.

5. Repeat step 4 to enter another pathname or press the RETURN key to continue the program. If you press the RETURN key, the program asks you to insert the first disk in the backup set.

6. Insert the disk, wait for the drive to accept the disk, and press the RETURN key. The program displays the inode numbers of the files you have given, then asks for the volume number of the backup disk containing the files.

7. Insert the disk containing the files, type the volume number, and press the RETURN key. The program searches the disk for the specified files. If found, the files are copied to your current directory. If not found, the program asks for the next volume.

8. Repeat step 7 until all files have been found and copied.

6.4 Using the tar Command

The tar command copies specified files and directories to and from floppy disks. On systems with one or two 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.

6.4.1 Copying Files to a tar Disk

You can copy a small number of files or directories to a floppy disk with the tar command. The command has the form:

```
tar cvf specialfile files
```

The *specialfile* must be the name of the special file corresponding to the disk drive containing the disk to receive the files (see Appendix A). The *files* are the names of the files or directories you wish to copy.

To use the tar command, you need a formatted floppy disk and the names of the files and/or directories you wish to copy. For details about how to format a disk, see the section "Formatting Floppy Disks" in Chapter 4. If you give a directory name, the command copies all files in the directory (including subdirectories) to the disk.

For example, to copy the files a, b, and c to the disk in the disk drive */dev/fd0*, type

```
tar cvf /dev/fd0 a b c
```

and press the RETURN key.

6.4.2 Restoring Files from a tar Disk

You may also use the tar command to restore files from a disk. The command simply copies all files on the disk to your current directory. In this case, the command has the form:

```
tar xvf specialfile
```

The *specialfile* must be the name of the special file corresponding to the disk drive containing the **tar** disk.

For example, to restore files from the disk in the drive */dev/fd0*, type

```
tar xvf /dev/fd0
```

and press the RETURN key. The command copies files on the disk in the drive to the current directory.

Since the **tar** command only copies files to the current directory, make sure you are in the desired directory before you invoke the command. You can change to the desired directory with the **cd** command.

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, and lineprinters to the system.

To add a peripheral device, the system manager must make the physical connection between the device and the computer, then use the correct system commands to enable the device for operation. This chapter explains how to use system commands to enable a device for use. It also describes how to maintain the devices once they are added.

Note that all physical connections between a device and the system are device-dependent. For information about these connections, see the hardware manual provided with the device.

Make sure that any serial wire is connected to your modem, terminal, printer, etc. or not connected to your computer at all. An unterminated wire connected to your computer can considerably reduce system performance. Always disconnect unused lines at your computer.

7.2 Using Multiple Screens

The multiple screen feature uses the console and **/dev/[tty02...tty10]** device files. These device files provide character I/O between the system and the IBM screen and keyboard. Each device file has an independent screen display and keyboard access. The files are used chiefly to display characters at the IBM screen devices and read characters from the keyboard.

Although all ten files may be open at the same time, only one can be active at any given time. The active device displays its own screen and takes sole possession of the keyboard. Rotating through the ten screens is accomplished by typing Ctrl−PrtSc (using the Ctrl key and the PrtSc key, the shifted asterisk, directly below carriage return). Any active screen may be selected by typing alt−Fn where Fn is one of the ten function keys on the far left side of the keyboard. For example, typing

```
alt−8
```

will switch you to screen 8, corresponding to */dev/tty08*. Refer to the XENIX *Reference* for more on **console**(M).

Please note that error messages from the kernel appear on the console screen. In such cases, the console screen displays instead of the current screen.

Small version XENIX (for systems with less than 384K random access memory) only supports the console. Do not try to use the additional device files.

7.3 Adding a Terminal

You can add up to two additional terminals, resulting in simultaneous access to the system for up to three users. Many different terminals work well with the XENIX operating system. An abbreviated list of recommended terminals is given in *terminals*(M) in the XENIX *Reference*.

Before you can add a terminal, you must know how to connect the terminal to a serial line on the computer. Physical connections for the terminal are usually explained in the terminal's hardware manual. The names of the available serial adaptor ports are given in Appendix A of this guide. Once a terminal has been connected, you may then enable the terminal for use with the **enable** command.

To add a terminal, follow these steps:

1. Using the recommended procedure in the terminal's hardware manual, connect the terminal to one of the RS−232 serial lines on the computer itself. Make sure that the terminal is compatible with the line configuration (for a description of the serial ports, see Appendix A).

2. Log in as the super−user on the console. Plug in the terminal and turn it on.

3. Use the **enable** command to enable the terminal.

   When using the **enable** command, make sure that you wait a full minute between each use of the command. Failure to do so can cause a system crash.

   The command has the form:

   ```
   enable specialfile
   ```

   where *specialfile* is the name of the serial line to which the terminal is attached.

   ```
   enable /dev/tty11
   ```

   enables the terminal connected on serial line */dev/tty11*. Likewise, to enable the alternate serial port, type:

   ```
   enable /dev/tty12
   ```

4. Press the RETURN key several times. The system should display a "login:" message. When it does, you may log in and begin work.

NOTE: If no "login:" message appears on the screen, if random characters appear, or if the terminal does not respond to your attempt to log in, you may need to change the baud rate (or "line speed") of the terminal to match the serial line. The default baud rate is 9600, unless altered by your system administrator. Check the rate and parity scheme recommended for the terminal, then check to see how the serial line is set by using the stty(C) command on the serial line in question. You can change the baud rate until you log out with the stty(C) command described in the section "Changing Serial Line Operation". Permanent changes are made by editing */etc/ttys* as described in the section "Setting Serial Line Baud Rate".

Another reason for no "login:" is more than one serial card set to either the main or alternate serial adapter. This will cause both cards to not function. Note that tty13 and tty11 refer to the same (main) serial line, and tty14 and tty12 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.

7.4 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 a serial line printer. Before you can remove a terminal, you must disable it with the **disable** command.

To remove a terminal, follow these steps:

1. Log in as the super−user the console.

2. Use the **disable** command to disable the terminal.

   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.

   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/tty11
   ```

   disables the terminal connected to serial line */dev/tty11*.

3. Turn off the power to the terminal.

4. Disconnect the terminal from the system.

7.5 Setting the Terminal Type

Several XENIX utility programs (for example the visual editor, **vi**(C), and the visual shell, **vsh**(C)), and many "screen−oriented" application programs, must make use of detailed information about your terminal. These programs communicate with the terminal hardware to move the cursor, highlight an area of the screen, clear the screen, and the like.

XENIX 1.1 sets the terminal type for you during installation. The console and ttys02−10 are automatically assigned default settings of ibm or ibmc. If you have a monochrome card, your terminal type is automatically set to ibm. If you have a color card, your terminal type is automatically set to ibmc. If at some future time you change from a monochrome to a color card, or if you add a terminal, you can use this section as a guide to set your terminal type.

The standard XENIX shell **sh**(C) (command interpreter) sets aside a variable, **TERM**, to refer to the name of your terminal. This variable is then passed on to programs that you invoke, so your terminal type is available to them if they need it. The file */etc/termcap* (short for "terminal capabilities") is an ASCII database that describes features of over 100 popular terminals. A list of terminals supported by XENIX, along with their names, may be found in **terminals**(M). Also refer to the **termcap**(M) manual page.

The easiest way to set the **TERM**, variable is with the **tset** command. The **tset** command determines the name of the line you have logged in on (e.g. tty02), then reads the file terminaltype for that line.

The */etc/ttytype* file supplied with an IBM PC XENIX distribution may look like this (or it may have entries for ibmc):

```
ibm      console
ibm      tty02
ibm      tty03
ibm      tty04
ibm      tty05
ibm      tty06
ibm      tty07
ibm      tty08
ibm      tty09
ibm      tty10
dialup   tty11
dialup   tty12
dialup   tty13
dialup   tty14
```

The lines labelled "dialup" refer to the serial lines.

The file */etc/profile*, which is read and executed by the **sh**(C) shell every time you log in, will set your terminal type based on the entry in */etc/ttytype*. It can contain the lines:

```
eval ‘tset -m dialup:?ibm -m ibmc:ibmc -m: ibm -e -s -r‘
export PATH SHELL TERM TERMCAP TZ
```

If you log in on the console, or tty02 through tty10 (recall that these logical devices are attached to the IBM keyboard and screen), the following will appear:

```
Terminal type is ibm
```

If you log in on one of the serial port lines (tty11 – 14) the following prompt appears:

```
TERM = (ibm)
```

At this prompt, you may press RETURN to indicate that this default value is suitable. In this case, **tset** will clear the screen, and announce the terminal type with:

```
Terminal type is ibm
```

If you are using another terminal type, respond with any of the names in **terminals**(M), and **tset** will automatically set your **TERM** variable, and announce the terminal type on the screen.

```
TERM = (ibm) vt100

Terminal type is vt100
```

If your response is not one of the names in **terminals**(M), or if you type the name incorrectly, you may see:

```
Terminal type unknown
```

In this case, you should log out and log back in again, then supply the correct name at the prompt.

You may modify both */etc/ttytype* and the **tset** command line in */etc/profile* to suit your particular needs. Examples for customizing terminal settings are provided on the **tset**(C) manual page in the XENIX *Reference*.

7.6 Changing Serial Line Operation

Whenever you enable a terminal with the **enable** command, the system automatically sets the characteristics of the serial line such as baud rate and parity 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**(C) (for "set tty") command.

You can display the current operating characteristics of a serial line by typing

```
stty
```

at the terminal connected to that line. If it is impossible to login at that terminal, you may use another terminal to display the characteristics. Log in as the super−user at the console and type

stty <*specialfile*

where *specialfile* is the name of the device special file corresponding to the serial line (see Appendix A). For example, the command

```
stty </dev/tty11
```

displays the current characteristics of the serial line named */dev/tty11*. 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. Log in as the super−user at the other terminal and type

stty *baud−rate* <*specialfile*

where *baud−rate* is the terminal's desired baud rate 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/tty11
```

changes the baud rate of the serial line */dev/tty11* 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 or to another type of serial device which requires parity generation and detection, and special input and output processing.

For a full description of this command, see *stty*(C) in the XENIX *Reference*.

7.7 Setting Serial Line Baud Rate

The changes that **stty**(C) makes to any aspect of a serial line (described in the preceding section) are volatile in the sense that they disappear when the line 'closes' (usually when the current user logs out). To make a change that will carry over to the next login, follow these steps:

1. Log in as root (super−user) on the console. Make sure nobody is logged in on the line you want to change.

2. **disable** the line you want to change. See **disable**(C) for instructions. When using the **disable** or **enable** commands, make sure that you wait a full minute between each use of the command. Failure to do so can cause a system crash.

3. Edit the file */etc/ttys*. The format of this file is described on the **ttys**(M) manual page. Find the entry corresponding to the serial line whose speed you want to change, then change the one character 'mode' to reflect the new speed. The **getty**(M) page furnishes a table of corresponding speeds.

4. **enable** the serial line. See **enable**(C) for instructions. The speed change will be read by **getty**(M), and a login message will appear at the new baud rate on the port.

7.8 Modem Usage under XENIX

7.8.1 Serial Lines

XENIX for the IBM PC supports modem control on up to two serial ports. The following device names refer to the serial ports with and without modem control.

/dev/tty11	uses main serial adaptor without modem control.
/dev/tty12	uses alternate serial adaptor without modem control.
/dev/tty13	uses main serial adaptor with modem control.
/dev/tty14	uses alternate serial adaptor with modem control.

7.8.2 Dialing Out From Your IBM PC

The **cu**(C) and **uucp**(C) utilities are used to call remote systems under XENIX. The file */usr/spool/uucp/L−devices (L−devices)* must be edited to your configuration before **cu**(C) will function properly. You must tell it whether or not you are using a modem, and the proper baud rate for the connection.

The *L−devices* file contains lines which specify the device for the line, the call−unit associated with the line (0 for direct ''hardwired'' lines) and the baud rate which 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)	tty11	0	9600*
main serial adapter	(modem)	tty11	tty11	1200*
alternate serial adapter	(direct)	tty12	0	9600*
alternate serial adapter	(modem)	tty12	tty12	1200*

*example baud rate. Substitute the baud rate for your other 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 lines are only used on lines dialing in, for outgoing calls use lines with no modem control.

Note that these *L−devices* lines are compatible; they can exist in your *L−devices* file at the same time. To invoke **cu** for a direct line, type:

```
$ cu −l tty11 dir
```

To invoke **cu** for a modem connection, type:

```
$ cu −l tty11 −a tty11 <telno>
```

For some modems, the telephone number you wish to dial can be part of the command line. Other modems (such as the Racal Vadic) require that you type in various codes before you can type in the telephone number. For such modems, <telno> is 0, since **cu** will not be dialing it.

Make sure that serial wires are not left dangling. An unterminated line connected to your computer can considerably reduce system performance. Unplug a modem wire at the computer.

7.8.3 Dialing Into Your IBM PC Under XENIX

To allow dialing into your IBM PC, you must **enable(C)** a serial line that recognizes modem control signals. When using the **enable(C)** command, make sure that you wait a full minute between each use of the command. Failure to do so can cause a system crash. Type:

```
# disable tty11
# enable tty13
```

if you are using the main serial adapter, or

```
# enable tty14
```

if you are using the alternate serial adapter. Note that tty13 and tty11 refer to the same (main) serial line, and tty14 and tty12 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 system problems.

7.8.4 Hayes Modem Settings

Proper modem configuration is necessary when using **uucp**. Modem settings differ for each modem. Consult your modem manual for the proper switch settings.

If you have a Hayes Smartmodem™ 1200, switches 3 and 8 should be down. This allows both the XENIX and DOS communications systems to work.

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, the file */usr/lib/uucp/L−dialcodes* must be used. 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* refer to the XENIX *User's Guide* chapter on "Building a **uucp** System"

NOTE: **uucp**(C) automatically dials a Hayes Smartmodem. See the XENIX *Release Notes* for a list of supported modems.

7.8.5 Modem Control

uucp must be used with serial lines with modem control for dialing in. When dialing out **uucp**(C) must use no−modem serial lines and the login must be disabled on the port **uucp** is using. Check the section above "Serial Lines" to see which lines support modem control.

The modem must be set to respond to DTR. Check you modem manual for instructions on setting DTR. Note that the dialer port must be disabled, and it is best to have the port owned by *uucp*.

7.9 Adding Lineprinters

7.9.1 Adding a Parallel Lineprinter

You can add a paralleled lineprinter to the system by connecting it to the parallel port on the parallel expansion card.

You must configure your system for your particular printer. Multiple printer cards can cause interrupt interference.

Plug your parallel printer into the parallel port on the parallel expansion card in the back of your IBM PC, log in a super−user and type the command

```
# ls −l /dev/lp
```

the system responds with (for example):

```
c-w--w--w- 1 root     6, 0 Jun 9, 1984  20:11 /dev/lp

          6 is the  major number   0 is the minor number
```

Make sure the major number is 6 and the minor number is 0. If not, remove it (rm /dev/lp) and type the command:

```
# /etc/mknod /dev/lp c 6 0
```

Turn on your printer, and type

```
# date > /dev/lp
```

The date should be printed on your line printer. If so, the printer configuration is probably correct.

The rest of this section discusses steps to take if the procedure is unsuccessful.

1. If you are using an IBM parallel printer adapter (the small card), make sure any parallel ports on other cards are disabled. Otherwise there may be competition between the parallel ports for interrupts.

2. If you are using a monochrome adapter use its parallel port instead of the port on any other card you may have (such as a multi-function card).

3. If you have a parallel printer adapter on a multi-function board and a color graphics board, make sure that the card is set up to respond from the first parallel port (*lp1*). Some multi-function board manuals refer to the correct device as *lp1*, while other manuals refer to it by its address, Hex 378.

   If the above steps fail to enable your printer, you may have a non-standard configuration; make two special files:

   ```
   # /etc/mknod /dev/lpbw c 6 1
   # /etc/mknod /dev/lpcolor c 6 2
   ```

   and type:

   ```
   # date > /dev/lpbw
   ```

   if this didn't work try:

   ```
   # date > /dev/lpcolor
   ```

   One of these should trigger a response from your printer. Remake */dev/lp* using the major and minor device numbers (from above) that provoked a response. For example:

   ```
   # rm /dev/lp
   # /etc/mknod /dev/lp c 6 1
   ```

   Your system should now be ready for use of the **lpr** utility. Remove the two special test files you created. Refer to the XENIX *Reference* for more information on **lpr**(C).

   NOTE: If you are still having problems, check the power supply to your printer, your printer/computer configuration, and run hardware diagnostics.

7.9.2 Serial Printer Setup

To set up a serial lineprinter, make sure you are in System Maintenance Mode.

To enter System Maintenance Mode login in as super−user and enter the super−user password when XENIX prompts you; At the pound sign prompt (#) type:

```
kill −1 1
```

You are now in single−user System Maintenance Mode. Use a text editor to edit the baud rate in the shell script called *lpopen* in */etc*. */etc/lpopen* is called from */etc/rc* to set the modes of a serial printer when the system goes multi−user. */etc/lpopen* that contains the following:

```
if ( test −c /dev/lp ) && tty −s < /dev/lp
then
          ( stty 1200 −parity −nl −echo −tabs < /dev/lp
            while true
            do
                    sleep 30000
            done ) > /dev/lp
fi
```

This shell script will test the device */dev/lp* to make sure that it is a serial port. The lineprinter speed is the specified baud rate. Consult your printer's hardware manual for for this information (in the example, the baud rate is set to 1200.) Edit the rate to the desired baud rate and exit the editor.

To make a serial printer node, remove the parallel printer (rm /dev/lp) and type

```
ln /dev/tty11 /dev/lp
```

to use a serial printer with the main serial port, or:

```
ln /dev/tty12 /dev/lp
```

to use a serial printer with the alternate serial port.

Leave System Maintenance Mode by typing CTRL−D. After you type CTRL−D (entering multiuser mode) the new speed is in effect. See the XENIX *Operations Guide* for more information about XENIX modes.

7.10 Adding a Second Hard Disk

You can give the system extra room for storing users' files and directories by adding a second 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 XENIX 1.1.

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. Insure that the second drive passes the manufacturers diagnostics before running XENIX.

To add a second hard disk, follow these steps:

1. Connect the hard disk, then boot the system and enter system maintenance mode.

2. Make device nodes corresponding to the second drive. Type the following command lines:

   ```
   #   /etc/mknod /dev/hd1      b   3 16
   #   /etc/mknod /dev/hd1ds    b   3 29
   #   /etc/mknod /dev/hd1bb    b   3 30
   #   /etc/mknod /dev/hd1p1    b   3 25
   #   /etc/mknod /dev/hd1p2    b   3 26
   #   /etc/mknod /dev/hd1p3    b   3 27
   #   /etc/mknod /dev/hd1p4    b   3 28
   ```

   Most users will probably want to put the user file system on this drive, so we suggest:

   ```
   # /etc/mknod /dev/usr b 3 31
   # ln /dev/usr /dev/rusr
   ```

   See **mknod(C)** for more information on the **mknod command**.

3. Run **fdisk** (see **fdisk(C)** for more information). to partition the disk. To do this type:

   ```
   # /etc/fdisk -f /dev/hd1
   ```

   At this point set up the partitions as desired. The partition to be used must be the active partition when finished.

4. Scan the second disk for bad blocks. To do this type:

   ```
   # /etc/badblkutil
   ```

   You will be prompted by the program. Type the following as requested (Do not type the comments given in parentheses):

   ```
   C                (for create)
   /dev/hd1         (for data partition)
   /dev/hd1bb       (for bad block partition)
   y                (for surface analysis)
   press RETURN     (for Known bad block number?)
   y                (for print bad block table)
   ```

5. Make a file system on */dev/usr*. If your second disk is to be used entirely for XENIX type:

   ```
   # /etc/mkfs /dev/usr 10064
   ```

   This creates a file system on the entire disk. After invoking the **mkfs** command, this message appears:

   ```
   This will destroy the contents of your
   disk; do you wish to continue?
   ```

   Answer appropriately, 'y' or 'n'.

To make the file system if you're not using the ENTIRE disk for XENIX.

1 Calculate the size (in Kb) of the partition you have created using this formula:

   Subtract 9 from the number of cylinders allocated to XENIX. Multiply that number by 34.

2 An example:

   The user makes the XENIX partition 250 cylinders.
   250 - 9 * 34 = 8194

3 Your command line would be;

```
# /etc/mkfs /dev/usr 8194
```

6. If you are allocating part of your disk to DOS, you will want to create an alias for the DOS partition in the file */etc/default/msdos*. Identify the partition DOS resides on by running fdisk(C) with the following command:

   ```
   fdisk f /dev/hd1
   ```

   Choose the "Display Partition Table" option. The first line of the table corresponds to */dev/hd1p1*, the second line corresponds to */dev/hd1p2*, and so on. Add a line to */etc/default/msdos* which contains a DOS drive identifier and the appropriate partition name such as:

   ```
   D=/dev/hd1p4
   ```

We recommend using the second disk for */usr*. To mount */usr* on the second disk follow this procedure.

1. Type:

   ```
   # /etc/mount /dev/usr /mnt
   ```

   to mount the new hard disk in the directory */mnt*.

2. Type:

   ```
   # copy -r -m /usr /mnt
   ```

   to copy the contents of the */usr* directory to the new hard disk. This operation may take several minutes. You can then use the lc command to display the new contents of the disk and to make sure all files and directories have been copied.

3. Type:

   ```
   # cd /
   ```

   to leave the */mnt* directory.

4. Type:

   ```
   # rm -r /usr/*
   ```

   to remove all files and directories contained in the /usr directory. Before typing this command, make sure that you have made a copy of every file and directory in */usr* or check the contents and integrity of the files copied into */mnt* before removing */usr*.

5. Type:

   ```
   # /etc/umount /dev/usr
   ```

   to unmount the new disk from the */mnt* directory.

6. Type:

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

   to mount the new disk in the */usr* directory. You may access all files and directories in */usr* as before.

Note that once the new hard disk has been added to the system, you should type the command:

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

Add the following line to */etc/checklist*.

```
/dev/usr
```

This allows your new filesystem to be automatically checked by **fsck** when appropriate.

You'll probably want to add command lines to your */etc/rc* file which mount */usr* on invocation of multi−user. */etc/rc* contains these lines which begin with a pound sign (#). The pound sign acts as the shell **sh** comment symbol. Delete the pound signs, add the "2>/dev/null" and the lines will be executed, mounting */usr* on invocation of multi−user.

```
CHANGE:
   #/etc/mount /dev/usr /usr
   #if test $? = 2; then
   #echo "Cleaning /dev/usr"
   #fsck -y -t /tmp/fsck$$ /dev/usr
   #/etc/mount /dev/usr /usr
   #fi >/dev/console 2>&1

TO:
   /etc/mount /dev/usr /usr 2>/dev/null
   if test $? = 2; then
         echo "Cleaning /dev/usr"
         fsck -y -t /tmp/fsck$$ /dev/usr
         /etc/mount /dev/usr /usr
   fi >/dev/console 2>&1
```

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 typed at the keyboard. This abnormal operation can occur if a program stops prematurely as a result of an error or the user pressing the BREAK key.

To restore the terminal to normal operation, follow these steps:

1. Press the CNTRL-J key. The system may display an error message. If it does, ignore the message.

2. Type:

   ```
   stty sane
   ```

   and press the CNTRL-J key. The terminal does not display what you type, so type carefully.

After pressing the CNTRL-J key, the terminal should be restored and you may continue your work.

8.3 Freeing a Jammed Lineprinter

Lineprinter errors, such as running out of paper, can cause the *lpd* program to "lock up" the printing queue, preventing the current file and any other files in the queue from being printed. The *lpd* program is the "lineprinter daemon", the program which does the actual printing for the system print command lpr.

To free a jammed lineprinter, follow these steps:

1. Log in as the super-user.

2. Type

   ```
   ps -a
   ```

   to find the process identification number (PID) of the *lpd* program. (The PID is in the first column of the display.) The command display should look similar to this:

```
PID     TTY     TIME    COMMAND
34      01      0:08    sh
135     01      0:25    lpd
```

3. Type

   ```
   kill PID
   ```

   and press the RETURN key. The *PID* is the process identification number of the the program.

4. Locate and fix the error that caused the lineprinter to become jammed.

5. Type

   ```
   cd /usr/spool/lpd
   ```

   to change to the lineprinter spool directory. This directory temporarily holds the files to be printed.

6. Type

   ```
   rm -f lock
   ```

   to remove the lineprinter spool's lock file. This frees the queue and allows printing to continue.

After freeing the lineprinter, you must issue another lpr command to start printing.

8.4 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 type 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. Type

   ```
   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. Type:

```
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, type

```
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 nommal operation, follow the instructions in the section "Restoring a Nonechoing Terminal" in this chapter.

8.5 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.

8.6 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 typing:

```
lc -a
```

and pressing the RETURN key. If you need to, you can remove most invisible files from a directory by typing:

```
rm .[a-z]*
```

and pressing the RETURN key. Remaining files can be removed individually.

8.7 Restoring Free Space

The system displays an "out of space" message whenever a file system has little or no space left to work. To restore system operation, you must delete one or more files from the file system. To delete files, follow the steps outlined in the section "Maintaining Free Space" in Chapter 5.

8.8 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.

8.9 Restoring an Inoperable System

On very rare occasions, one or more of the critcal 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.

8.10 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 and 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 and 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 1) the system has displayed at the system console a message beginning with "panic:" and/or 2) the system refuses to process all input (including INTERRUPT and CNTRL-\\s+1 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 corresponding 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.

8.11 Changing XENIX Initialization

One common problem is how to adapt 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*, */etc/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*, */etc/profile*, and *.profile* files contain XENIX commands and comments and have the command file format described in the chapter "The Shell" in the XENIX *User's Guide*.

8.11.1 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. Type

```
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 general 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 type the command correctly, otherwise the system will reject any incorrect command and all following commands when it reads the file 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.

8.11.2 Changing the */etc/profile* Files

/etc/profile contains a shell variable, TZ, which specifies the user's local time zone. At present, TZ is set at

```
PST8PDT
```

which means:

Pacific Standard Time, 8 hours past Greenwich
Mean Time, Pacific Daylight Time

To change the local time zone, edit the */etc/profile* file, using a text editor. Locate the TZ variable and change the PST8PDT line to the local equivalent. For example, on the east coast, change the TZ to:

```
EST4EDT
```

which means:

Eastern Standard Time, 4 hours past Greenwich
Mean Time, Eastern Daylight Time

8.11.3 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 the chapter "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, using a text editor. Commands can be added or removed as desired.

8.11.4 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 such messages 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.

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 must decide how the computers are to be connected, make the actual physical connections, then use 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 will require you to 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 are users on the computer; however, you may use any name you wish. The name must be unique and consist of letters and digits. The Micnet programs only use 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 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 similar 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 (tty01), then the topology map should look like this:

```
a ---------- b ---------- c
tty01
```

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
        tty01   tty02  tty03    tty04
```

If a computer does not have enough available serial lines to meet its needs, you can make the lines available by removing 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 may be any in the normal range for XENIX serial lines (i.e., 110,300...19200). 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.

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 an 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 will be 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 (i.e., the aliases that refer to just one user) and the login names of each corresponding user.

2. Make a separate list of the group aliases (i.e., 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, 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. Type

   ```
   netutil
   ```

   and press the RETURN key. The program displays the network utility menu. The install option is the first item in the menu.

3. Type the number *1* and press the RETURN key. The program displays the following message.

   ```
   Compiling new network topology
   Overwrite existing network files? (yes/no)?
   ```

   Type *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 disk with the save option described later in this chapter.

   Once you have typed *y* the program displays the following message.

   ```
   Enter the name of each machine
           (or press RETURN to continue installation).
    Machine name:
   ```

4. Enter a machine-name by typing the name and pressing 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, simply press the RETURN key to continue to the next step. The program displays the names you have entered and asks if you wish to make changes.

5. Type *y* (for "yes") if you wish to enter all the names again. Otherwise, type *n* (for "no") or just press the RETURN key to move on to the next step. If you type *n*, or RETURN, 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 asks for the names of the computers connected to the next computer.

7. Repeat step 5 for all remaining computers. As the program asks 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 a loop, it ignores the machine-name that creates the loop and asks 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. Type *y* if you wish to enter the connections again. Otherwise, type *n* to move to the next step. If you type *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, type the serial line name or number (e.g., tty03 or 3) for first computer in the pair and press the RETURN key. The program displays the message:

```
  Tty on b:
```

10. Type the serial line name for second computer in the pair and press the RETURN key. The program displays the message:

```
  Speed:
```

11. Type the speed (e.g., 9600) and press the RETURN key. The program asks for the serial lines and transmission speed of the next pair.

12. Repeat step 7 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. Type *y* if you wish to enter the serial lines and speeds again. Otherwise, type *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 a name by typing 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. Type *y* if you wish to enter the user names again. Otherwise, type *n*. If you type *n*, the program asks for the users on the next computer.

16. Repeat steps 13 and 14 for all remaining computers. When you have given names of users for every computer, the program asks if you wish to enter aliases.

    ```
    Do you wish to enter any aliases? (yes/no)?
    ```

17. Type *y* if you wish to enter aliases. Otherwise, type *n* to complete the installation. If you type *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, type 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, type the login name of that user and press the RETURN key. The program asks for another alias.

    If, on the other hand, the alias is to name several users, type the login names of the users, OR if one or more of the users to be named by the alias are already named by other aliases, type the aliases instead of the login names, OR if all the users on one computer are to be named by the alias, type the machine-name instead of the login names. In any case, make sure that each item typed 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, type 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 typed, press the RETURN key. The program asks 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. Type *y* if you wish to enter all aliases again. Otherwise, type *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 floppy 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, you need to format a floppy disk (see the section "Formatting Floppy Disks" in Chapter 4).

To save the files, follow these steps:

1. Log in as the super-user.

2. Type

   ```
   netutil
   ```

   and press the RETURN key. The program displays the network utility menu.

3. Insert a blank, formatted floppy disk into disk drive 1. Wait for the drive to accept the disk.

4. Type the number *2* and press the RETURN key. 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 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. Insert the micnet disk into disk drive 1 and wait for th edrive to accept the disk.

3. Type:

   ```
   netutil
   ```

   and press the RETURN key. The program displays the network utility menu.

4. Type the number *3* and press the RETURN key. The program begins to copy the network files to the appropriate directoiries. It displays the name of each file as it copies it. Finally, it displays the message:

   ```
   Enter the name of this machine:
   ```

5. Type the machine name of the computer you are at 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 each computer of the network, 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. Type:

   ```
   /etc/netutil
   ```

   and press the RETURN key. The system displays the network utility menu.

3. Type *4* and press the RETURN key. The program searches for the */etc/systemid* file. If it finds the file, it starts the network. If not, it asks 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 netweork for the first time, type *n* 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

```
/usr/lib/mail/daemon.mn
```

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 7.

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 the section "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.

If all users have received the message, the network is properly installed. 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 a record 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 typing

```
cat /usr/spool/micnet/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 at.

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 similar to:

```
daemon.mn: running as MASTER
Local system: a
Remote system: b, /dev/tty02
Tue Sep 27 22:30:35 1983
```

A startup message is added to the file each time the network starts successfully. If the message is not present, then 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 yet no handshake message appears, then the serial line may be improperly connected or damaged. 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, you can make the network log a record of the errors it encountered while transmitting and a record of each transmission by stopping and then restarting the network with the -e and -x switches of the *netutil*start option.

The -e (for "errors") switch causes error messages generated by the program to be copied to the file. 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, e.g., make sure it does not lie near any electric motor, generator, or other source of electromagnetic radiation. Also, make sure the cable is in good condition.

The -x (for "transmissions") switch causes a record of normal transmissions between computers to be copied to the file. 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.

To start the network with the -e or -x switches, type *y* (for "yes") when the start option asks if you wish to log errors or transmissions.

9.5.3 Stopping the Network

You can stop the network with the stop option of the *netutil* program. The 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. Type

   ```
   netutil
   ```

   and press the RETURN key. The program displays the network utility menu.

3. Type *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 Miscellaneous section of the XENIX *Reference Manual.*

In general, a copy of a file should be made before making any changes. 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.

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 the XENIX *Reference Manual.*

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 corresponding 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 that you may use for the IBM PC.

Block I/O	
Special Filename	**Disk Drive**
/dev/fd048ds9	Floppy Drive 0
/dev/fd148ds9	Floppy Drive 1
/dev/wn0 /dev/root /dev/usr	Hard Disk Drive

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 1024 bytes of storage. The **mkfs**, **df**, and **quot** commands use block size when creating or reporting the status of a file system. The size of the 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.

A.5 Gap and Block Numbers

The gap and block numbers are used by the **mkfs** command to describe how the blocks are to be arranged on a disk. The the gap number for the hard disk is 3, and the blocking factor is 34.

A.6 Terminal and Network Requirements

The enable and disable commands used to add and remove serial lines from a system and the install option of the *netutil* program used to build a network require the names of the serial lines through which a terminal or network is to be connected. The names of serial line available on the IBM PC is /dev/tty10. The character I/O files corresponding to these serial lines can be found in the */dev* directory. Note that the files */dev/console*,*/dev/tty0*, through */dev/tty9* represent "hardwired" devices and are not available for connection to terminals.

The XENIX system supports optional modem control on two serial ports. These ports have the following device names:

/dev/tty11	uses main serial adaptor without modem control
/dev/tty12	uses alternate serial adpator without modem control
/dev/tty13	uses main serial adpator with modem control
/dev/tty14	uses alternate serial adaptor 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 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
cp
date
dump
dumpdir
echo
expr
fsck
login
mv
passwd
rm
sh
sleep
stty
su
sync
tar
restor
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 /*etc* 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/mtab	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 file 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

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.

/usr/adm/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.

/usr/spool/at/past
Records each use of the **at** command.

/usr/spool/micnet/*/LOG
Records transmissions between machines in a Micnet network. The * must be the name of a remote machine connected to the current machine.

Appendix C
Using DOS and XENIX
On the Same Disk

C.1 Introduction

Many users purchased the PC−DOS or MS−DOS (DOS) operating system with their IBM PC. This appendix 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 PC just to run XENIX.

Several programs make this possible. The most important is a utility program called **fdisk** which is available in DOS and XENIX versions. This section explains how to use **fdisk** to run DOS and XENIX on the same hard disk.

Each version of **fdisk** is documented in the respective operating system's manual. **fdisk**(C) is found in the XENIX *Reference*.

Fdisk is interactive, so it is easy to use. 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.

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 cylinders. The number of cylinders and their size vary depending on the size of the hard disk. A partition consists of a group of cylinders. There are exactly four partitions. Each partition can have a different operating system and associated directories and files.

The **fdisk** command allows you to specify which partition is "active". This means that when you turn on (boot) your IBM PC, the operating system installed in the active partition will start running.

The **fdisk** command allows you to specify the number of cylinders 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. A 10 megabyte disk contains 305 cylinders. We recommend that you use at least 6 megabytes for the XENIX partition. Given the above you will set aside at least 183 cylinders for XENIX.

The **fdisk** command allows you to specify where the partition begins. **Fdisk** will not allow you to construct overlapping partitions. We recommend that you use the first partition for XENIX. When you are running XENIX, the device name of the partition running XENIX will be */dev/hd0*.

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 cylinder, the number of cylinders used, and the associated operating system.

If you wish to set up XENIX on a hard disk which previously contained only DOS, first back up all DOS files and directories. Use **fdisk**(C), under XENIX, to change the size of the first partition to at least 6 megabytes. Then install XENIX in the first partition. You can reinstall DOS in a separate partition.

If you wish to switch operating systems once you have set up separate XENIX and DOS partitions, use **fdisk** to change the current active partition. Turn the IBM PC off. When you re−boot, the operating system associated with the active partition will start running.

There are a set of other utility programs which help you bridge the two operating systems. These are the XENIX commands, such as doslsand doscat, described in the XENIX manual page dos(C). These programs allow you to use DOS files and directories which reside in a non–active DOS partition while running XENIX.

Index