MICROSOFT®

Introducing XENIX®

A Tutorial for the Microsoft® XENIX® System V/286 Operating System

Introducing XENIX®
MICROSOFT®

Introducing XENIX®

XENIX System V/286
Operating System

Microsoft Corporation

Document Number: 291190040-500-R00-1085
Part Number: 091-192-001

Contents

Preface

Your computer has been set up, it's working, and you'll be able to use it to do all sorts of wonderful things — or so you've been told. Maybe you've even been lucky enough to have someone who "knows computers" sit down with you and explain how it works.

Now comes the moment of truth. You are sitting all alone in front of the keyboard and the blank screen. (Well, nearly blank. Actually, it says "login:" and it's waiting for you to do whatever it was you were told to do.) By now, your memory may have gone blank, too. You know there are some XENIX® manuals that give all sorts of instructions on using the computer, but you don't know where to start looking. All you want right now is for someone to lead you step-by-step through your first session with the computer.

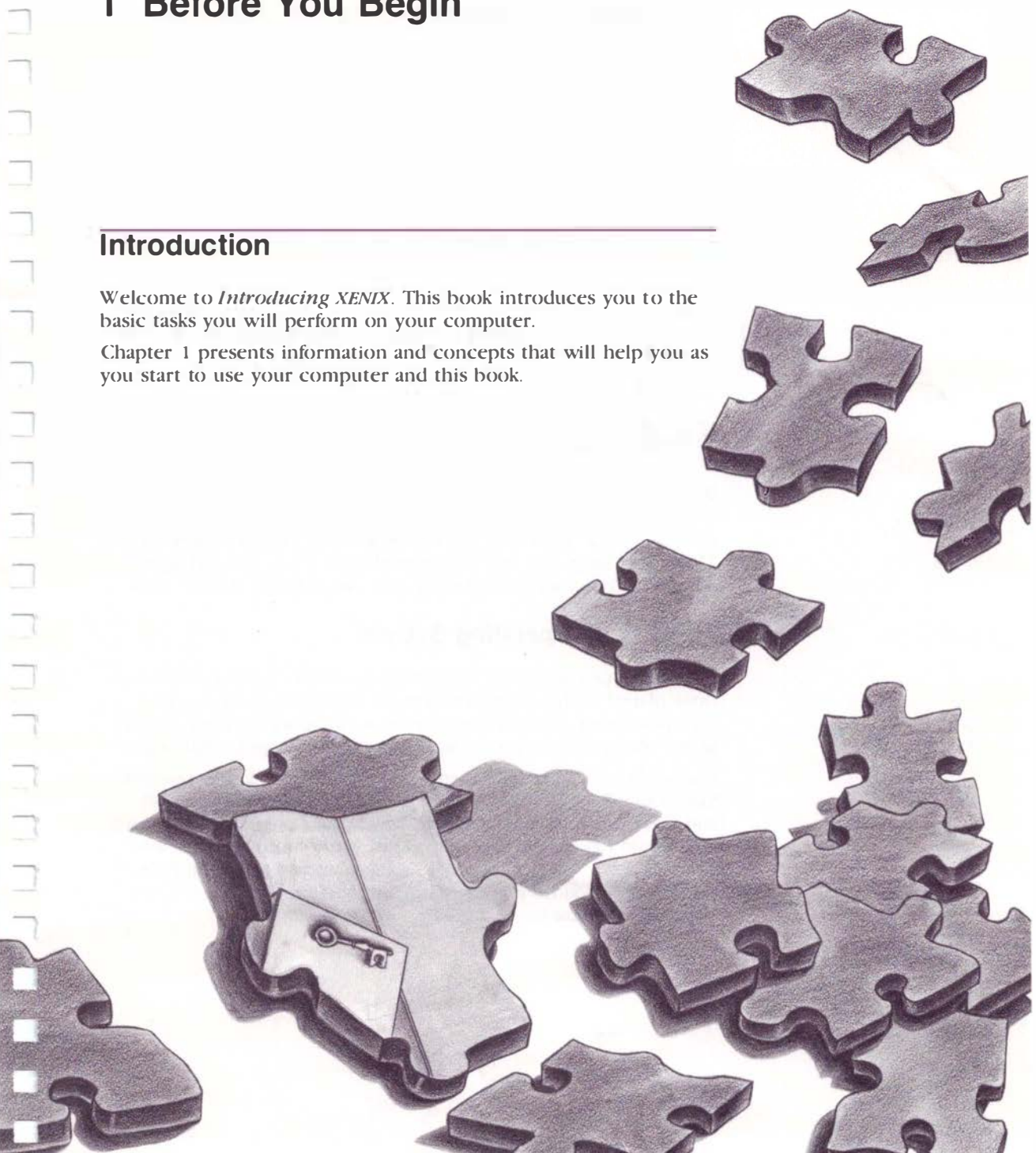
That's why we wrote this book. We know that computers sometimes have a reputation for being complicated and hard to learn, and we want to help you see that they don't have to be. By following the exercises in this book, you'll be able to begin using your computer right away to do some simple, everyday tasks. Then, you can advance to some of the other things it can do — some of the timesaving, amazing things that computers are famous for.

1 Before You Begin

Introduction

Welcome to *Introducing XENIX*. This book introduces you to the basic tasks you will perform on your computer.

Chapter 1 presents information and concepts that will help you as you start to use your computer and this book.



Why Read This Chapter?

The first thing you might want to do with your computer is put your hands on the keys and try it out. If so, this book was designed with you in mind, because you will begin to work at the keyboard very soon. First though, you need to learn a few important details that will make it easier to learn to use your computer. It is important to read this chapter before you begin to work through the exercises in this book.

Is This the Right Book For You?

Introducing XENIX is intended for people who have little or no experience using computers. If you are an experienced computer user who just needs to learn XENIX, you may find the XENIX *User's Guide* more helpful.

What Is XENIX?

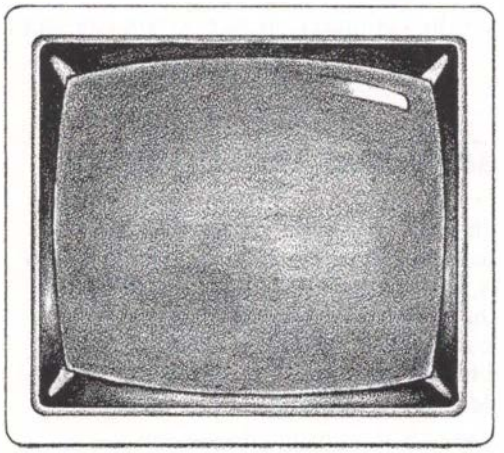
If you look at your computer, you won't see "XENIX" anywhere. That's because XENIX is not part of the computer "hardware" — the screen, the keyboard, the cables, and the circuit boards inside. Those are things you can see, or could see if you opened up the computer. You cannot *see* XENIX; you can only see what it *does*.

XENIX Is an Operating System

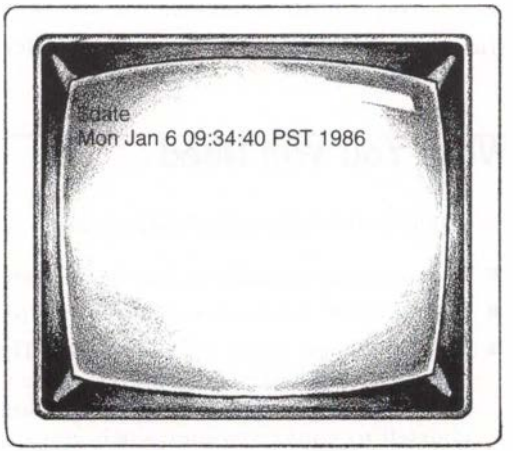
XENIX is your computer's "operating system" — the set of instructions placed in the computer to make it work for you. In more technical terms, an operating system is "software" — a set of programs that translates your commands to the computer hardware.

If you are unfamiliar with computers, this may be a new concept. You could think of it like this: it is as if you speak only English and you want to talk to someone who speaks only Swahili. Clearly, you would need an interpreter. XENIX is the interpreter between you and the computer. You tell XENIX what you want the computer to do (you do this by typing on the keyboard) and XENIX tells the hardware how to accomplish the task.

Without the XENIX Operating System, your computer doesn't know how to do anything for you. Nothing can happen.



When XENIX is installed, your computer knows how to do the things you tell it to do. It works.



This idea will become clearer as you learn more about your computer. For the time being, just remember that the XENIX Operating System is so closely connected with your computer that learning XENIX is pretty much the same thing as learning to use your computer. The advantage of learning XENIX is that it works on different computers: once you learn to use it on *your* computer, you automatically know how to use any other computer that uses the XENIX Operating System.

XENIX Is Multi-Tasking

An important feature of XENIX is that it is "multi-tasking." This means that you can tell it to work on one process while you go on to do something else on the computer. In fact, XENIX can work on several tasks at once.

What You Will Learn from This Book

In this book, you will work through some introductory steps for learning to use XENIX. You will learn how to get started and how to do some common tasks, such as typing in information, using electronic mail, and looking at files of information stored in the computer. You will also learn what to do if you make a mistake. For a list of tasks covered in this book, refer to the Table of Contents or to the Index, under "Try it out."

Many of the features available on the XENIX system will not be covered in this book because there are hundreds, and there isn't enough space here. For example, sometimes there are four ways to do the same thing; in this book, you will usually learn just one way. To learn other methods or more details on the topics discussed, you will be referred to the appropriate XENIX manual.

This symbol is used to point out these references.

What You Will Need

To proceed, the following things must be completed:

- Your computer hardware has been correctly hooked up.
- An "account" has been set up for you to use.
- You have been given a "login name." This may be your own name, your department name, or some other name you've been told to use. You will need your login name to identify yourself to your computer when you start working.
- The demonstration files and directories (located under /demo) are available for you to use.

If you're unsure about any of these things, check with your system manager — the person in charge of keeping your computer system running properly.

Using This Book

This section tells you how to best use this book as you learn to use XENIX on your computer.

Start at the Beginning

This might seem obvious, but the point is that this book is intended to be used as a tutorial. You will need to learn the tasks presented in one section before you move on to the next.

At the beginning of each of the next four chapters, you will find a section called "What You Will Learn in This Chapter." This is a list of specific tasks you will be able to perform once you've finished the chapter. A review of the tasks appears at the end of the chapter so you can check to see how your learning process is going. Don't be concerned if you can't do each task from memory. As long as you know where to find the instructions, you can always figure out how to do the task.

Using the Index

If you need to refer back to an idea and you can't remember where it was discussed, be sure to check the Index. All of the important ideas presented in this book are referenced in the Index.

New Words

Words that may be unfamiliar to you are explained in the text when they are first used. They are also defined in the "Terms" section at the back of this book for easy reference.

You Can't Harm Your Computer

Most beginning computer users are a bit cautious when they start using their computers. It's natural to wonder what will happen if you push the wrong button. Be assured that you can't harm your computer just by using it, even if you do something wrong.

Don't be concerned with any stories you may have heard about destroying all your information with one wrong move. For example, as you work through this book, you'll be using some demonstration information that is already stored in your computer. This information has built-in protection: you can't destroy anything important. Feel free to try things out even if you're not absolutely sure what will happen.

Practicing What You Learn

Introducing XENIX explains some basic computer concepts that may be new to you. After you read about a new concept, you will have a chance to practice it. Practice sessions are set off from the rest of the text under "Try it out." The steps are listed in sequence, so be sure to do them in order.

If You Make a Mistake or Receive a Strange Response

If you follow all the steps listed for a task and you receive an unexpected response, don't panic. Just read a little further in the book. Some of the things that could go wrong are listed for you along with instructions on how to correct the situation. Look for the heading, "If It Didn't Work...."

If you try a procedure and receive a message that shows you have done something wrong, chances are you have simply typed something wrong. Try the exercise over again at least once. Some of the messages you might receive, along with suggestions to correct the errors, are listed in the "Messages" section in the back of this book.

To Stop a Process

If you start a process and then decide that you want to stop it, try pressing the DELETE (DEL, INTERRUPT) key a few times. This stops most programs and commands from continuing and in most cases, will return you to your starting point.

If You Are Really Stuck

If you've tried everything you can think of and nothing works, as a last resort, you can usually "log out," then "log in" again (both are discussed in Chapter 2, "Getting Started") and start the task over from the beginning. Note however, that if you are editing a file (discussed in Chapter 4, "Creating and Changing Files With the Vi Editor"), you will have to leave the file before you can log out.

Some Things You Need to Know

On the next few pages is some general information about using XENIX. You will need to know these things before you begin using your computer.

XENIX Can Tell the Difference

To XENIX, there is a difference between the number "zero" (0) and the capital letter "O". XENIX also distinguishes between the number "one" (1) and the small letter "l". These characters may look similar to you, but they mean different things to the computer, so be sure you use the correct key. Most computers display the zero on the screen with a diagonal line through it (Ø).

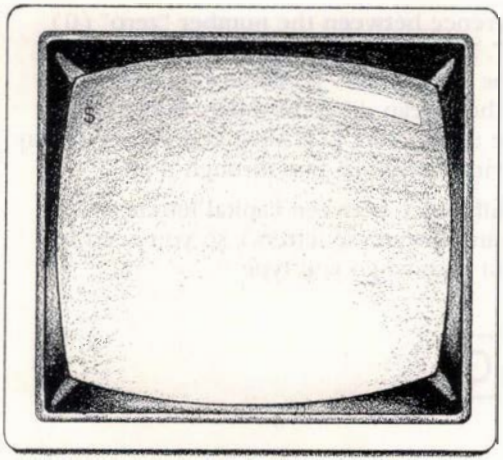
XENIX can also tell the difference between capital letters and small letters (uppercase and lowercase letters), so you need to be careful to use the right case when you type.

is not the same as


is not the same as


The System Prompt

The "system prompt" is the symbol you see on the screen that indicates the computer is waiting for you to tell it to do something. In XENIX, the prompt is a dollar sign ($). You will see the $ prompt after you log in, which is the first thing we discuss in Chapter 2, "Getting Started."

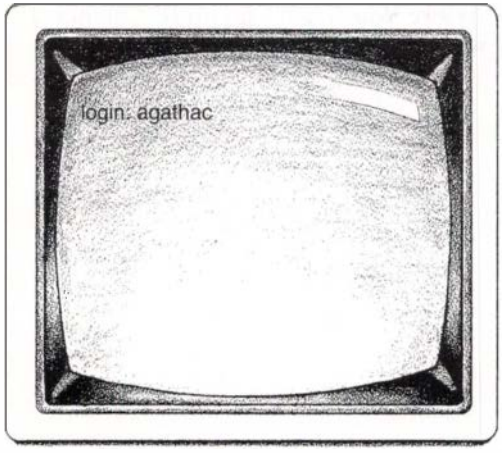


The ENTER Key

Information you type at the keyboard usually is not sent to the computer's memory until you press the ENTER key. In the exercises in this book, you will be reminded when to press ENTER so you will develop the habit of doing so after every command.

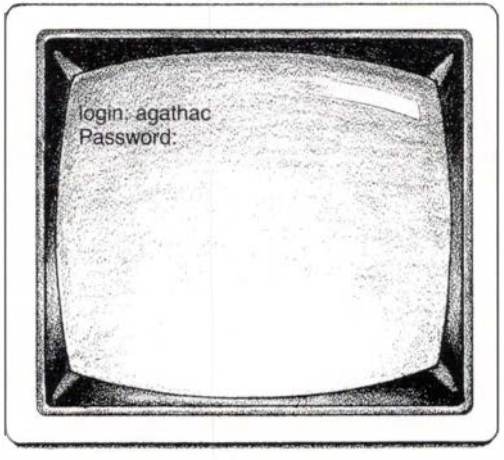
Note On your computer, the key might be labeled ENTER, RETURN, RET, NEWLINE, or ←┘.

You type something at the keyboard and it shows up on your screen.



You press ENTER

What you type is sent to the computer.
You receive a response from the computer.

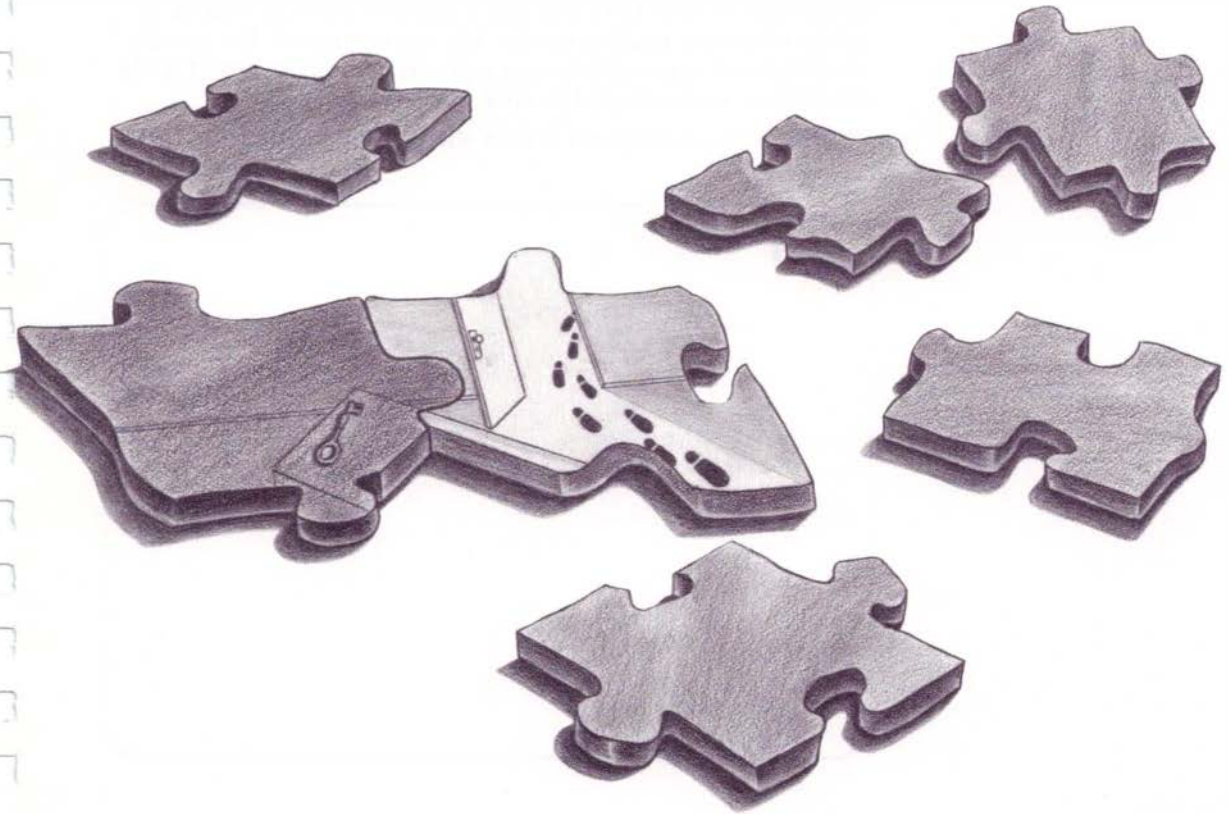


What Comes Next?

Now that you have read "Before You Begin," go on to Chapter 2, "Getting Started" to start working with your computer.

2 Getting Started

In the last chapter, you learned some basic information to prepare you for learning XENIX. In this chapter, you will begin to use your computer to perform some simple tasks that you are likely to do frequently with XENIX.



What You Will Learn in This Chapter

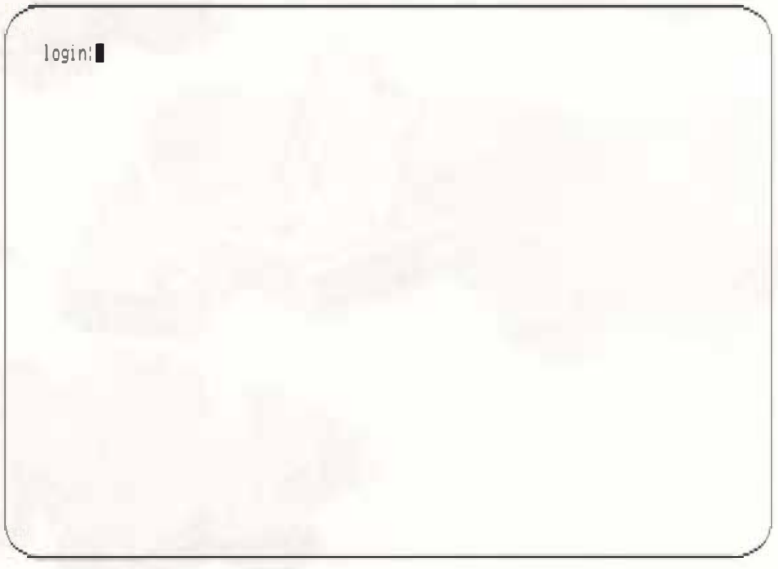
By the time you finish this chapter, you will be able to:

- Log in to your computer
- Use commands to display information from XENIX, such as the date and who else is using the system
- Change your password
- Log out

Logging In

Before you can make the computer start to work for you, you need to "log in." Logging in opens a line of communication between you and the computer. It's a little like having a telephone sitting next to you. Until you dial a number, you can't use it to talk to someone. Logging in also tells the computer that you are an authorized user and prevents unauthorized people from using information you've stored in your computer.

Right now, your computer screen should look like this:



If your screen is *completely* blank or if it displays nonsense characters when you type, get help from your system manager. First, be sure that both the computer and the monitor (screen) are turned on.

The Cursor

The lighted shape on your screen following "login:" is the "cursor." (It's probably either a blinking line or small box.) The cursor's function is to tell you where you are on the screen. If you're used to a typewriter, you could think of the cursor as the place on the page where the next character will show up when you type.

In the examples in this book, the cursor looks like this:

Login Procedure

To log in, use the following procedure.

Try it out: log in

1. Type your login name, which you should have received from your system manager. For example, suppose Agatha Christie were using the computer to write one of her mystery novels. Her login name might be "agathac". Once she's typed it in, her screen will display:

   ```
   login: agathac
   ```

   Your screen should look the same except that your own login name will be displayed.

   This book uses "agathac" as the login name in the examples. Each time you see "agathac," you should interpret it as your own login name.

2. Next, press the ENTER key.

   After a moment, your computer screen will display this message:

```
Password:
```

   If you don't have a password, you will not be asked for one.

3. If your system manager gave you a password, type it in and then press the ENTER key. Notice that your password does not appear on the screen as you type. This is to keep someone else from seeing it as you type it in.

If You Make a Typing Error

If you make a typing mistake, you can correct your mistake if you have not gone on to the next line. Press the BACKSPACE key enough times to back up over your error. Then, type the correct text and continue with what you were doing. For example, let's say you typed the following instead of "agathac":

```
agathaac
```

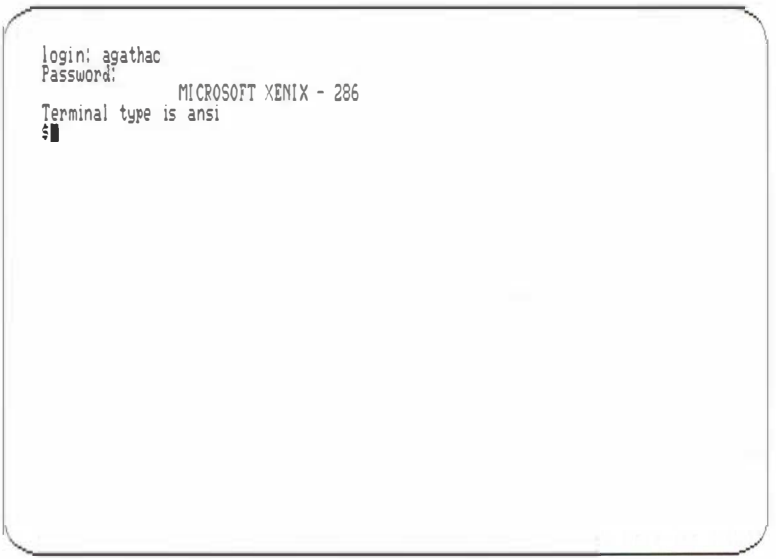
To correct this, you would press the BACKSPACE key until you place the cursor at the second "a":

```
agatha
```

All you have to do next is type the "c" and continue with what you were doing.

Assuming All Went Well

If you successfully completed Steps 1 – 3, your computer screen should look something like this:



There may be additional messages, depending on how your system is set up. The last line displayed on your screen will show the $ prompt, which means the computer is ready for you to give it a command.

As stated earlier, by logging in, you have opened a line of communication between the computer and you. From now until you break that line of communication by "logging out," you can use the computer to perform tasks for you.

If your screen shows the $ prompt, go on to the section "Commands." If not, read the following section "If It Didn't Work...."

If It Didn't Work...

If you followed the procedure and your screen displayed the following message, you may have made a typing mistake:

```
Login incorrect
login:
```

1. Try the login procedure again and be careful to type your login name and password correctly. Note that the computer does not allow you to correct typing mistakes when you type your password.
2. If you still receive the "Login incorrect" message after a couple more tries, check with your system manager to make sure everything is set up correctly and that you have all the correct login information (login name and password, if you need one).

Commands

Once you've logged in and have the $ prompt, XENIX is waiting for you to tell it what to do. In the next section, you'll learn a few simple "commands" that will give you some information from XENIX.

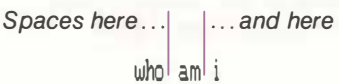
A command is what you type to tell the computer to do a task for you. All commands in this book are shown in **boldface** type.

Case Is Significant

XENIX makes a distinction between capital letters and small letters, so you need to be careful to use the right case when typing. One command you will be introduced to is the **date** command. You must type "date" all in lowercase letters. If you type "Date" using a capital "D," XENIX will not recognize the command.

Spaces Are Significant

When you type commands, it's important to leave spaces where they are shown and *not* to put spaces in where there should be none. Another command you'll soon learn is the **who am i** command. To use this command, you would type:



If you type it without spaces, it won't work.

Getting Information from the System

In this section, you will learn to use some XENIX commands that give you information.

date

You can find out the date and time whenever you want by using the **date** command.

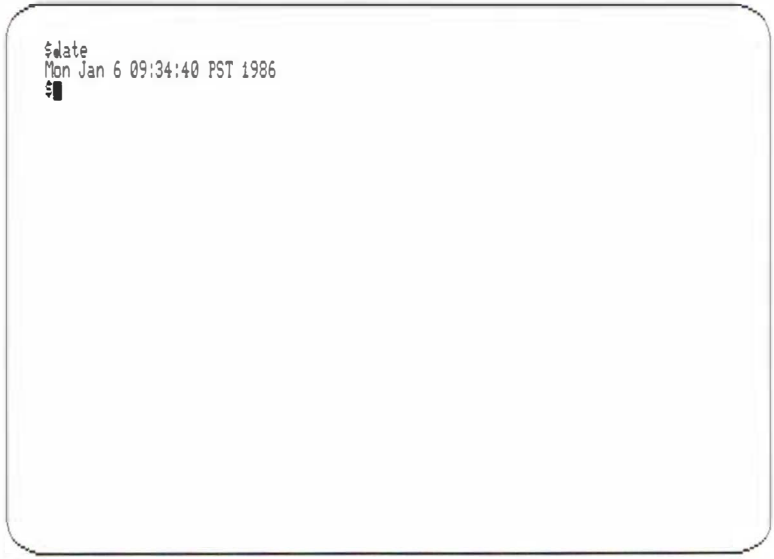
Try it out: display the date and time

1. When you see the $ prompt, type the following:

```
date
```

2. Press ENTER.

   Your screen will display something like this:



who

By using the **who** command, you can find out who is using the system at the same time you are.

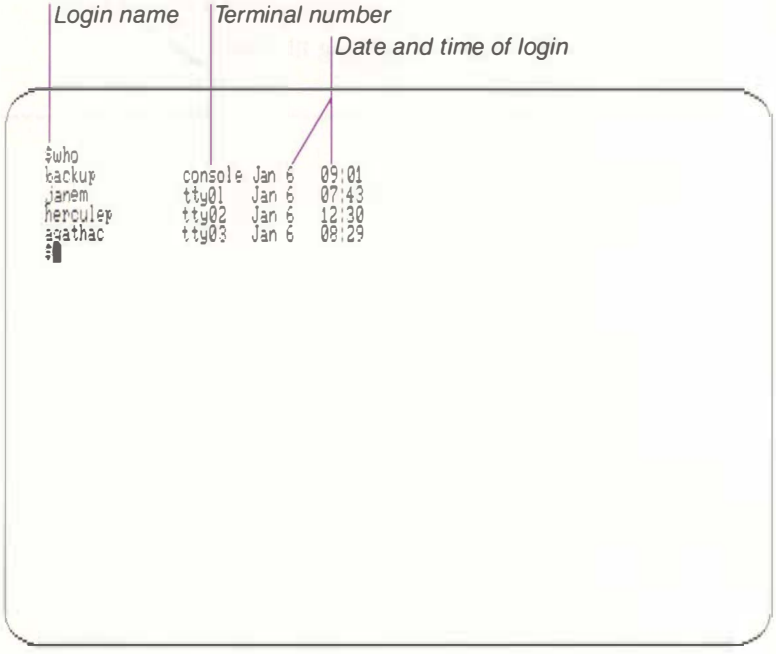
Try it out: find out who is using the system

1. When you see the $ prompt, type the following:

```
who
```

2. Press ENTER.

   Your screen will look something like the one that follows. The columns are labeled to show what information is displayed when you use the **who** command.



who am i

The **who am i** command is similar to the **who** command except that you receive information only about yourself. This could come in handy if you just want to quickly check the time you logged in, or if you use more than one login name at different times.

Try it out: check your login information

1 When you see the $ prompt, type the following:

Spaces here . . . *. . . and here*

```
who am i
```

2 Press ENTER.

Your computer will display something like this:

```
$who am i
agathac tty03 Jan 6 08:29
$
```

cal

The **cal** command prints a calendar on your screen. You can display a calendar for an entire year or specify a given month.

If you use the **cal** command followed by the year (a number from 1 to 9999), you will see the entire year. If you use **cal** followed by the month (a number from 1 to 12) and year, the specified month appears.

Try it out: display a calendar

In this example, assume you want to print the calendar for the month you were born. Remember to type the number "1", not the small letter "l".

1. At the $ prompt, type the following (substitute the correct month and year of your birth for "9 1890"):

   Spaces here . . . *. . . and here*

   ```
   cal 9 1890
   ```

2. Press ENTER.

   Your screen will display the calendar for your month of birth.

Note If you try to display the calendar for an entire year, the first few months will "scroll" off the top of your screen. To stop the scrolling, press and hold down the CTRL (CONTROL) key, then press the "s" key. To resume scrolling, press any key.

Creating or Changing Your Password

Your password is a word that identifies you to the computer when you log in. If you have a password, you cannot log into the computer without using it. Unlike your login name, your password should be known only by you: it prevents unauthorized people from using information you have stored in your computer.

If you were given a password by your system manager, you should change it, so you are the only one who knows what it is. Once you tell XENIX your password (using the following procedure), no one can log in using your login name unless they know your password.

If you didn't have a password the first time you logged in, you can create one for yourself by using the following procedure. From then on, you will be the only person who can log in using your login name.

Choosing a Password

When you choose your password, pick something easy for you to remember, but difficult for someone else to guess. Your password should be at least five characters long. It's a good idea to include letters, digits, and punctuation marks (for example, 0clues?!).

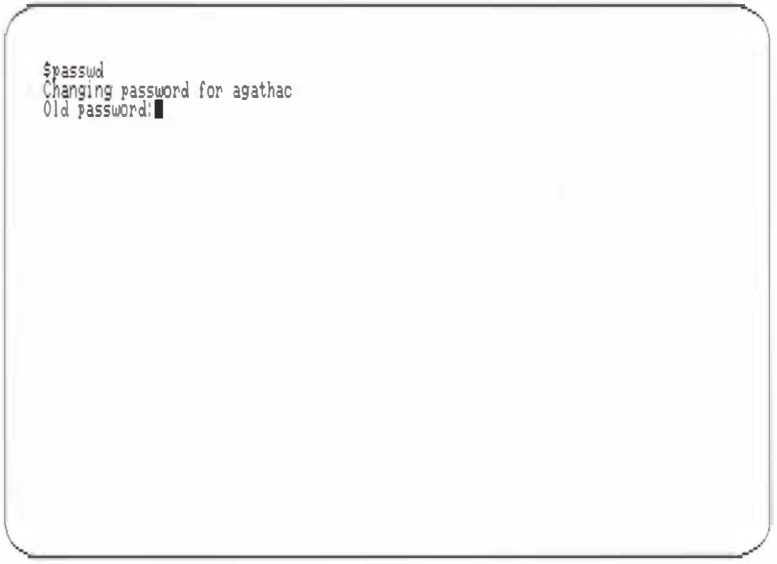
To change your password, follow these steps:

Try it out: change your password

1. Type the following and press ENTER:

```
passwd
```

Your screen will look something like this:



2. Type your old password and press ENTER. Your password is not displayed on the screen.

Your screen will look like this:

```
$passwd
Changing password for agathac
Old password:
Enter new password (minimum of 5 characters)
Please use a combination of upper and lowercase letters and numbers.
New password:
```

You can't correct typing mistakes when you type your password. If you make a mistake, just press ENTER and you will see the message "Sorry," then the $ prompt. Start again with Step 1.

3. Type your new password and press ENTER.

   The following message will appear:

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

4. Type your new password again and press ENTER.

   When you have succeeded in changing your password, you will see the $ prompt. The next time you start your computer, you will need to type the new password in response to the "Password" prompt.

If It Didn't Work...

- Did you press ENTER at the end of each step?
- If you received the following message, you used a new password with less than 5 characters:

```
Too short. Password unchanged.
$
```

  Try the password procedure again and be sure your new password has at least 5 characters.
- If you received the following message, you may have typed your old password incorrectly:

```
Sorry.
$
```

  Start again with Step 1 and use the correct password.
- If you received the following message, you may have accidentally typed your old password when you should have typed the new one:

```
Password unchanged.
$
```

  Start again with Step 1 and be sure to use a different word for your new password.
- If you received the following message, you probably made a mistake when you retyped your new password:

```
They don't match; try again.
New password:
```

  Begin again at Step 3.

If You Forget Your Password

You may want to write your password down and put it in some safe place, in case you forget what it is. If you ever find that you can neither remember your password nor find your copy, contact your system manager.

Quitting/Signing Off: Logging Out

As mentioned earlier, once you log in you can continue to use your computer until you "log out," which breaks your line of communication with the computer.

Try it out: log out

1 To log out, press the following:

CTRL-D

This means you press and hold down the CTRL key, then press the "d" key. (The key may be labeled CONTROL or CNTRL.)

2 You will hear a short beep, and your screen will show the "login:" prompt.

To continue with the exercises in this book, log in again. Logging in was described on page 13.

If It Didn't Work...

If instead of the "login:" prompt you see a "d" after the $ prompt, this means you pressed the "d" key *before* you pressed the CTRL key. Try again and be sure to press the CTRL key before you press "d".

When To Log Out

You should log out whenever you will be away from your computer for a significant amount of time (an hour or so). This prevents others from coming by and using your computer without your permission.

Note that logging out is not the same as turning off your computer with the on/off switch. Generally speaking, you should leave your computer running when you are not using it.

Review

In this chapter, you learned how to:

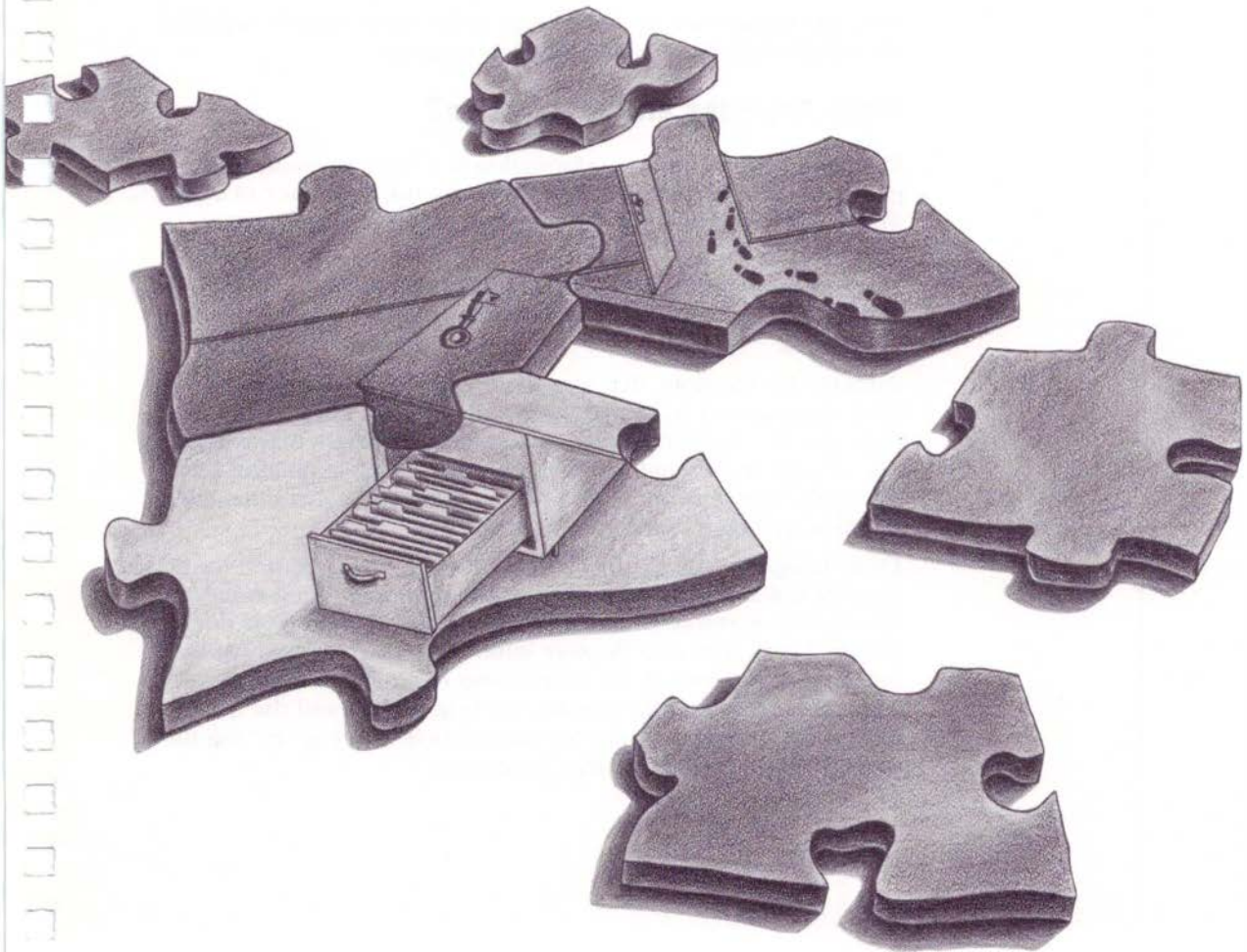
- Log in
- Use these commands to display information:

  date
  who
  who am i
  cal
- Change or create a password using the **passwd** command
- Log out

If you are unclear about logging in or logging out, review these sections before going on to the next chapter. For more information about any of the subjects discussed here, refer to the chapter called "Demonstration" in the XENIX *User's Guide*.

3 Storing Information— Files and Directories

One advantage of having a computer is that it can store a lot of information. In this chapter, you will learn how stored information is organized in the XENIX system and how you can find the information you want to use.



What You Will Learn in This Chapter

When you finish this chapter, you will know how to:

- Create a file or directory
- Obtain file and directory information from the system
- Move to another directory or file in the file system
- Make a copy of a file
- Move or rename a file or directory
- Delete a file or directory

How the System is Organized

Files and directories are the storage units XENIX uses to organize the information that is stored in the computer.

What Are Files and Directories?

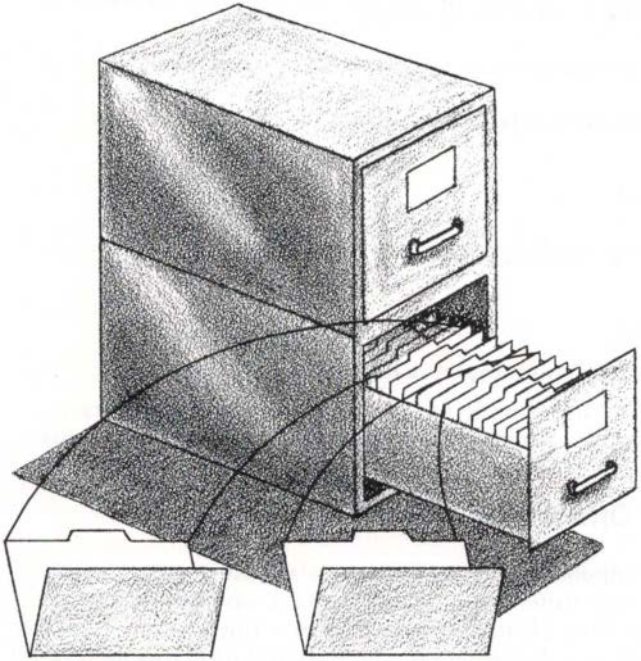
The basic unit for storing information is a "file." Someone has gathered the information and stored it on the computer in a file. This is somewhat like the way you might put some important papers into a file folder.

These computer files are, in turn, stored in a "directory." You could think of a directory as a file drawer where you store your file folders.

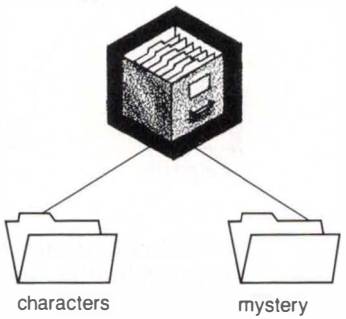
Suppose, for example, that you write mystery stories and you use your computer to write and store your manuscripts. You might have one file containing a list of possible characters that will be in one of your stories and another file containing a beginning draft of the story. You store these two files in a directory that contains all the files relating to your current story.

Even though we say the files are "stored in" the directory, the files are not actually physically "in" the directory. On a computer, information is stored in electronic "bits" and "bytes." This doesn't translate very well into the way information is stored on paper, but the organization of the information is similar to the way you might store files in a file cabinet. To help understand the XENIX file system, you can picture the information as being "in" the files, and the files as being "in" the directories.

A XENIX directory is like a file drawer: it contains files.



A XENIX file is like a file folder: it contains information.



On your computer, the characters file and the mystery file are stored in a directory.

Your XENIX system contains some sample files and directories that you will use along with this book to learn XENIX. Check with your system manager to be sure the sample file system is available to you.

If you wish, you can look at one of these files right now.

Try it out: look at a file

1 At the $ prompt, type the following:

```
more Space /demo/publishing/inprogress/characters
```

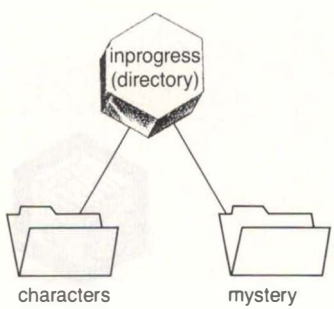
2 Press ENTER.

Your screen should display a list of character names.

You will see this sample file again later on, and you'll learn more about the **more** command you used to look at the file.

Files and Directories Have Names

Because a computer can store many files and directories, you must have a way to tell them apart. Each file and directory is given a name. You tell the computer these names when you want to locate your files. For example, the files and directory described earlier might be named as follows:

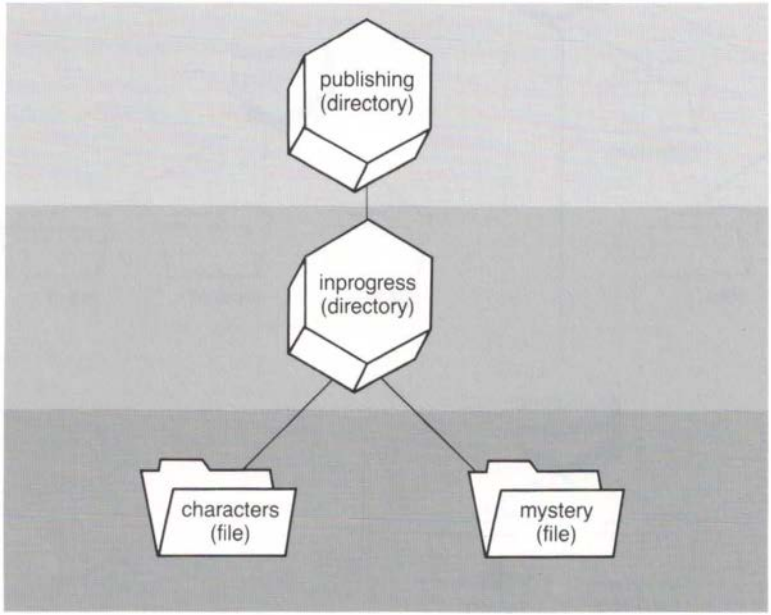


In this book, *italics* are used to show the names of files and directories when they are mentioned in the text.

Directories Are Multi-Level

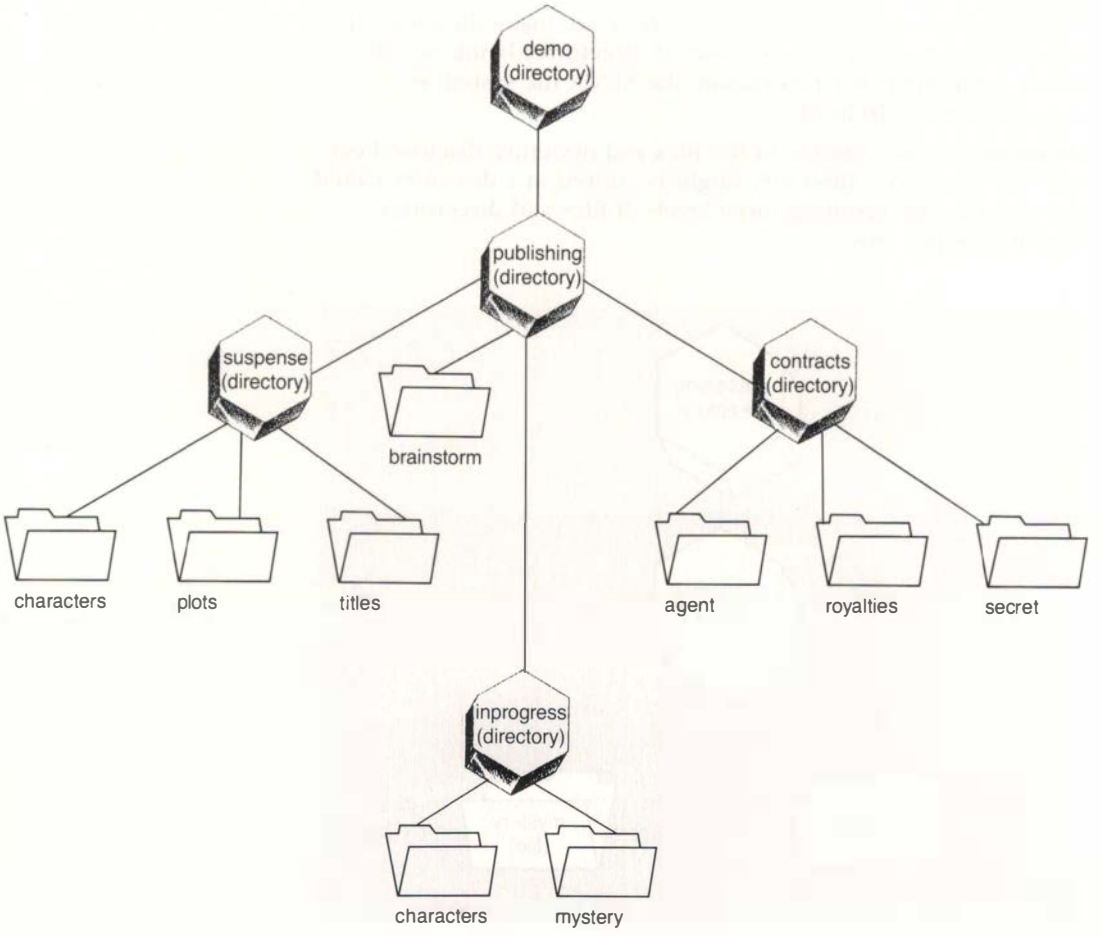
Besides containing files, XENIX directories can also contain other directories. These directories, in turn, can contain other directories, and so on. As you can see, if there are many files and directories, this results in several levels of directories being nested inside each other. For this reason, the XENIX file system is described as "multi-level."

Returning to the example of the files and directory discussed earlier, the *inprogress* directory might be stored in a directory called *publishing*. The resulting three levels of files and directories would look like this:



If we add even more files and directories to the file system, it might look like the diagram on the following page.

The file system shown here is the set of sample files and directories that you will be using with the exercises in this book.

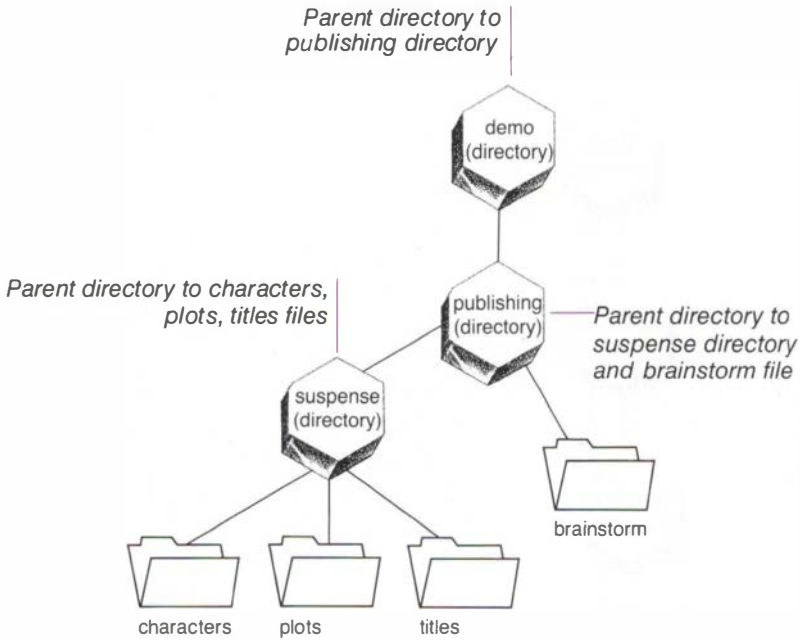


Remember that the XENIX file system is simply a way of organizing information. The preceding diagram of the file system is a kind of map to help you locate the files and directories in the file system. This "map" shows how the files or directories on one level are stored in a directory on the level above.

Parent Directories

The lines connecting the various files and directories show which files and directories are stored in a given directory. In each case, the directory on the upper level is called the "parent" directory of the files and directories that are stored in it.

The diagram that follows shows parent directories.



Finding a File: Using Pathnames

In the preceding diagram of the file system, each file or directory is connected to its parent directory by a line. This line means there is a "path" from one to the other. When you want to locate information stored in your computer, you need to tell XENIX exactly where the file is in the file system. You do this by typing the "pathname" of the file.

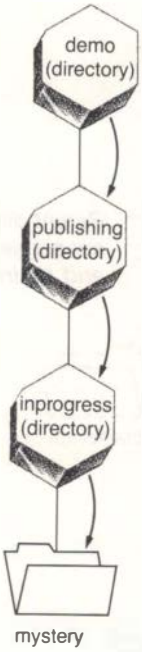
The pathname is a set of directions XENIX follows from one directory to the next until it reaches the file you requested. The pathname starts at the top of the file system and moves down to the requested file. For example, the pathname of the *mystery* file in the sample file system is shown as follows. Notice that the directories and files in the pathname are separated by a slash (/).

```
demo/publishing/inprogress/mystery
```

This pathname tells XENIX to locate the *mystery* file in your file system by looking in the *demo* directory, moving down to the

publishing directory, then down to the *inprogress* directory. XENIX finds the *mystery* file in the *inprogress* directory.

The following diagram shows the path XENIX follows to find the *mystery* file.



One thing is missing from this pathname. It will be added in the next section.

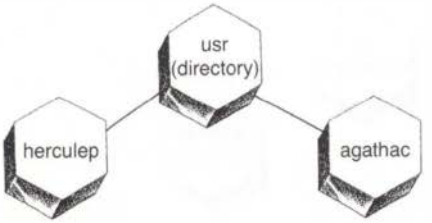
XENIX Directories You Should Know About

Within the XENIX system, there are some standard files and directories that are automatically included as part of your file system. For now, there are two standard XENIX directories that you need to know about.

Your Home Directory

Each XENIX user has his or her own personal directory, called the "home" directory. Your home directory, which usually has the same name as your login name, is where you generally store your own files.

In XENIX, the directory just above your home directory is named *usr*. Generally, when you log in, you are automatically placed in your home directory. The diagram below shows how you (disguised as "agathac") and another user fit into the XENIX file system.



(other user's home directory) (your home directory)

The Root Directory

The highest-level directory in the entire XENIX file system is the "root" directory. In a pathname, the root directory is represented by a slash (/), and since it is the highest-level directory, the slash is placed at the beginning of the pathname. This lets you distinguish the slash that indicates the root directory from the slashes used to separate directories and files in a pathname.

The root directory is just above the *usr* directory, so the pathname for Agatha's home directory is:

Notice the slash indicating the root directory

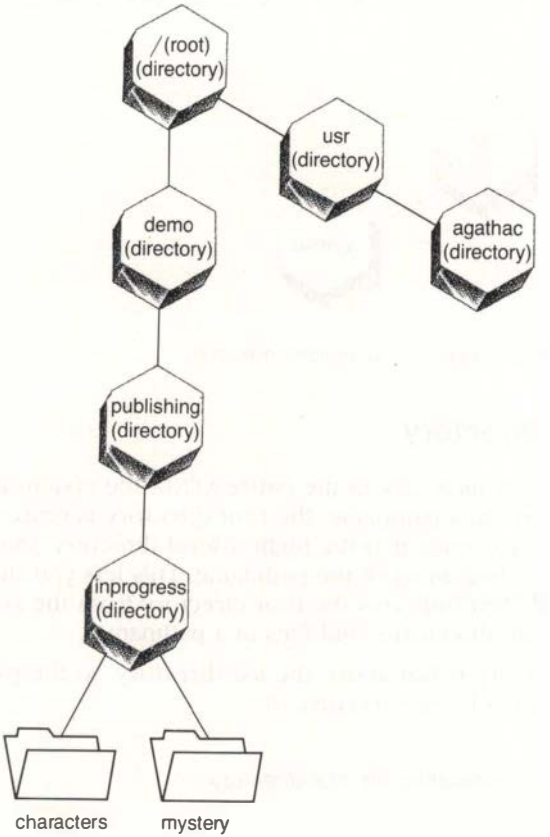
```
/usr/agathac
```

Remember that in the last section, there was something missing from the pathname for the *mystery* file. The missing part was the root (/) directory. The root directory is just above the *demo* directory, so the complete pathname for the *mystery* file looks like this:

Notice the slash indicating the root directory

```
/demo/publishing/inprogress/mystery
```

The following diagram shows the root directory at the top of the file system.



Being In the File System

When you log in to XENIX, you are placed "in" a certain directory in the file system (usually, your home directory). From there, you can "move" to the other directories that contain files you want to use. If the directory you're in already contains the files you want, you don't have to move; you have easy access to them.

Obviously, the only things really moving are your fingers and the keys. It's just more convenient to talk about the file system as if the directories and files were actual locations that you move to when you use files in the various directories.

Remember that the diagram of a file system is like a map. Picture yourself as being placed at some location on the map. From there, you follow the paths from your present location to the file or directory where you want to work.

Finding Out Where You Are: pwd

With so many directories in your file system, it is easy to forget which directory you are in. The **pwd** command, which stands for "print working directory," shows you the pathname of the directory you are currently working in. That directory, no matter which one it is, is always referred to as your "working directory."

For example, assume Agatha is in the *inprogress* directory and she uses the **pwd** command. The computer responds with the following pathname:

```
/demo/publishing/inprogress
```

This means Agatha is currently in the directory called *inprogress*, which is in the directory called *publishing*, which is in the directory called *demo*. Agatha's working directory is *inprogress*.

Try it out: find out which directory you're in

In this example, you will find out which directory you are in.

1. When you see the $ prompt, type the following:

   ```
   pwd
   ```

2. Press ENTER.

   Your computer screen will display the following:

   ```
   /usr/agathac
   ```

   In this example, you are in your home directory. (You will see your own home directory name in place of "agathac".) Your home directory is in the *usr* directory; the *usr* directory is in the root directory, which is shown by the slash (/).

If It Didn't Work...

If nothing happens after you type **pwd**, make sure you pressed the ENTER key.

Moving Things Around: Working with Files and Directories

There are many tasks you can perform with files and directories:

- You can create and delete them.
- You can move their contents to another location in the file structure.
- You can rename them.
- You can make copies of them.

The sections that follow show you how to do these things.

Making a Directory: mkdir

The **mkdir** command, which stands for "make directory," is the command you use to create a new directory. For instance, you may need a directory for storing the information you're researching for your mystery novel.

When you create a new directory, you are really creating a "subdirectory" of your working directory.

Try it out: create a directory

In this example, you should start in your home directory, where you will create a new directory called *research*.

1 If you need to move to your home directory to do this procedure, type the following and press ENTER. You will learn more about the **cd** command in one of the following sections.

```
cd
```

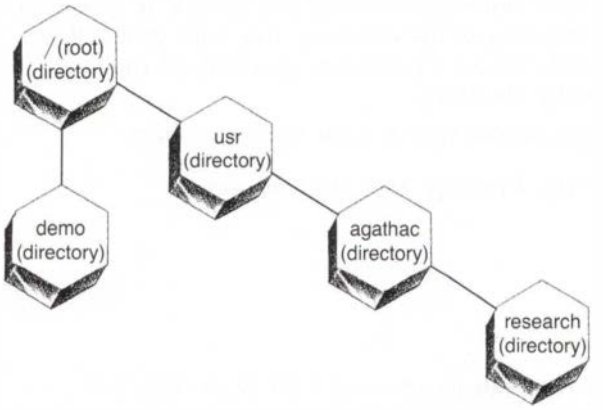
To create the new directory, type the following and press ENTER:

```
      Space
mkdir research
```

Your screen will display the $ prompt.

If you were to diagram your file system, it would now look something like the one that follows.

The right side of the diagram shows the new *research* directory in your home directory (*agathac*).



Right now, you can only assume that you created the *research* directory. In the next two sections, you will check that the *research* directory was created.

You may have noticed that so far you haven't put anything *into* the directory you created. XENIX doesn't mind empty directories. Once you create a directory, that directory will continue to exist until you delete it, even if you don't put any files or other directories into it.

If It Didn't Work...

- Try the procedure again and be sure to type the line exactly as shown, including the space.
- If you see a message saying "Cannot make directory research," the directory may already exist. If you see this message, go on to the next section.

Listing a Directory's Contents: lc

The **lc** command stands for "list contents in columns" and is an abbreviated way of listing what is stored in a directory. If you want to find out just the names of the files and directories in a directory, without any other information, you use the **lc** command followed by the pathname of the directory you want to list. If you use the **lc** command without a pathname, you will list the contents of your working directory.

Try it out: list contents of your working directory

In this example, you should start in your home directory.

1. When you see the $ prompt, type the following:

```
lc
```

2. Press ENTER.

   The computer responds by showing a list of the files and directories in your home directory.

   You should see the word "research," which shows that you created that directory.

Try it out: list directory contents using a pathname

Now, suppose you want to list the contents of the sample directory called *inprogress*. You are not in that directory, but you can list the contents anyway by typing **lc** followed by the directory's pathname.

Type the following:

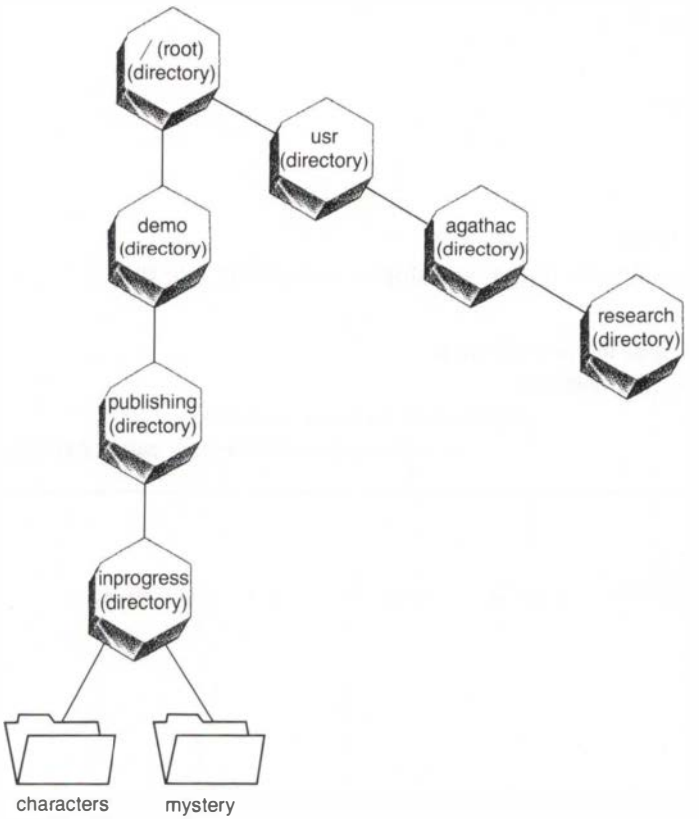
```
lc Space /demo/publishing/inprogress
```

2. Press ENTER.

   You should see the words "characters" and "mystery," which are the names of the files in the *inprogress* directory.

The **lc** command shows you only the *names* of files and directories in the directory. If you want more information, you need to use the **ls** command, which is described in the next section.

The following diagram shows the location of the *characters* and *mystery* files in the file system.



Listing a Directory's Contents: ls

The **ls** command stands for "list." This command, when used with **−l** (for "long"), gives you a more complete list of information about the contents of a directory. For example, if you are in your home directory and you want to find out not only what files and directories you have in it, but some additional information about the files and directories, you would use the **ls −l** command.

If you want to list the contents of a specific directory, you would use the **ls −l** command followed by the directory's pathname. The **−l** is called an "option." Options are sometimes added to a command to make the command work in a slightly different way.

Try it out: request a long list of a directory's contents

In this example, you should start in your home directory.

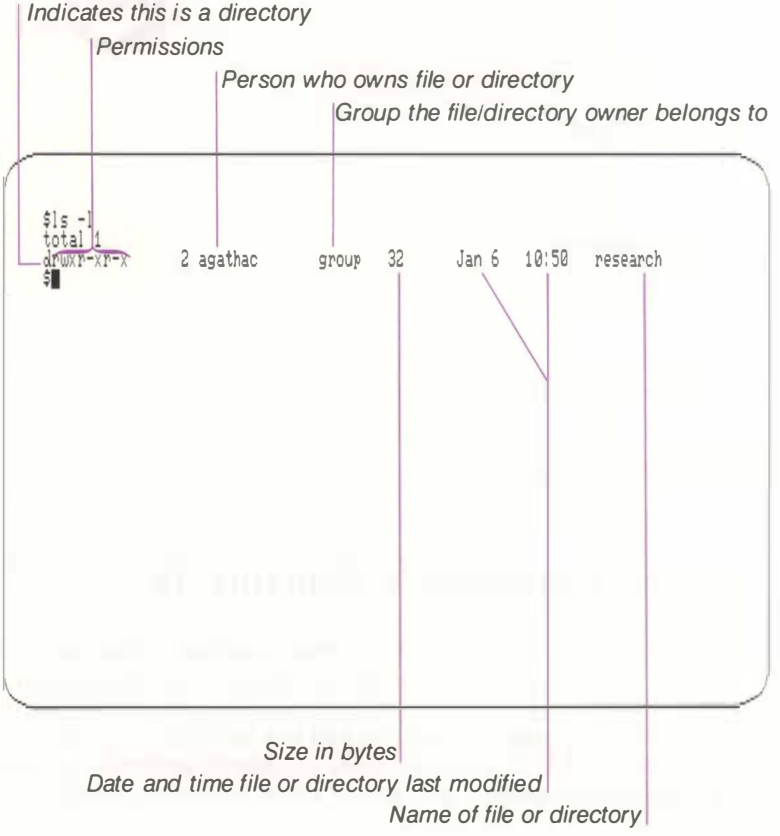
1. Type the following (be sure to type the letter "l", not the number "1"):

Space

```
ls -l
```

Press ENTER.

Your computer screen will display something like this:



As you can see, this command results in a lot of information being displayed on the screen. This information is explained thoroughly in the chapter called "Tasks" in the XENIX *User's Guide*, so it won't be duplicated here. Two things are important to note:

- You can see that the *research* directory you created is in your home directory. Note that a "d" in the first column indicates a directory. If there is no "d," the item listed is a file.
- The first column shows the "permissions" for the file or directory listed on that line. Permissions control which files and directories in the file system can be used by which people. In other words, permissions keep unauthorized people from making changes in your files and directories or even from reading them without your knowledge.

To learn more about permissions or any of the other information provided by the **ls −l** command, refer to the chapter called "Tasks" in the XENIX *User's Guide*.

Changing Directories: cd

The **cd** command lets you "change directories." Changing directories is a way of saying that you "move" from one directory to another in the file system. (Remember, you are automatically placed in a directory when you log in, usually your home directory.) To change directories, you use the **cd** command followed by the pathname of the directory to which you want to move.

Try it out: change directories

In this example, you will move from your home directory to the directory called *demo*. *Demo* is just below the root directory (/).

1. When you see the $ prompt, type the following and press ENTER:

   Space

   ```
   cd /demo
   ```

   This takes you back up to the root (/) directory, then down to the *demo* directory.

2. To see that you have moved, type the following and press ENTER:

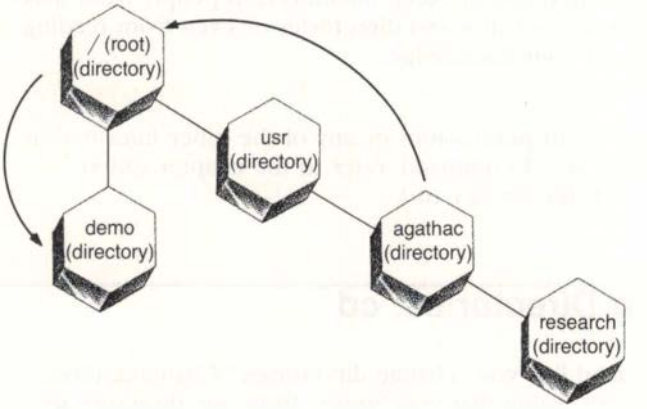
   ```
   pwd
   ```

You will see the following on your screen:

```
/demo
```

This shows that you are now in the *demo* directory.

The following diagram shows the path you follow to move from your home directory to the *demo* directory.



Using cd to Move to Your Home Directory

XENIX provides a quick way to return to your home directory: you use the **cd** command without a pathname. This will take you back to your home directory from anywhere in the file system.

Try it out: move to your home directory

1 At the $ prompt, type the following:

```
cd
```

2 Press ENTER.

If you wish, use the **pwd** command to check that you have returned to your home directory.

Copying a File: cp

At times, you may want to make changes to a file but keep the original file as it is. The **cp** (for "copy") command makes a copy of the file while leaving the original file unchanged.

To copy a file, you use the **cp** command followed by the pathname of the file you are copying, then the pathname of the copy.

This is how the command looks:

```
cp (pathname of original) (pathname of copy)
```

You can be anywhere in the file system and still copy a file this way. You don't have to be in the directory that contains the file.

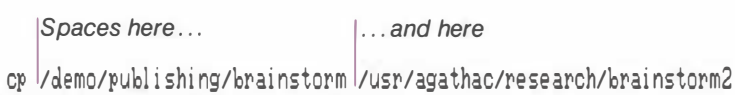
Note that the **cp** command simply makes a copy and leaves the original file untouched.

You can put the copy of the file in any directory. If you put it in the directory that contains the original file, you must give the copy a new name to tell it apart from the original. If you put the copy in a different directory, you can use the same filename, since the two files will have different pathnames.

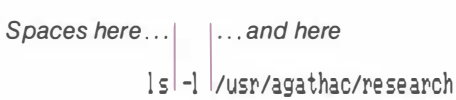
Try it out: make a copy of a file

In this example, you will put a copy of the *brainstorm* file into the *research* directory you created earlier. You will call the copy *brainstorm2*.

1. Type the following and press ENTER (remember to type your home directory name instead of "agathac"):

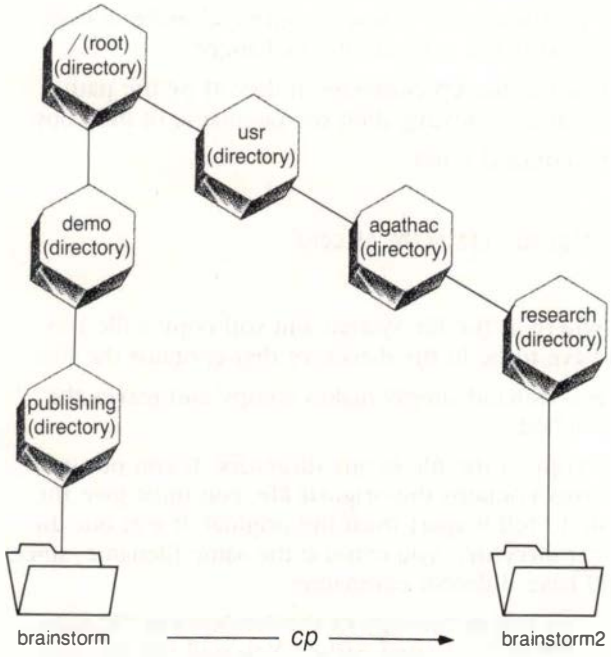


2. To see that the copy was made, type the following and press ENTER:



You should see the *brainstorm2* file listed.

The following diagram shows the copy of the *brainstorm2* file in the *research* directory.



Moving Files and Directories: mv

At times, you may decide that a certain file fits better in a different directory. The **mv** command *moves* a file or directory to another directory: it no longer exists in its original location.

To move a file or directory, use the **mv** command followed by the pathname of the file or directory you're moving, then the pathname of the destination file or directory.

This is how the command looks:

```
mv (pathname of original location) (pathname of destination)
```

Try it out: move a file to another directory

In this example, you will move the *brainstorm2* file from the *research* directory to your home directory.

1. Type the following and press ENTER (remember to type your home directory name instead of "agathac"):

```
mv /usr/agathac/research/brainstorm2 /usr/agathac/brainstorm2
```

Spaces here... *...and here*

2. To see that the file has moved, type the following and press ENTER:

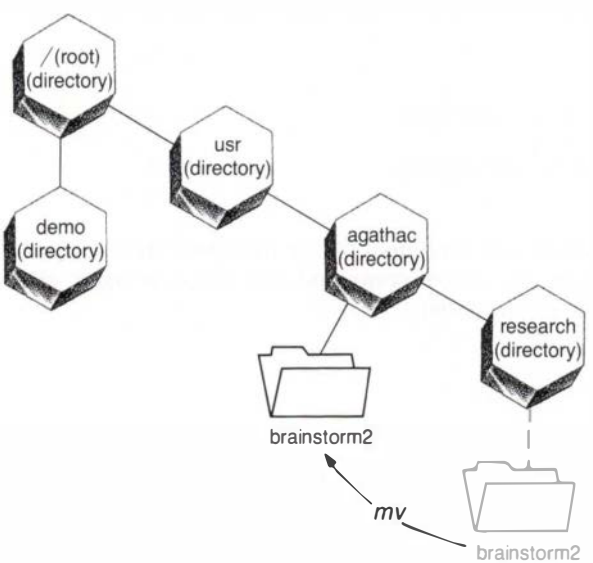
```
ls -l /usr/agathac
```

Spaces here... *...and here*

You can see that *brainstorm2* is now listed in your home directory.

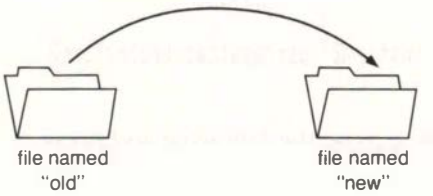
If you check the *research* directory, you will see that *brainstorm2* is no longer listed.

The following diagram shows the result of the **mv** command you just typed.



Renaming Files and Directories: mv

You can also use the **mv** command to *rename* a file or directory. In renaming, **mv** moves the contents of the old file or directory to a newly named one.



Try it out: rename a file

In this exercise, you will rename the *brainstorm2* file you just moved. The new name will be *outline*.

1. Type the following and press ENTER (remember to type your home directory name instead of "agathac"):

```
mv /usr/agathac/brainstorm2 /usr/agathac/outline
```

   Spaces here... *...and here*

   The file is now called *outline*.

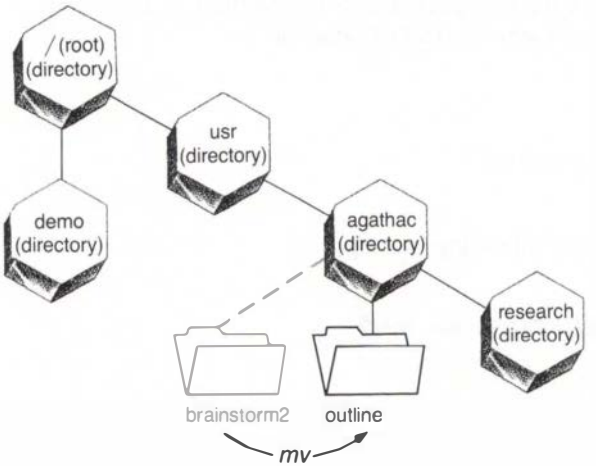
2. If you list the contents of your home directory again, you will see *outline* listed but not *brainstorm2*. Type the following and press ENTER:

```
ls -l /usr/agathac
```

   Spaces here... *... and here*

Note You can move and rename a file or directory in one operation. When you type the **move** command, use the new name as the destination file or directory.

The following diagram shows the *brainstorm2* file being renamed as *outline* using the **mv** command.



Removing a File: rm

You might want to remove a file because the information in it is outdated. The **rm** command deletes a file. Before using this command, be sure there is nothing in the file you want to keep: once you remove a file, you may not be able to get it back. (If necessary, check with your system manager about retrieving deleted files.)

To remove a file, you use the **rm** command followed by the pathname of the file you want to delete.

This is how the command looks:

```
rm (pathname of file)
```

Be sure you type the pathname correctly or you may accidentally delete the wrong file.

Try it out: delete a file

In this example, you will delete the *outline* file from your home directory.

1 Type the following and press ENTER (remember to type your home directory name instead of "agathac"):

```
rm Space /usr/agathac/outline
```

2 You may see the following message:

```
rm /usr/agathac/outline: 444 mode
```

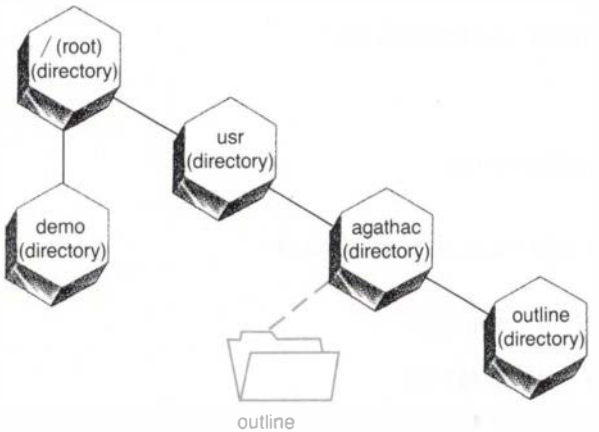
3 If so, to remove the file, type the following and press ENTER:

```
y
```

The preceding message relates to permissions, which control who can use and delete which files and directories. The original *brainstorm* file that you copied has its permissions set to prevent anyone from deleting the file. Your copy, *brainstorm2* (now *outline*), takes on the same restrictions. In the case of your copied file, you can override this by pressing "y", as shown above. For more information on permissions, see the chapter called "Tasks" in the XENIX *User's Guide*.

If you list the contents of your home directory, you will see that the *outline* file has disappeared from the listing.

The following diagram shows that the *outline* file has been deleted with the **rm** command.



rm

Removing a Directory: rmdir

At times, you will probably want to get rid of a directory. For example, you may have completed your research on a project and have moved or deleted all the research files. Now, you want to clean up your file system and remove the empty directory.

To remove a directory, use the **rmdir** command. First, you must be sure it is empty, since a directory cannot be removed if it still contains any files or other directories. To check the contents of the directory, use the **ls −l** command. If it is not empty, move or delete the files or directories still in it. If it is empty, you can remove it with the following command:

```
rmdir (pathname of directory)
```

Try it out: delete a directory

In this example, you will remove the *research* directory from your home directory.

Type the following and press ENTER (substitute your home directory name for "agathac"):

Spaces here... *...and here*

```
ls -l /usr/agathac/research
```

If the message "total 0" is displayed, the directory is empty and you can go ahead and remove it. If not, you will need to remove any files. Refer to the preceding section, "Removing a File," on page 49.

2 Type the following and press ENTER:

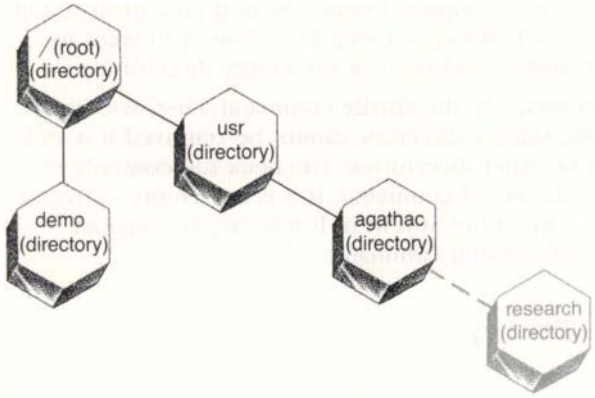
```
        Space
rmdir  /usr/agathac/research
```

3 Now, type the following and press ENTER:

```
Spaces here...   ...and here
          ls -l  /usr/agathac
```

As you can see by the listing, the *research* directory has disappeared.

The following diagram shows that the *research* directory has been deleted with the **rmdir** command.



rmdir

If It Didn't Work...

If *research* is still listed, follow the procedure again. Be sure to type the command in Step 2 exactly as shown, including the space after the **rmdir** command.

Shortcuts: Using Relative Pathnames

It may have occurred to you that while typing the commands in the examples, you are duplicating a lot of information in the pathnames. There is a shortcut. XENIX has two types of pathnames: "full" and "relative."

Full Pathnames

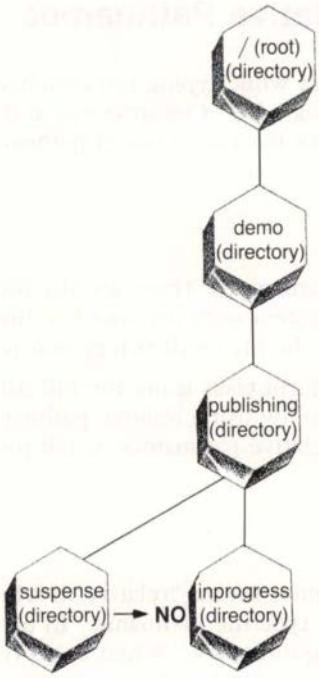
So far, we have used only full pathnames. These are the pathnames that start at the top of the file system with the root (/) directory and move down step-by-step to the file or directory you want.

Now that you've learned to perform tasks using the full pathnames, you can save a lot of time by using relative pathnames. In fact, once you learn to use relative pathnames, you'll probably rarely use full pathnames.

Relative Pathnames

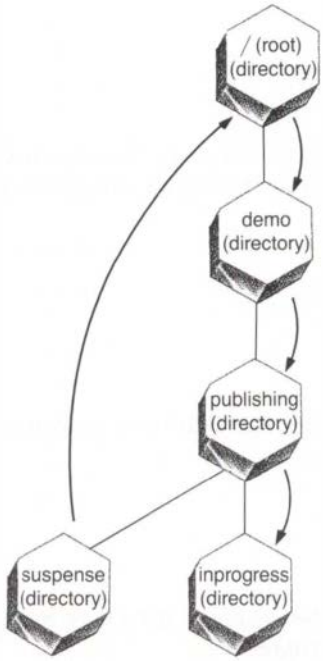
The "relative" in relative pathnames means "relative to where you are in the file system when you type the pathname." In other words, "relative to your working directory." When using relative pathnames, you start from your current location in the file system, then follow the shortest path to the file or directory you want.

In the following diagram, assume you are in the *suspense* directory and you want to move to the *inprogress* directory. *Suspense* and *inprogress* are both in the *publishing* directory. Since no lines connect the *suspense* and *inprogress* directories, no path exists between them and you can't move directly from one to the other.



You can't go straight from suspense to inprogress.

To move from one to the other, you have two choices. You could use the full pathname (cd /demo/publishing/inprogress), as you did in the section on changing directories. This takes you all the way up to the root (/) directory, then down the file system to the *inprogress* directory, as you can see in the following diagram.



Or there is an easier way. Since you are in the *suspense* directory and since both directories are connected to the *publishing* directory, you can use the relative pathname and simply move up the structure to *publishing*, then down to *inprogress*.

You start by moving to the directory just above your current location. A shorthand name for the parent directory of any directory you are in is two periods (..). So, to move to the parent directory of a file or directory, you use the **cd** command followed by the shorthand name (..).

Then, to move down to the directory you want, you add a slash (/), then the name of the directory you're moving to. Remember that when slashes are used in the middle of a pathname, they separate the files and directories.

Try it out: change directories using relative pathnames

In this example, you will use the shortcut to move from the *suspense* directory to the *inprogress* directory.

1 You need to start in the *suspense* directory. To move there, type the following and press ENTER:

```
cd Space /demo/publishing/suspense
```

2 Now that you are in the *suspense* directory, you can use the relative pathname to move to the *inprogress* directory. Type the following and press ENTER:

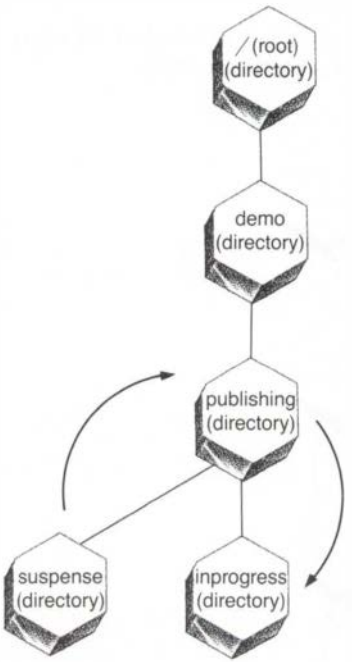
```
cd Space ../inprogress
```

3 Use the **pwd** command to verify that you have moved. You will see this pathname:

```
/demo/publishing/inprogress
```

The relative pathname from the *suspense* directory to the *inprogress* directory is "../inprogress".

The following diagram shows what happens when you change directories using the relative pathname.



Moving Up in the File System

To move up more than one level in the file system, use the shorthand name (..) for each level.

In this example, you will move from the *inprogress* directory to the *demo* directory.

Try it out: move up using relative pathnames

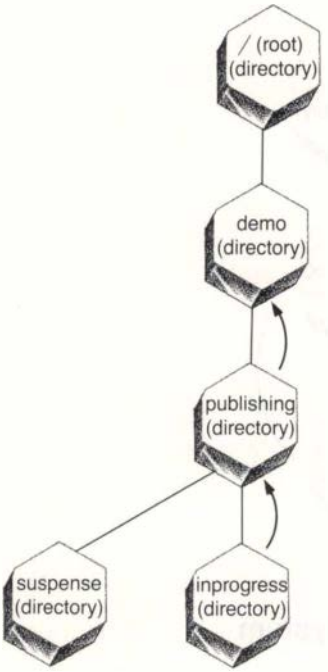
1 After the last exercise, you should be in the *inprogress* directory. To move up two levels to the *demo* directory, type the following and press ENTER:

```
cd Space ../..
```

2 Use the **pwd** command to see that you have moved to the *demo* directory.

The relative pathname of the directory two levels up in the file system is "../..". From there, you can move to any directory just below it by adding a slash (/) and the directory's name. To go up more levels, add "/.." for each level.

The following diagram shows the path you follow to move from the *inprogress* directory to the *demo* directory.



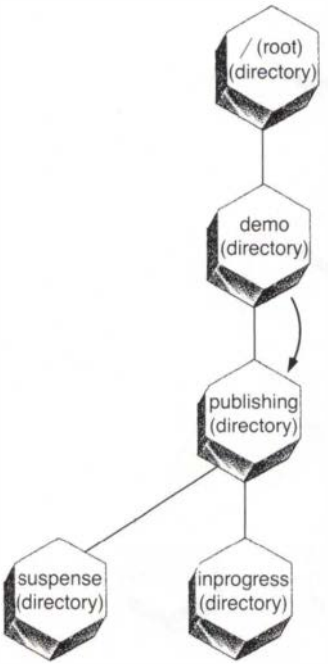
If It Didn't Work...

If you are in the *publishing* directory instead of the *demo* directory, you probably put a space after the first "..". Move back to the *inprogress* directory by typing "cd inprogress". Press ENTER. Then, try the exercise again and leave a space only after **cd**, as shown.

Moving Down in the File System

If you want to move down one level in the file system, you use the **cd** command followed by the name of the next lower directory. Note that you do not type a slash (/) in front of the directory name when you move to the directory just below you.

For example, when you are in the *demo* directory, typing "cd publishing" moves you down one level to the *publishing* directory.



If you want to go down more than one level, type the pathname starting with the first directory you will move down to (again, no beginning slash). Note that this pathname is *relative* to the directory you are in, so it is the relative pathname.

Try it out: move down using relative pathnames

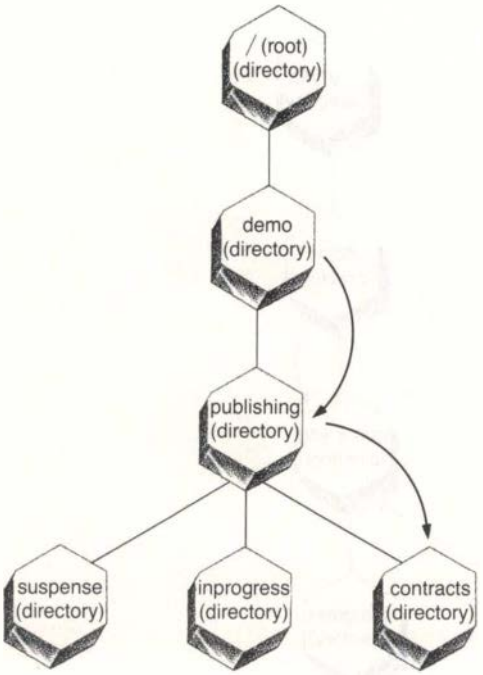
After the last exercise, you should be in the *demo* directory. In this example, you will move from the *demo* directory to the *contracts* directory.

1 Type the following:

```
cd Space publishing/contracts
```

2 Press ENTER.

The following diagram shows the path you follow to move from the *demo* directory to the *contracts* directory.



The Value of Relative Pathnames

Now that you have learned to use relative pathnames, you can save yourself a lot of typing when you are using XENIX. You can substitute the relative pathname for the full pathname in any XENIX command. This eliminates typing full pathnames, which at times can be long and complex.

Review

You have covered a lot of information in this chapter. Don't be concerned if you can't perform these tasks from memory: you now know where to find out how to do them.

In this chapter, you have learned how to:

- Find out where you are in the file system: **pwd**
- Create a directory: **mkdir**
- Display a short listing of a directory's contents: **lc**
- Display a detailed listing of a directory's contents: **ls −l**
- Move from one directory to another: **cd**
- Move back to your home directory: **cd**
- Copy a file: **cp**
- Move or rename a file or directory: **mv**
- Remove (delete) a file: **rm**
- Remove (delete) a directory: **rmdir**
- Use relative pathnames in commands

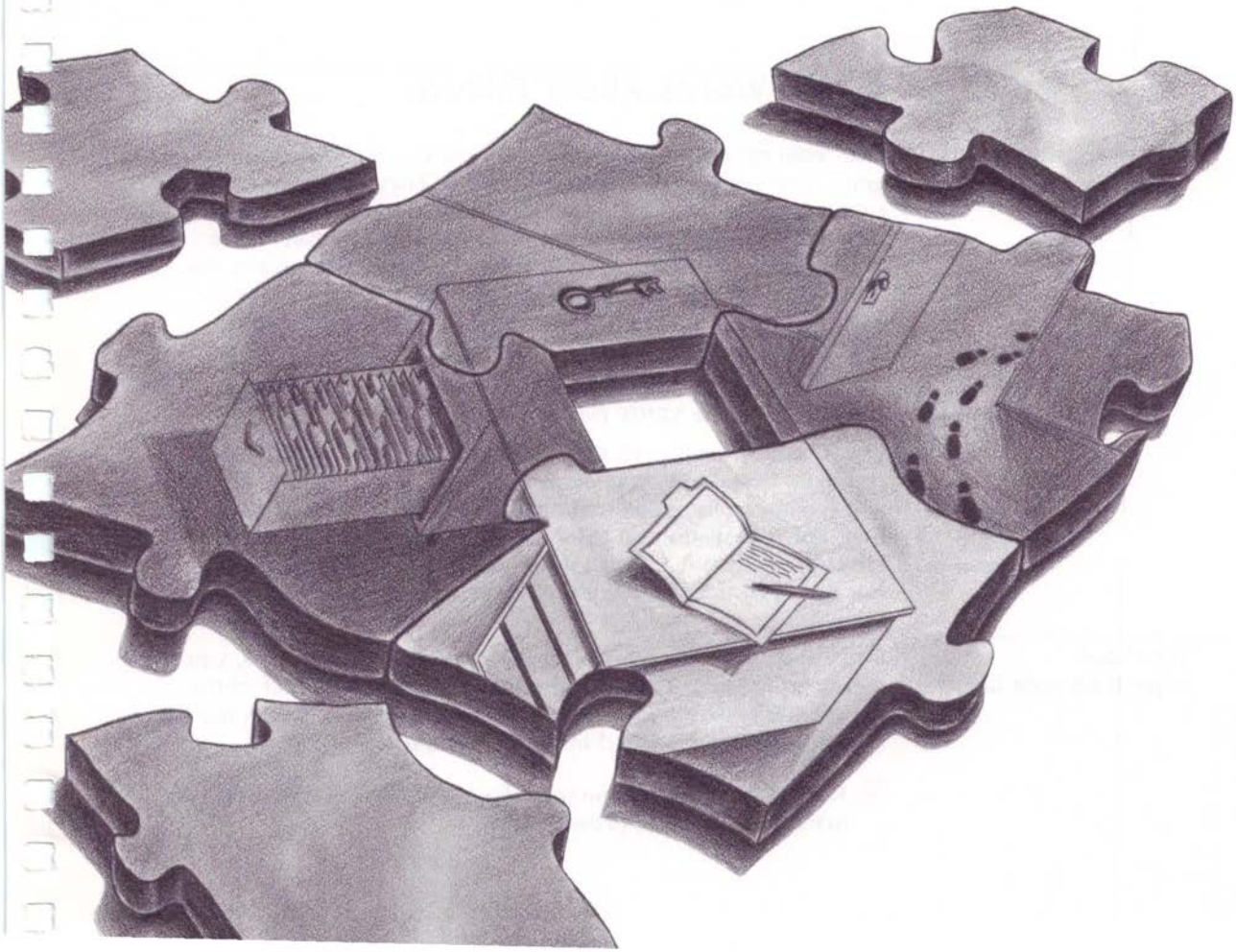
You also learned the following important concepts:

- Being in the file/directory system
- Directories are multi-level
- What a parent directory is
- What your home directory is
- What pathnames are
- What the root directory is
- How the slash (/) symbol is used
- Moving up in the file system
- Moving down in the file system

In the next chapter, you will learn how to create a new file and how to make changes to the text of an existing file.

4 Creating and Changing Files With the Vi Editor

So far you have learned about moving files around, renaming them, deleting them, and so forth. You haven't yet learned how to move *into* a file and make changes to the text, or how to create a new file and put some information into it. These tasks are the subject of this chapter.



What You Will Learn in This Chapter

When you finish this chapter, you will know how to:

- Use the Vi editor to create a new file
- Use the Vi editor to open a file
- Change to "command mode" in the Vi editor
- Change to "insert mode" in the Vi editor
- Move the cursor around in a file
- Put new text into a file
- Correct errors in a file
- Delete text from a file
- Save the changes you've made and store the revised file on your computer

A Few Words About Editors

The "editors" discussed here are not people, but computer programs that are part of the XENIX system. They are more properly called "text editors," but generally are referred to simply as editors. An editor program is a set of commands you can use to create and make changes to the information in a computer file. A word processing program is an example of a text editor.

Two editors are included with XENIX: "Ed," which won't be covered in this book, and "Vi" (for "visual"), which is the one you'll learn to use. (For information on Ed, refer to the appendix called "Ed" in the XENIX *User's Guide*.)

The Vi editor, or just **vi**, as it is usually called, is a very powerful program. Eventually, you can learn to perform complex search and replace tasks, move text around, and insert text from other files. For information on these and other **vi** features, see the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*. But first, you need to learn the basics, which is the purpose of this chapter.

Try it out: copy a sample file

To prepare for your work with the next several examples, you will need to make a copy of the *mystery* file that is part of the sample file system included in your XENIX system. This is the file system that was discussed in the last chapter.

1. Start in your home directory. To move there, use the **cd** command (type **cd** and press ENTER).

2. Type the following and press ENTER (remember to type your home directory name instead of "agathac"):

```
cp /demo/publishing/inprogress/mystery mystery2
```

Spaces here... *...and here*

3. List the contents of your home directory using the **ls −l** command to see that the *mystery2* file is there.
4. Type the following and press ENTER:

```
chmod u+w mystery2
```

Spaces here... *...and here*

This gives you permission to make changes in the file. For more information on permissions, refer to the chapter called "Tasks" in the XENIX *User's Guide*.

If It Didn't Work...

If you don't see the *mystery2* file listed, check with your system manager to be sure your system contains the sample files and directories in the *demo* directory.

Opening an Existing File or Creating a New File

Before you can change the text in any of your files, you have to "open" the file. This simply means that you use the Vi editor program to bring the file to your screen so you can make changes to the information in it.

You use this same procedure to create a new file. Instead of bringing an existing file to the screen, **vi** creates a new, empty file and you type information in it.

In this section, you'll learn one way to open an existing file (or create a new file). If you want to learn about other ways to do this, refer to the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*.

If You Are Already in the Right Directory

Opening (or creating) a file using **vi** is simple when you are already in the directory that contains the file (or in the directory where you want to put the new file). You just use the **vi** command followed by the filename of the file you want.

Try it out: open a file

In this example, you will open the file named *mystery2* that you just copied into your home directory.

1. Start in your home directory. To move there, use the **cd** command.
2. Type the following and press ENTER:

```
   Space
vi mystery2
```

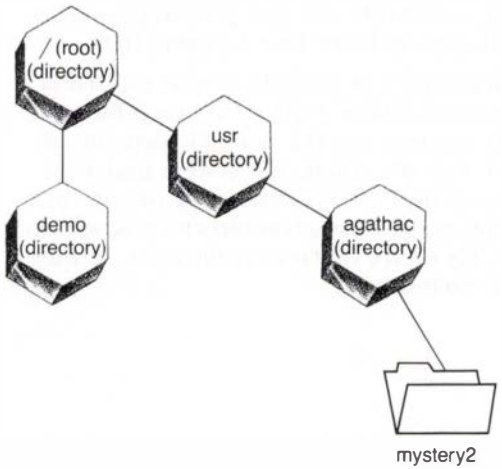
If your screen looks like the one that follows, you have successfully opened the file named *mystery2*, and the computer is waiting for you to work on the file. Notice the name of your file at the bottom of your screen followed by the number of lines and characters.

```
                    WORK IN PROGRESS
                  (The Epiphany Affair)

     It was a dark and stormy night.  Miss Jane Marple sat by the hearth
and its cozy firer, oblivious to the racket the wind made at the shutters. It
was so good to be home again in one's favorite chair after a few weeks away.
A long string of of holidys, actually.  Christmas, Boxing Day, New Years,
and at last, Twelfth Night -- Epiphany.  Her nephew, Raymond, had
been insistent taht she stay with him in his new flat in London for
the season.

     Miss Marple knew this was Raymond's tactful way of suggesting she
must surely be bored with village life by now.  Young people had so
little understanding of the intrigue one found staying in one small
place and watching carefully.  Subtleties, nuances....  Oh well, London
had been a change of pace -- not exactly a breath of fresh air, but the
bustle of the city had been stimulating, and Raymond had been an
attentive and gracious host.  His new flat was modern and spacious,
"mystery2" 42 lines, 2675 characters
```

The following diagram shows the location of the *mystery2* file in the file system.



If It Didn't Work...

If you followed the procedure and did not see the file on your screen, here are a few things to check:

- Did you do the four preparation steps on pages 64-65? If not, do them now.
- Did you type the command lines *exactly* as shown? Note the spaces and slashes.
- Did you press ENTER after typing the command?

If You Are Not in the Right Directory

If you are not in the directory that contains your file (or the directory where you want to put the new file), you have two choices: either move to the right directory using the **cd** command, or use the **vi** command followed by the pathname of the file.

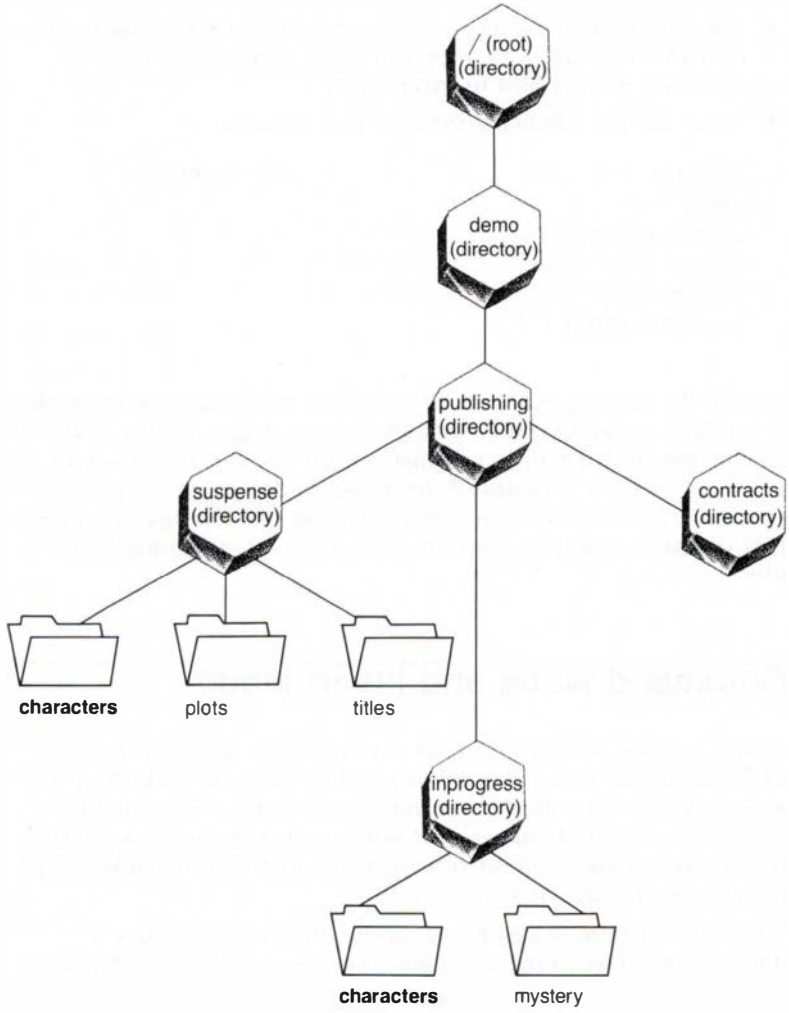
You will learn how to work in your file shortly, but first a few words about naming files.

Naming Files

In this book, you will work only with existing files, but once you start using XENIX for your own work, you will probably be creating new files and you will need to know how to name them.

Filenames should be as descriptive as possible so you can tell at a glance what the file contains. Within a directory, each filename must be unique, although you may use the same filename in different directories. (Notice that the sample file system under the *demo* directory includes two files named *characters*, one in the *suspense* directory and one in the *inprogress* directory. Since each is in a different directory, they have different pathnames, so they will not be confused with each other.)

The following diagram shows the location of the two *characters* files in the file system.



When naming your files, use the following guidelines:

- Use 1-14 characters.
- Do not use spaces.
- Use only letters, numbers, and the period (.). (Although other characters are allowed, they can cause problems if used incorrectly, so it is best to avoid them.)
- *Never* use the following symbols in a filename:

  slash (/)
  dash (−)
  question mark (?)
  asterisk (*)
  brackets ([ ])
  quotation marks (" ")

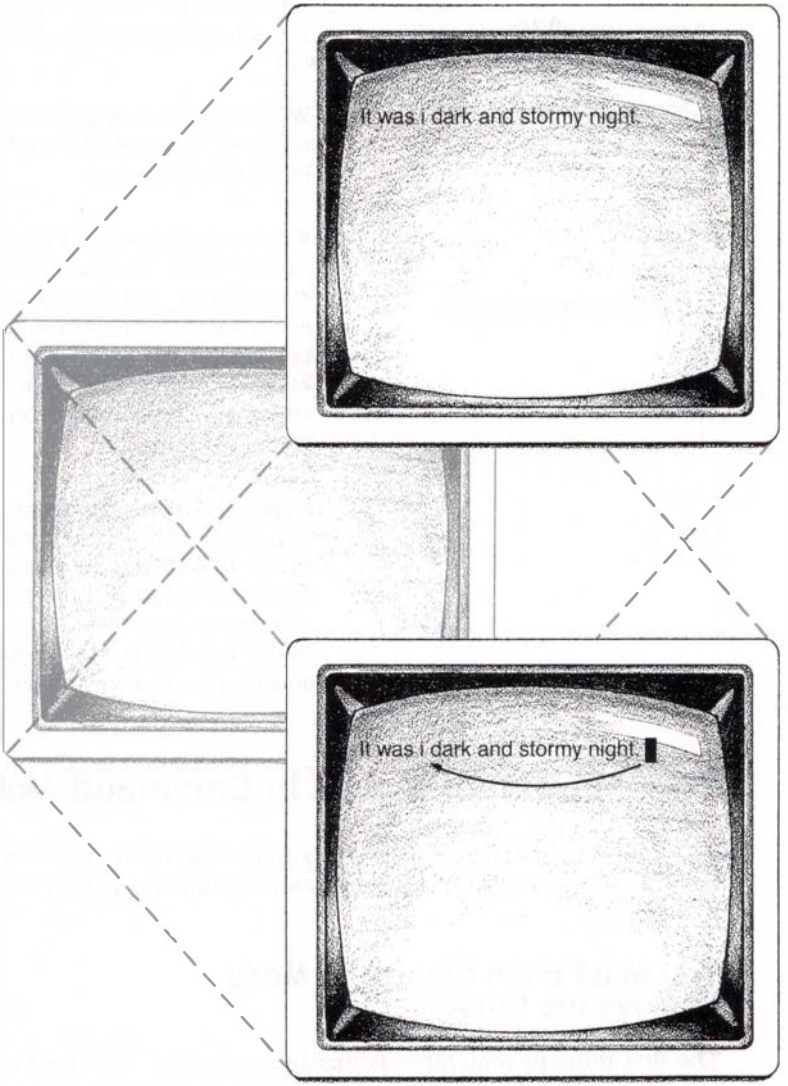
Careful file naming can help you organize your files. For example, if you have several versions of a letter, you might call the first version *proposal.1*. Then, as you make revisions, you could simply call later versions *proposal.2*, *proposal.3*, and so on. As long as you don't go over fourteen characters, you can use any combination of letters, numbers, and the period (.) when naming your files.

Command Mode and Insert Mode

Now, to return to work on your *mystery2* file. It's important to understand that when you are using **vi** to work on a file, you will always be in either the "command mode" or the "insert mode." Being in insert mode means that when you type on the keyboard, the characters show up on your screen, much like when you type on a piece of paper in a typewriter.

When you change to command mode, the same keys have a different meaning to the computer. In command mode, the characters you type do not appear on the screen. Instead, pressing certain keys "commands" certain things to happen, like making the cursor move on the screen, deleting words from the screen, and so forth.

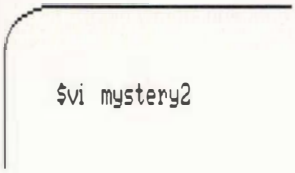
In insert mode, the letters you type show up on your screen.



In command mode, the keys you press make things happen (the cursor moves, words are deleted, etc.).

Changing Between Modes

Two keys on the keyboard allow you to change between modes. (There are others you can learn about later, but to keep this simple, you'll learn only two right now.)



You are in command mode.

When you first open a file using **vi**, you are automatically in command mode.

When you press the **i** (for "insert") key, you turn off command mode and change to insert mode. (After that, the **i** key works just like the other keys.) In insert mode, the characters you type appear on your screen.

When you press the ESCAPE (ESC) key, you turn off insert mode and change to command mode. In command mode, you can move the cursor around, move and delete words, lines, and blocks of text, and so forth.

How the Cursor Moves in Command Mode

To make changes to your text or to add or delete text, you will need to know how to move the cursor to different places in the file.

You Must Be in Command Mode to Move the Cursor

The first thing to remember is that in most cases, you need to be in command mode to move the cursor. (Exceptions: in insert mode, you can use the SPACEBAR and BACKSPACE key to move the cursor on the current line.)

You should probably make a habit of always pressing the ESCAPE key just before you move the cursor. If you're in insert mode, this changes you to command mode. If you're already in command mode, you won't hurt anything by pressing the ESCAPE key.

Some Simple Ways to Move the Cursor

You can move the cursor in four directions: up, down, right, and left. The chart that follows shows you a few simple ways to move the cursor. When you want to learn more, refer to the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*.

If you want to practice moving the cursor in the *mystery2* file you opened in the last exercise, press ESCAPE and use the instructions in the following chart.

Note If you try to move the cursor when the CAPS LOCK feature is on, your file will behave strangely. If the cursor does not move as expected when you're using **vi**, be sure CAPS LOCK is turned off.

Moving the Cursor in Vi

Press ESCAPE to enter command mode.

To Move Left ←	**Press**	
One character at a time:	Left arrow	
One word at a time: (punctuation is considered a word)	**b**	
To beginning of line:	**0**	

To Move Right →	**Press**	
One character at a time:	Right arrow	
One word at a time: (punctuation is considered a word)	**w**	
To end of line:	**$**	

To Move Up ↑	Press	
One line at a time:	Up arrow	
To the first line in the file:	**1** (one) capital **G**	

To Move Down ↓	Press	
One line at a time:	Down arrow	
To the last line in the file:	Capital **G**	

Inserting Text in a File

In a previous section, you opened a file named *mystery2* using **vi**. Since you have just opened the file, you are in command mode. Now, suppose you want to add a paragraph to the file. To put text into the file, use the following procedure.

Try it out: add text to a file

1. To verify you are in command mode, press ESCAPE until you hear the bell or beep. In **vi**, the bell means you are in command mode.
2. Press the following until your cursor is on the blank line above the second paragraph (the one that begins "Miss Marple knew..."):

3. Press the following once:

i

Now you are in insert mode and can begin entering text to your file.

4 Type the sample paragraph shown below. At the end of each line, press ENTER to move the cursor to the next line. (If the "wrap" feature is set up, you won't need to press ENTER at the end of each line. The cursor will automatically move to the next line. Check with your system manager to set up the wrap feature.)

If you make a typing mistake, read the next section before continuing to type.

```
        "Aunt Jane, you're becoming quite the stay-at-home," he'd persisted.
"Really, you haven't been away from St. Mary Mead since your trip to the
Caribbean, and that was three years ago.  Wouldn't a change of scene do
you good?"
```

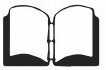
If You Make a Typing Mistake

If you notice your mistake before the cursor moves to the next line, you can correct it by using the BACKSPACE key to back up over what you have typed. The characters don't disappear, but you can type over them to correct your error.

If your mistake is at the beginning of the line and you don't discover it until the end of the line, or if you have already moved to the next line, just ignore it for now. You'll learn an easier way to correct errors soon.

Deleting and Correcting Text

There are several ways you can delete text and correct errors in your files. In this section, you'll learn a few of them. For more information, refer to the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*.



Deleting a Character

You may have noticed a few errors in the first paragraph of the *mystery2* file. For example, on the second line, the word "fire" has an extra "r" at the end. To delete a character from your text, you simply change to command mode, move the cursor to the appropriate place, and then press **x** (for "cross out").

Try it out: delete a character

In this example, you will delete the last "r" in "firer."

1. To begin this exercise, be sure you are still in the *mystery2* file (if necessary, see "Opening an Existing File or Creating a New File" on page 65).
2. Press ESCAPE to change to command mode.
3. If the cursor is below the error, press the following until the cursor moves up to the line you want:

4. If the cursor is above the error, press the following until the cursor moves down to the line you want:

5. To move the cursor forward word by word, press the following:

```
w
```

(The **w** stands for "word.") To move it back, press the following:

```
b
```

(The **b** stands for "back.")

6. To move the cursor letter-by-letter to the last "r" in "firer," press the following:

If your cursor is a box, it should be directly on the letter you want. If it is a line, it should be directly beneath the letter.

(If you go past the "r", just press the following until you move back to it):

7 Press the following once:

```
x
```

The unwanted "r" disappears and the space closes up.

If It Didn't Work...

If the cursor did not move as expected or if characters were inserted in your file, check these possibilities:

- Did you press ESCAPE before starting to move the cursor?
- Is the CAPS LOCK feature turned on? If so, the cursor will behave strangely. Turn off CAPS LOCK and try again.

Replacing One Character at a Time

Another error in the *mystery2* file occurs on the next to last line of the first paragraph. The word "taht" should be "that." Since only two letters are incorrect, it is easiest just to replace those letters one by one. To replace a character, you change to command mode, move the cursor to the appropriate place, press **r** (for "replace"), then type the new character.

Try it out: replace a character

1 Press ESCAPE to be sure you are in command mode.

2 If the cursor is below the error, press the following until the cursor moves up to the line you want:

3 If the cursor is above the error, press the following until the cursor moves down to the line you want:

4 Press the following until you reach the word "taht":

```
w
```

To move the cursor back, press the following:

```
b
```

5 To move the cursor letter-by-letter to the "a" in "taht," press the following:

If you go past the "a", press the following until you move back to it:

6 Press the following once:

```
r
```

This allows you to retype a letter without leaving command mode. The cursor will not advance as it does when you are in insert mode.

7 Type the letter "h" and the correction is made. You are still in command mode.

8 To correct the next letter, move the cursor to the right.

9 Press the following once:

```
r
```

10 Type the letter "a" and the correction is made.

Inserting a Character

At times, you may want to insert a character in your text. For example, in the fourth line of the *mystery2* file, the word "holidys" needs an "a" in it. To insert a character, change to command mode, move to the place you want to insert the character, press **i**, and type the character.

Try it out: add a character

In this example, you should still be in the *mystery2* file.

1. Press ESCAPE to be sure you are in command mode.
2. Move to the "y" in "holidys."
3. Press the following once to change to insert mode:

```
i
```

4. Type the letter "a". Note that the text to the right moves over to make room for the inserted letter. Remember that you are now in insert mode until you press ESCAPE, so anything you type will be inserted into your file.

Deleting or Correcting One Word

In many cases, you will want to delete a word or replace one word with another. To delete a word, change to command mode, move to the beginning of the word, press **d**, then **w** (for "delete word").

Try it out: delete and replace a word

In this example, you should still be in the *mystery2* file. Assume you have decided that instead of the words "few weeks" on the third line in the first paragraph, you want the word "holiday."

1. Press ESCAPE to be sure you are in command mode.
2. Move the cursor to the line you want.
3. Press the following to move the cursor to the beginning of "few," the first word you want to delete:

```
w
```

(If you need to move the cursor back, press **b**).

4. Press the following:

```
dw
```

The word "few" disappears.

5 To delete the next word, "weeks," move to the first letter of the word and press the following:

```
dw
```

6 Now press the following to change to insert mode so you can type the word you want:

```
i
```

7 Press the SPACEBAR to move the cursor to the right. Type the word "holiday". Press the SPACEBAR to separate it from the next word.

Notice that the text to the right of your new word moves over as you type. If you continued typing, it would continue moving over, so you could type as many words as you want.

8 Press ESCAPE to return to command mode.

9 You might have noticed that there is an extra "of" just in front of the word "holidays" on the fourth line. For more practice deleting a word, delete one "of." Move to the word and press the following:

```
dw
```

Deleting One Line at a Time

There may be times when you want to delete one or more lines of text from a file. To delete a line, change to command mode, move to the line you want to delete, then press **d** (for "delete") twice.

Try it out: delete one line

In this example, you will delete the last line of the *mystery2* file.

1 Press ESCAPE to change to command mode.

2 Press the following to move the cursor to the end of the file:

```
G
```

Capital **G** stands for "go to the end of the file." Note that pressing lowercase **g** does not work.

3 Press the following and the line disappears:

```
dd
```

Suppose you decide you didn't want to delete the line after all. In the next section, you'll learn how to recover it.

If You Change Your Mind: u

With **vi**, you can always "undo" your last command by changing to command mode and pressing **u**. You can undo your most recent command, whether you deleted just one letter or all the words in a file. But you must still be in **vi** and in the command mode.

It is important to emphasize that **vi** can only undo the *most recent* change you made to the file. In other words, if you delete a paragraph and change to insert mode and type some new text, then realize you didn't want to delete the paragraph, the **u** command won't bring your paragraph back. Instead, it will delete the text you just typed, since that was your last change to the file.

Try it out: undo your last command

In the previous exercise, you deleted the last line of the *mystery2* file. Assume that you have just finished doing this and you decide you really didn't want to delete this last line after all.

1 Press ESCAPE to be sure you are in command mode.

2 Press the following:

```
u
```

The line you deleted in the last section will reappear.

If It Didn't Work...

If you followed these steps and the line didn't reappear, you probably did something else to the file after deleting the line. If so, when you pressed **u**, you "undid" your last step. Try deleting a line (press **d** twice) and then immediately pressing **u**.

Other Ways to Change to Insert Mode

So far, we have told you about only one way to change from command mode to insert mode: pressing **i**. You will find that to efficiently add text to your files, you need to know some other methods. The following list shows you four of the most common ones.

If you press any of these keys while in command mode, you will move the cursor to the place shown and change to insert mode automatically. You then can begin adding text without pressing **i**.

If you want to	Press
Move right one space	a (lowercase)
Move to end of line	A (uppercase)
Create an empty line above	O (uppercase)
Create an empty line below	o (lowercase)

For more information on moving the cursor or changing to insert mode, refer to the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*.

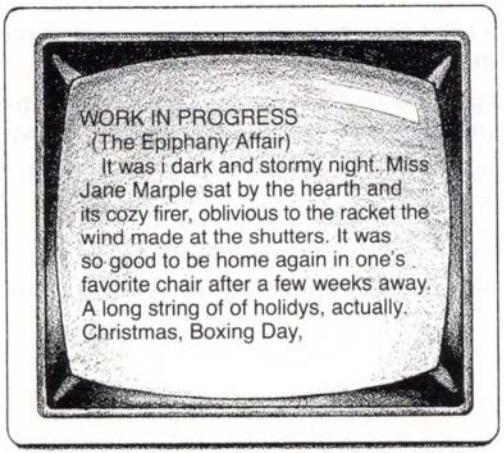
Saving Changes You Make in a File

Whenever you open a file and make changes to the text, your changes are not permanent until they are "written" or "saved" in the computer. This is because the file you see on your screen is a *copy* of the file stored in the computer. So if you want your changes to be in the file the next time you use it, you need to save them.

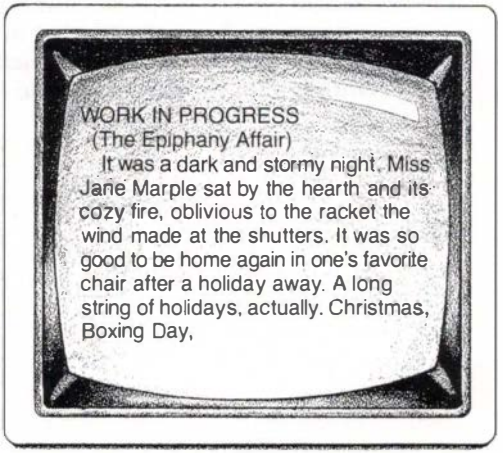
Generally, you should save any changes you have made before you leave a file.

When saving your revised file, you have two choices: you can save the changes and stay in the file, or you can save the changes and leave the file.

Using Vi brings a copy of the file to your screen.



You make changes to the copy, then you save your changes by pressing :w or :x. The next time you use the file, your changes are there, since they are now in the computer's memory.



Save Your Changes and Stay in the File: w

To reassure yourself that your changes are being saved, you may want to develop a habit of periodically saving your changes while you are working in a file.

To save your changes and continue working in the file, change to command mode, press the colon (:), then press **w** (for "write").

Try it out: save changes while you are in a file

In this example, you will use your *mystery2* file since you have made several changes to it.

1. Press ESCAPE to change to command mode.
2. Press the following (use the SHIFT key to type the colon):

```
:
```

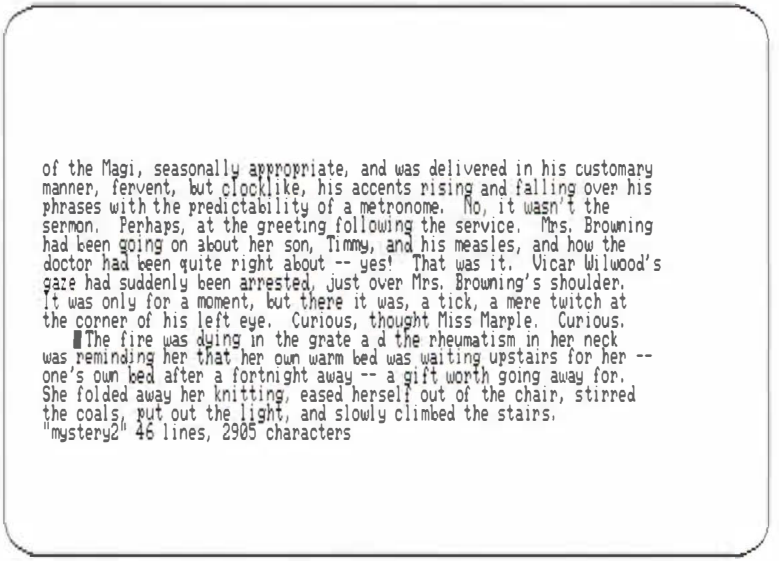
   Note that the cursor moves to the lower left corner of your screen.

3. Type the following and press ENTER:

```
w
```

   You will see the name of your file in the lower left corner of your screen. When the file has been saved, you will see information about the number of lines and characters. The cursor returns to its location in the file before you pressed the colon.

Any changes you have made to the file are now stored in the computer. Your screen will look something like this:



When the cursor returns to its original location on the screen, you can continue editing your file.

Save Your Changes and Leave the File: x

To save your changes and leave the file when you have finished editing, change to command mode, press the colon (:), then press **x** (for "exit"). Your changes will be saved, and you will leave the file and return to the $ prompt.

Try it out: save changes and leave the file

Assume that you have finished making changes to the *mystery2* file and you want to leave the file to work on some other task.

1. Press ESCAPE to change to command mode.
2. Press the following (use the SHIFT key to type the colon):

```
:
```

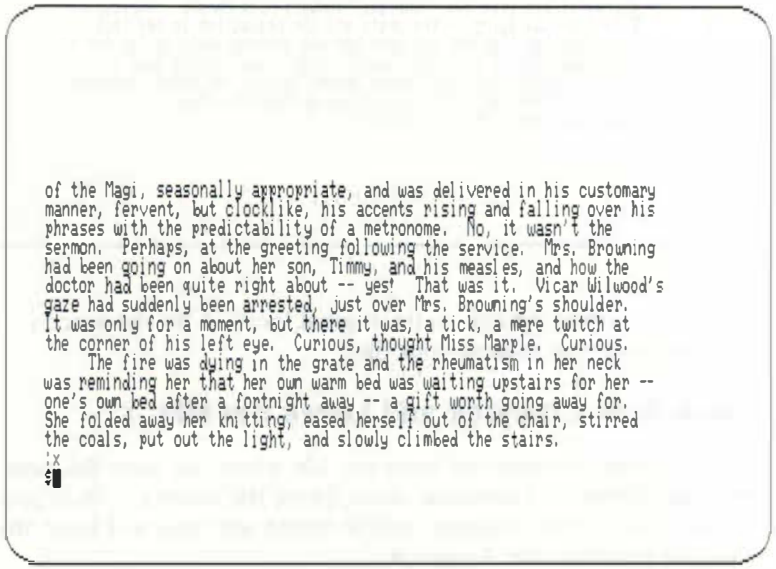
Note that the cursor moves to the lower left corner of your screen.

3 Type the following and press ENTER:

```
X
```

If you recently made changes to the file, you will see information about the number of lines and characters in the file. Then the **$** prompt appears, showing you have exited from the file and from **vi**.

If you have made no changes since the last time you saved your file, the screen will simply display the **$** prompt. Your screen will look something like this:



Whether you use **w** or **x**, any changes you have made to the file are now stored in the computer. The next time you look at the file, it will include the changes.

Recovering Changes if the System Fails

It is possible that while you are working on a file in **vi**, your system will fail due to a power outage or some other problem. If this happens, your screen will probably go blank. You should check with your system manager to find out how to restart the system.

Periodically, XENIX will automatically save a copy of your file in the computer. Normally, if the system fails while you are editing a file, all but your most recent changes will be saved automatically. Once the system is working again, you will probably receive a message telling you your file was saved.

To "recover" your file, you use the **vi −r** command followed by the filename. For example, to recover the *mystery2* file, you would type "vi −r mystery2". Be sure to type the space before "−r".

Other Ways of Leaving the File

You have just learned how to save your changes and leave the file. There are other ways to leave the file and return to the $ prompt so you can use other XENIX commands. In this section, we will discuss two additional methods of leaving your file.

Leave the File Without Saving Your Changes: q!

You may decide for some reason that you don't want to save any of the changes you have made. For example, you may have made several changes, then realized that you deleted some vital information you should have kept. The easiest solution would be to quit your editing session without saving the changes you made by using the **q!** command.

Try it out: leave a file without saving changes

In this example, you will leave your *mystery2* file without saving any changes.

To practice leaving a file without saving your changes, return to your *mystery2* file by typing the following and pressing ENTER (remember to type your home directory name instead of "agathac"):

```
vi Space /usr/agathac/mystery2
```

2. Press ESCAPE to change to command mode.
3. Press the following (use the SHIFT key to type the colon):

```
:
```

   The cursor moves to the lower left corner of the screen.
4. Type the following and press ENTER (use the SHIFT key to type the exclamation point):

```
q!
```

   You will see the $ prompt, showing that you are out of the file and out of **vi**. The **q!** command does not update the file stored in the computer, so the file remains the same as the last time you saved it.

Leave the File Temporarily: !

This is a very handy feature of XENIX. Using this procedure, you can temporarily leave the file you are editing, perform another task (such as printing a file, checking the time, or reading your messages), and instantly return to your file.

To leave the file, change to command mode and press **:!**.

Try it out: leave a file to use a command

In this example, you will leave your *mystery2* file temporarily.

1. To practice this command, return to your *mystery2* file by typing the following and pressing ENTER (type your home directory name in place of "agathac"):

```
   Space
vi /usr/agathac/mystery2
```

2. Press ESCAPE to make sure you are in command mode.
3. Press the following:

Type ! followed by the command you want to use, for example, **date**. What you have typed will look like this:

```
:!date
```

4 Press ENTER. The following message may appear:

```
[No write since last change]
```

This means you haven't saved your file since you made some changes.

It is usually a good idea to save your file with the **w** command before using the ! command, just in case something happens to the system while you are using your other command. For now, just ignore the message.

The command you typed will run in its normal way until it is finished. Since you used the **date** command in this example, the date and time will appear.

5 When the command is finished, you will see the following message:

```
[Hit return to continue]
```

Press ENTER and you will be back in your file, ready to continue editing. The file you were editing has not been affected by this procedure.

Review

In this chapter, you have learned how to:

- Use **vi** to create a new file
- Use **vi** to open a file
- Change to command mode in **vi**
- Change to insert mode in **vi**
- Move the cursor around in a file
- Put new text into a file
- Correct errors in a file
- Delete text from a file
- Save the changes you've made and store them on your computer
- Leave a file without saving your changes
- Leave **vi** temporarily to run another command

You also learned the following important concepts:

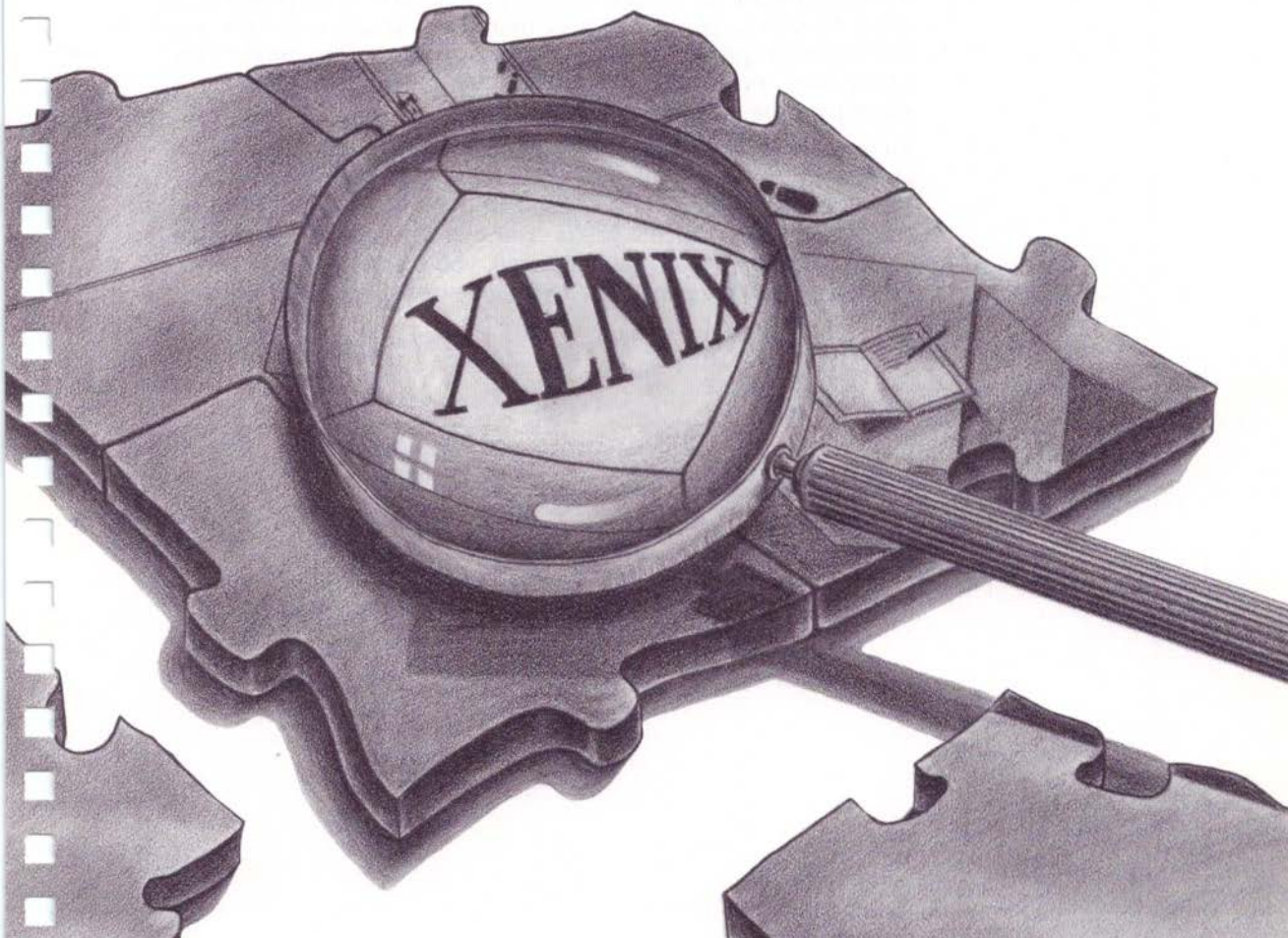
- What a text editor is
- Being in command mode
- Being in insert mode

Once you become familiar with the tasks presented in this book, you may want to refer to the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*. There you will learn some advanced features of the Vi editor, as well as other ways to perform some of the tasks you've already learned.

In the next chapter of this book, you will learn some other useful XENIX features.

5 Other XENIX Features

In this chapter, you will learn about some additional XENIX features that you will probably use often as you use and print files, communicate with other users on your system, run special programs, and perform multiple tasks.



What You Will Learn in This Chapter

When you finish this chapter, you will know how to:

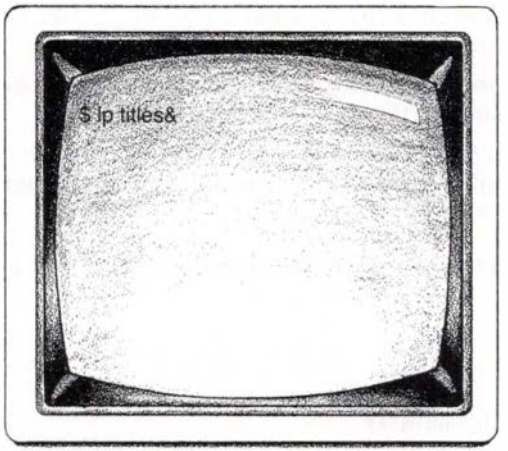
- Perform several tasks at the same time
- Print a file
- Use XENIX's built-in calculator
- Look at a file using the **more** command
- Look at a file using the **cat** command
- Send, read, edit, and save a message using **mail**

Putting a Process in the Background

XENIX can work on more than one process at a time. However, in order for you to perform two or more tasks at one time, you must tell XENIX to "put a process in the background." This allows you to use your screen to do something else while XENIX works silently on the assigned task.

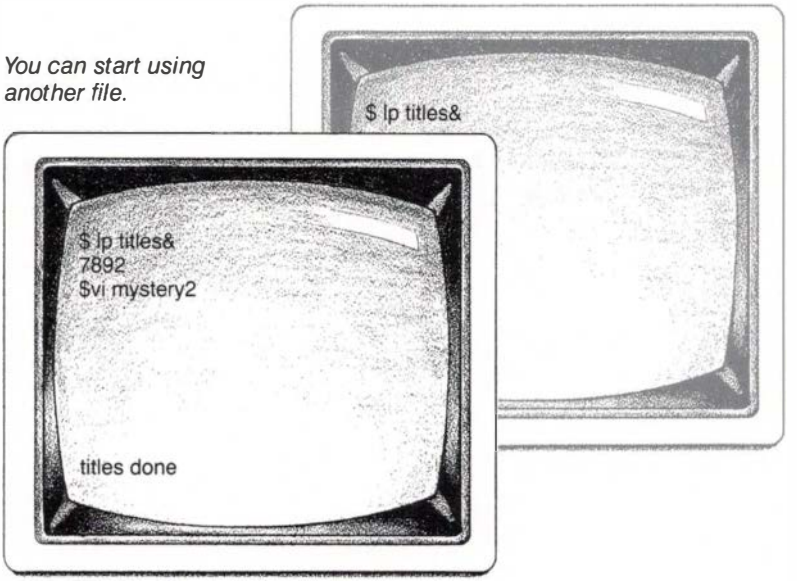
To put a process in the background, you type an ampersand (&) at the end of the command. For example, you might put the & after a print command. Then, you can perform another task on the computer while your file prints. You will discover that if you forget to type the & after the command, you cannot use your computer for anything else until XENIX finishes the specified task.

You type the print command, then the & sign, then press ENTER.



The titles file continues to print.

You can start using another file.



You may receive a message telling you that your file has finished printing.

Printing a File

In this section, it is assumed that you have a printer hooked up to your computer. If you don't, you can skip this section.

To print a file, you use the **lp** (for "lineprinter") command followed by the pathname of the file you want to print.

Try it out: print a file

In this example, you will print the file called *titles*, which is included in the sample file system you used in earlier chapters. You should start in your home directory.

1. Move to the *suspense* directory. Type the following and press ENTER:

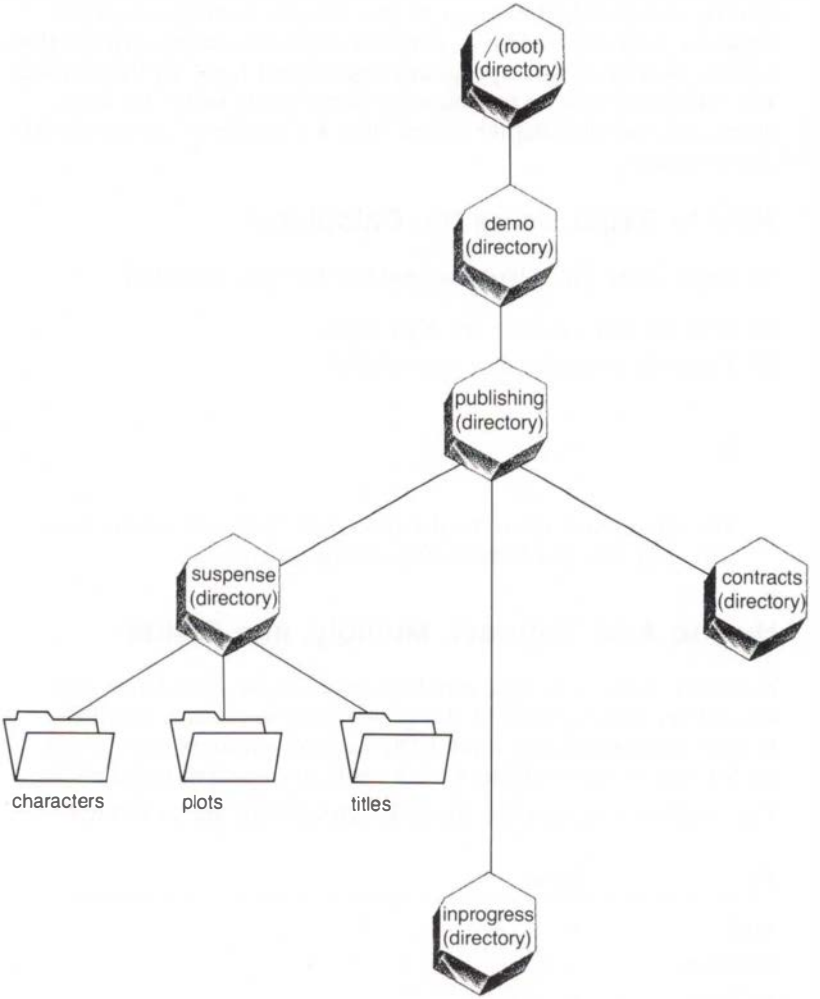
```
cd Space /demo/publishing/suspense
```

2. Type the following and press ENTER:

```
lp Space titles&
```

You will see the $ prompt on your screen. The *titles* file will begin printing.

The following diagram shows the location of the *titles* file.



For more information about printing, refer to the chapter called "Tasks" in the XENIX *User's Guide*.

If It Didn't Work...

If the $ prompt doesn't reappear, did you forget to type the & after your print command? If so, the $ prompt will reappear when your file has finished printing.

Using the Calculator: bc

The **bc** command allows you to use XENIX's built-in calculator program. You can use **bc** to perform ordinary mathematical calculations, as well as some programming-related tasks. In this section, you will learn how to use **bc** to do some basic tasks. To learn more, refer to the chapter called "BC: A Calculator" in the XENIX *User's Guide*.

How to Begin Using the Calculator

To begin using the calculator, you use the **bc** command.

Try it out: start the calculator

1. First, be sure you see the $ prompt.
2. Type the following and press ENTER:

```
bc
```

The cursor will move to the next line. You will see no message, but you can begin using the **bc** calculator.

How to Add, Subtract, Multiply, and Divide

Normally, when you type numbers into the **bc** calculator, you should use the numbers at the top of your keyboard, not the keypad numbers at the right. (The keypad numbers may be set up for you to use with **bc**. Check with your system manager.)

The symbols you type for the four calculations are as follows.

To	Press
Add	+
Subtract	−
Multiply	*
Divide	/

To add, subtract, multiply, or divide, type the first number, then the appropriate symbol, then the next number. Press ENTER to complete the calculation. The answer will appear on the next line.

Try it out: multiply

In this example, you will multiply 5 times 8.

1 Type the following (use the SHIFT key to type the asterisk):

```
5*8
```

2 Press ENTER.

The answer appears on the next line on your screen:

```
40
```

If It Didn't Work...

If you typed the numbers shown above and nothing appeared on your screen, check to see whether you were using the numbers at the top of your keyboard or the keypad numbers at the side. Remember that the keypad numbers may not be set up for you to use. Try again and be sure to use the numbers at the top.

Try it out: subtract

In this example, you will subtract 533 from 811.

1 Type the following (remember to type the number 1, not the letter "l"):

```
811-533
```

2 Press ENTER.

The next line on your screen shows the answer:

```
278
```

Using Decimals in Division

Normally, the **bc** calculator does not display either a remainder or a decimal when a division comes out uneven. You can set up the calculator to display decimal places.

Try it out: use decimals

In this example, you will set the calculator to display two decimal places.

1 Type the following:

```
scale=2
```

2 Press ENTER.

Now, the results of your calculations will be displayed with two decimal places. This feature is in effect until you quit using the calculator. The next time you use **bc**, you will need to type "scale = 2" again if you want decimals displayed.

You can have as many decimal places as you want. If you decide you need three, type "scale = 3". For four places, type "scale = 4", and so forth.

Using Parentheses

You can use parentheses in your calculations if you wish. When parentheses are used, **bc** does the calculations inside the parentheses first, just as you would do in calculating algebra.

Try it out: use parentheses in calculations

In this example, **bc** calculates 12 + 4 and 9 − 1, then divides the results.

1 Type the following:

```
(12+4)/(9-1)
```

2 Press ENTER.

The answer you see is:

```
2
```

To Stop Using the Calculator

When you have finished using the calculator, type **quit**. The **$** prompt appears.

Try it out: stop the calculator

1 To quit using **bc**, type the following:

```
quit
```

2 Press ENTER.

You must spell out the word. Just typing **q** won't work. The **$** prompt appears on your screen, showing that you are out of **bc**.

Looking at a File: more

At times, you may just want to read through a file and not make any changes to it. You could do this by using the Vi editor, but earlier in this book you used the **more** command, which is a much quicker and easier way to display a file on your screen. In this section, you will learn about the **more** command in greater detail.

A nice feature of **more** is that it displays a screenful of text, then stops and waits for you to tell it to advance to the next screenful. To look at a file, you use the **more** command followed by the pathname (full or relative) of the file you want to see.

In this example, suppose you want to look at your *mystery2* file.

Try it out: look at a file using more

1. Type the following (remember to type your home directory name instead of "agathac"):

```
more  /usr/agathac/mystery2
      |
      Space
```

2. Press ENTER.

   The first part of the file will be displayed on your screen, as follows:

```
$more /usr/agathac/mystery2
                        WORK IN PROGRESS
                    (The Epiphany Affair)

     It was a dark and stormy night.  Miss Jane Marple sat by the hearth
and its cozy fire, oblivious to the racket the wind made at the shutters.
It was so good to be home again in one's favorite chair after a holiday away.
A long string of holidays, actually.  Christmas, Boxing Day, New Years,
and at last, Twelfth Night -- Epiphany.  Her nephew, Raymond,
had been insistent that she stay with him in his new flat in London
for the season.
     "Aunt Jane, you're becoming quite the stay-at-home," he'd persisted.
"Really, you haven't been away from St. Mary Mead since your trip to the
Caribbean, and that was three years ago.  Wouldn't a change of scene do
you good?"
     Miss Marple knew this was Raymond's tactful way of suggesting she
must surely be bored with village life by now.  Young people had so
little understanding of the intrigue one found staying in one small
place and watching carefully.  Subtleties, nuances,...  Oh well, London
--More--(34%)
```

The "--more--(34%)" at the lower left of the screen shows that you are using **more** to view a file and that 34% of the file has been displayed thus far, leaving 66% still to be displayed.

3. To look at the next line in the file press ENTER.
4. To view another full screen of text, press the SPACEBAR.

   You can continue viewing the file in either way, until you reach the end. After the last line of the file has been displayed, you will return to the $ prompt.
5. If you want to stop reading the file before you reach the end, press the following:

   q

   The $ prompt appears.

Note that if you want to look at a file in your working directory, you can follow this same procedure but use just the filename of the file you want. You don't need the full pathname.

Looking at a File: cat

At times, you may want to skim quickly through a file instead of paging through it using the **more** command. This is especially useful with a long file. One way to do this is to use the **cat** command followed by the pathname of the file you wish to see.

Stopping the File from Scrolling

When you use the **cat** command to look at a file, the file will continue to "scroll" upwards on your screen until the end of the file is reached or until you stop it. It is fairly easy to control the scrolling.

To stop the scrolling, press CTRL-S. This means you press and hold down the CTRL (CONTROL) key, then press the "s" key. To restart the scrolling, press any key.

Try it out: look at a file using cat

In this example, assume you want to look at your *mystery2* file again.

1. Type the following (use your home directory name instead of "agathac"):

```
cat Space /usr/agathac/mystery2
```

2. Press ENTER.

If you are already in your home directory, you can just type the filename. You don't need the pathname.

As noted, the file will continue to scroll through to the end, unless you stop it using CTRL-S. If you don't stop it, you will automatically reach the end of the file and return to the $ prompt.

Stopping the Display Before the End of the File

You might want to quit looking at a file and return to the $ prompt before the entire file has been displayed, for example, if you are looking at a very long file. To do this, press the DELETE key.

Communicating With Other Users: mail

If there are other users on your system, you will probably use the electronic mail feature of XENIX fairly often. By using the **mail** command, you can send and receive messages, send files to other users, and save your messages so you have a complete record of your communications.

In this section, you will learn how to send, edit, read, save, and delete a message. If you want to know more about using the **mail** feature after reading this section, refer to the chapter called "Mail" in the XENIX *User's Guide*.

Sending a Message Using Mail

To send mail to another user, you need to know the person's "user name." This is usually the person's login name.

To send mail, you use the **mail** command followed by the user name of the person to whom you are sending the message.

Try it out: send a message

In this example, you will send a message to yourself.

1. To send a message with **mail**, start at the $ prompt.
2. Type the following and press ENTER (substitute your login name for "agathac"):

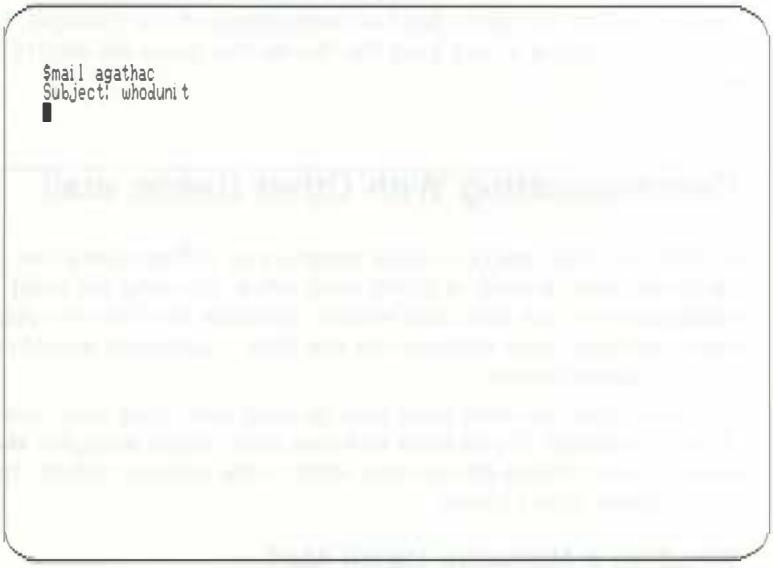
```
      Space
mail agathac
```

After a moment, you will see the following on your screen:

```
mail agathac
Subject:
```

3. Type the subject of your message (in this case, "whodunit") and press ENTER.

Your screen will look like this:



4. You can now type your message. If your message is more than one line, you must press ENTER at the end of each line to move the cursor to the next line. In **mail**, the text will not automatically wrap onto the next line.

Your message could be something like the following:

```
It was the butler.
```

5. When you have finished typing your message, press ENTER to move to a new line. You must be at the beginning of a blank line before you can send your message.
6. To send your message, press the following:

```
CTRL-D
```

This means you press and hold down the CTRL (CONTROL) key, then press the "d" key.

You will see the following on your screen:

```
(end of message)
$
```

The $ prompt tells you that your message has been sent. You may also see the message "You have mail." This tells you that the message you sent has been received at your computer. You will learn how to read the message shortly.

Correcting Typing Errors in Mail Messages

If you make an error while typing your message, you can use the BACKSPACE key to erase and correct it; however, if you have gone on to the next line, this won't work. In that case, you can use the Vi editor to correct your error. To enter **vi** while using **mail**, press ~**v**.

Try it out: edit a mail message

In this example, you will compose another message to yourself, only this time you won't send it right away.

1. At the $ prompt, type the following and press ENTER (type your own login name instead of "agathac"):

```
mail Space agathac
```

2. Type a subject and press ENTER.

3. Start typing your message. Type a few lines and include some typing errors.
4. Press ENTER to move the cursor to a new line.
5. To use **vi**, type a tilde (~) character, then **v** as follows (use the SHIFT key to type the tilde):

```
~v
```

6. Press ENTER. Your screen will change and you will see the text of your message and some information about the number of lines and characters in the message. You are now in **vi** and can use any of the commands you learned in the last chapter to edit your message.
7. To leave **vi** and return to **mail** so you can send your message, press ESCAPE.
8. Then, type the following and press ENTER (use the SHIFT key to type the colon):

```
:x
```

9. When you see the message "(continue)," you can add more text to your message if you wish. To send your message, press the following:

```
CTRL-D
```

Interrupting Your Mail Message

If you are in the process of typing a mail message and you decide not to send it for some reason, you can interrupt your message and return to the $ prompt. To interrupt a message you are typing, press the DELETE key twice.

Try it out: cancel a message

In this example, you will start to send yourself another message, then interrupt it and delete the message.

1. Type the following and press ENTER (substitute your login name for "agathac"):

```
mail agathac
```
Space

2. Type a subject and press ENTER.
3. Now assume you decided not to send this message after all. Press the DELETE key. You will see the following message:

```
(Interrupt -- one more aborts message)
```

   This warns you that your message is about to be canceled. If you wanted to send your message, you would just continue typing it and the interruption would be ignored.
4. To delete your message, press DELETE again and you will see the $ prompt. You may also see the following on your screen:

```
Letter saved in "usr/agathac/dead.letter"
[New file] 7/118
$
```

The $ prompt shows that you have left **mail**. For more information on the *dead.letter* file, refer to the chapter called "Mail" in the XENIX *User's Guide*.

Reading a Mail Message

When you see the following message, you have received mail:

```
You have mail.
$
```

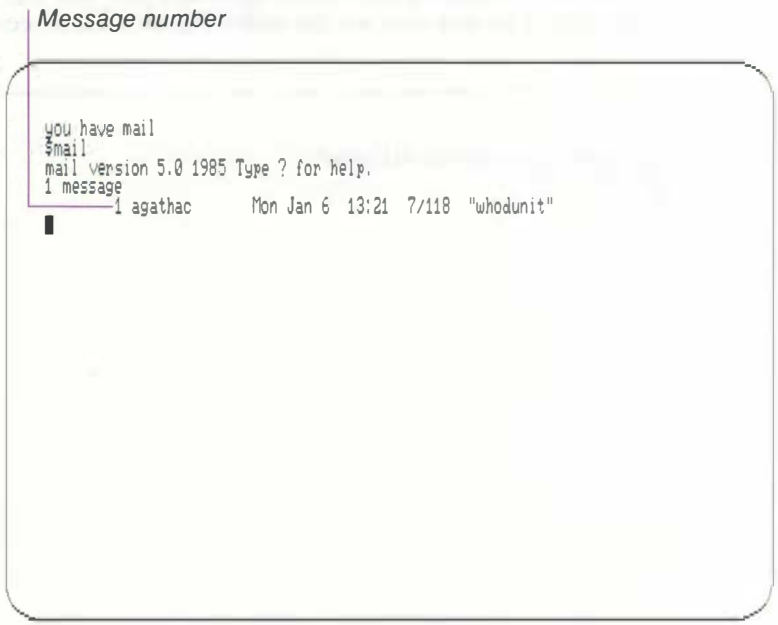
To read your message(s), use the following procedure.

Try it out: read a mail message

1. To read your message (this is probably one you sent to yourself earlier), type the following and press ENTER:

```
mail
```

Your screen will look something like the following one. Notice the "message number." Each message has a different number.



2. Press ENTER. The screen will display the text of a message. If the message is a long one, it will scroll up your screen until it reaches the end. You can stop the scrolling by using CTRL-S as described in the section on **cat** on page 100.
3. If you have another message, press ENTER and the next message will be displayed. When you have read the last message, press ENTER. Your screen will display "Can't go beyond last message."
4. If you want to reread a specific message, type **p** (for "print"), then the message number. For example, if you want to reread message number 1, type the following and press ENTER:

```
p1
```

After you read a message, it stays in your "mailbox" (a file that contains your messages) until you remove it, either by saving it in a file or deleting it. Both of these procedures are described in the next two sections.

Saving a Mail Message

In many cases, you will want to save a copy of the **mail** messages you receive.

You can save a message and place it in a file by typing **s** followed by the message number and the pathname of the file in which you want to put the message. If you don't specify a pathname, XENIX assumes the filename you type is in your working directory. When you save a message in a file, it is removed from your mailbox when you leave **mail**.

Try it out: save a message in a file

In this example, assume you want to put message 1 into a new file called *guilty* in your home directory. If you were not in your home directory when you started reading your mail, you'll need to type the full pathname (/usr/agathac/guilty).

1. Type the following:

   Space
   s1 guilty

2. Press ENTER.

   If the file *guilty* did not exist before, a new file is created. You will see something like the following on your screen:

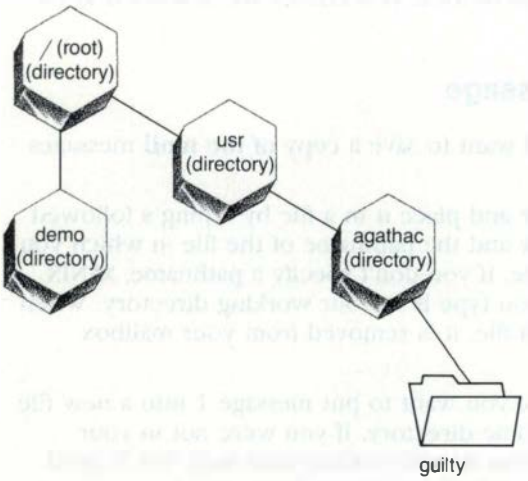
   ```
   "guilty" [New file]  6/73
   ```

   If you already had a file called *guilty*, you would see the following on your screen:

   ```
   "guilty" [Appended]  6/73
   ```

   Both messages tell you your mail message has been saved in a file called *guilty*.

The following diagram shows the location of the *guilty* file.



Deleting a Mail Message

You may want to delete a message you have received instead of saving it. To delete a message, press **d** (for "delete").

Try it out: delete a message

In this example, you will delete the second message you sent to yourself.

1. To reread message 2, type the following and press ENTER:

   ```
   p2
   ```

2. After you have read the message, type the following and press ENTER:

   ```
   d
   ```

   When you quit the **mail** session, the message will be gone.

Quitting the Mail Session

When you have finished reading your messages and want to leave **mail**, type **q** (for "quit").

Assume you have finished reading your mail for now.

Try it out: quit mail

1. Type the following:

```
q
```

2. Press ENTER.

   You will see this message:

```
Held 0 messages in /usr/spool/mail/agathac
```

   This tells you that your mailbox is currently empty.

Running an Application

You may have one or more special programs on your XENIX system that you will use to process or store a certain kind of information. These programs are called "applications." For example, suppose you run a detective agency and you have a program for keeping track of the hours you have worked for each client. This program might also compute the amount of money each client owes you and print a bill at the end of each month. If you have a special program such as this, your system manager will need to set up the program on your XENIX system.

To use the program, you need to know the command to type. For example, the billing program described above might be used by typing "bills". Thereafter, most programs will guide you by displaying instructions and questions on the screen. You simply indicate your choices and enter information as directed. Check with your system manager to find out about applications you can use.

Advanced XENIX Topics

In this book, you have learned how to perform the basic XENIX tasks. Since space is limited, many other topics have not been discussed. The following list introduces you to some advanced XENIX subjects that can help you work more easily, and refers you to your system manager or other XENIX manuals where you can learn more about each topic.

To	See
Back up and restore files on a floppy disk	Your system manager
Connect an additional floppy disk drive to your computer	Your system manager
Search for and replace specific words and phrases	The chapter "Vi: A Text Editor" in the XENIX *User's Guide*
Locate a file when you are not sure which directory it is in	The **find** command in the XENIX *Reference Manual* or the XENIX *User's Guide*
Alphabetize or otherwise sort the contents of a file	The **sort** command in the XENIX *Reference Manual* or the XENIX *User's Guide*
Combine files	The **cat** command in the XENIX *Reference Manual* or the XENIX *User's Guide*
Select and print specific lines of a file	The **grep** command in the XENIX *Reference Manual* or the XENIX *User's Guide*
Check for spelling errors in a file	The **spell** command in the XENIX *Reference Manual* or the XENIX *Text Processing Guide*

An illustration showing further topics and references, called "A Quick Guide to XENIX," appears on page 127.

Review

In this chapter, you have learned how to use some additional features of XENIX. You have learned how to:

- Print a file
- Do simple calculations on the **bc** calculator
- Read through a file one screenful at a time using the **more** command
- Scan quickly through a file using the **cat** command
- Send, edit, read, save, and delete messages using **mail**

More Practice

You have now completed all the exercises in this book. If you want more practice, you can use the files in the sample file system included on your XENIX system. Just remember that if you want to edit any files, you will need to make a copy, change permissions, and work with the copy. Refer to page 65 to see how to change permissions. To delete a copy of a file or directory, you will need to override the existing permissions. Refer to page 50.

The following sections provide a list of commands you have used in this book, a list of messages you might receive while doing the procedures discussed in this book, definitions of terms, a quick reference guide to learning other features of XENIX, and an index.

Command Summary

Command	What It Does
&	Not technically a command, but adding this at the end of a command causes the command to be run in the background.
bc	Starts the XENIX calculator.
cal	Displays a calendar for the month or year requested.
cat	Displays the named file, causing it to scroll upward on your screen.
cd	Moves you to another directory.
cp	Makes a copy of the named file.
CTRL-D	Logs you off the computer. Also, sends a **mail** message.
CTRL-S	Stops text from scrolling upward on your screen.
date	Displays the date and time.
lc	Displays a short listing of a directory's contents.
lp	Prints the named file on the lineprinter.
ls −l	Displays a detailed listing of a directory's contents.
mail	Allows you to read the messages in your mailbox.
mail *user name*	Creates a message that you can send to the named user.
mkdir	Creates the named directory.
more	Displays the requested file one screenful at a time, or one line at a time.

mv	Moves or renames a file or directory.
passwd	Allows you to change or create your password.
pwd	Displays the pathname of your working directory.
quit	Shuts off the XENIX calculator.
rm	Deletes the named file.
rmdir	Deletes the named directory.
vi	Opens the named file for editing.
who	Displays information about other users currently using your system.
who am i	Displays information about the user who types the command.

Messages

Can't go beyond last message

- *This tells you that you have reached the end of the messages in your mailbox.*

command: cannot make directory name

- *You tried to create a directory called (name) that already exists.*

  Either use the existing directory for the remaining exercises, or remove it and use the **mkdir** command again.

command: not found

- *You typed something that is not a XENIX command.*

  Try again and be sure to type the command exactly as shown.

***filename* [New file]**
***filename* [Appended]**

- *Both of these messages tell you that your mail message has been saved in the named file.*

Held x messages in /usr/spool/mail

- *You have (x) messages in your mailbox.*

[Hit RETURN to continue]

- *This message appears in response to various commands.*

  To continue with what you were doing, simply press the ENTER key.

(Interrupt—one more aborts message)

- *You pressed the DELETE (DEL) key while typing a mail message.*

  To delete the message, press the DELETE key again. To continue with your message, just resume typing.

Login incorrect

- *You typed something incorrectly when logging in.*

  Try again and be careful to type your login name and password correctly.

name: bad directory:

- *The directory called (name) that you typed in your command does not exist.*

  Try again and be sure to type the name correctly. If you continue to receive this message after a few more tries, check with your system manager to be sure the directory exists.

No lines in the buffer

- *You may see this message when you open a file using Vi.*

It has no effect on what you are doing, so you can ignore it. For information about buffers, refer to the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*.

No local user named "*name*"
Letter saved in "/usr/_____/dead.letter"

- *The name you addressed your mail message to doesn't exist.*

  Check the spelling of the user's name, then try again. (Did you accidentally type "agathac"?)

No previous regular expression

- *You accidentally stumbled on the "search" feature of Vi.*

Ignore this message and go on with what you were doing. If you want to learn about the search feature, refer to the chapter called "Vi: A Text Editor" in the XENIX *User's Guide*.

[No write since last change]

- *You are probably attempting to temporarily leave your editing session to run another command.*

  This message warns that you have made some changes to your file that you haven't saved yet.

No write since last change (:quit! overrides)

- *You are attempting to leave a file by pressing :q.*

  This message warns that you have made some changes to your file that you haven't saved yet. If you want to quit the file without saving the changes, type **:q!**.

Password unchanged.

- *You may have accidentally typed your old password when you should have typed the new one.*

  Try the password procedure again, and make sure to use a different word for your new password.

Sorry

- *You may have typed your old password incorrectly.*

  Try the password procedure again, using the correct password.

syntax error on line 1, teletype

- *You may see this message if you type something incorrectly while using the bc calculator.*

  Try your calculation again.

They don't match; try again.
New password:

- *You probably made a mistake when you retyped your new password.*

  Try the password procedure again, and be sure to type your password the same both times.

Too short. Password unchanged.

- *You used a new password that had less than 5 characters.*

  Try the password procedure again, and be sure your new password has at least 5 characters.

Unknown command character '_':

- *In command mode in Vi, you pressed the colon, but then you pressed some character other than w, x, or q!.*

  Try again.

Terms

$ The dollar sign is the symbol for the system prompt.

/ This is the slash symbol. *See also* slash.

.. This is the shorthand symbol for the parent directory. *See also* relative pathname.

account Only certain individuals are authorized to use the computer. The system manager gives each authorized user an account on the system. If you have an account, you can log in to the computer using your login name and use the files and directories.

application An application is a special program that you use on your computer to process or store a certain kind of information. An example of an application is the program Microsoft® Multiplan®.

background Background relates to the multi-tasking feature of XENIX. When you run two or more tasks at a time, only one will show on your screen. The others are said to be running in the background. *See also* multi-tasking.

bit Bit is short for "binary digit." A bit is the smallest unit of information on a computer. A bit represents the presence or absence of an electric current. A bit can be either 1, which means the current is switched on, or 0, meaning the current is off. Eight bits make up one byte.

byte A byte is a unit of information on a computer. A byte is made up of eight bits, and represents a character such as a letter, number, or symbol you type at the keyboard or see on your screen. For example, the letter "A" is usually represented by the byte that looks like this: 01000001.

case Case relates to capital letters and small letters. Uppercase letters are capital; lowercase letters are small. XENIX responds differently to uppercase and lowercase letters, so be sure you use the right case.

character A character is a letter, number, or symbol that you type at your keyboard or see on your screen.

command A command is an instruction to the computer to perform a task. For example, you type "date" and the **date** command tells the computer to display the current date and time on your screen.

command mode When using the Vi text editor, you are in either command mode or insert mode. In command mode, pressing certain keys moves the cursor, deletes words, deletes lines, and so forth. *See also* insert mode.

computer A computer is a machine that processes and stores information. The word "computer" is commonly used to mean the screen and keyboard, as well as the information-processing and storage components of the machine. *See also* hardware.

control character A control character is a combination of the CTRL key plus a letter key. Pressing these keys gives XENIX a special command. For example, CTRL-S stops text from scrolling on your screen, and CTRL-D logs you off the computer.

cursor The cursor is the lighted shape on the screen that shows where the next character you type will appear. The cursor is usually a blinking line or small box.

directory A directory is a unit that XENIX uses to store information. In your computer, files of information are stored in directories. There are usually many directories on your computer, and they are organized in several different levels. Directories on the lower levels are stored in the directories above them. *See also* file.

Ed Ed is one of two text editors that are included with your XENIX system. *See also* Vi.

editor Editor is short for "text editor." A text editor is a computer program that you use to create new files and to make changes to files.

electronic mail Electronic mail is the XENIX message feature that you use to send and receive messages to and from other users.

file A file is XENIX's basic unit for storing information. For example, you might have a file called *mystery* that contains the first draft of a story you are writing. *See also* directory.

file system A file system is a group of related files and directories. For example, in this book you have used the files and directories that are contained in the *demo* directory. All of these files and directories are part of the XENIX file system.

full pathname The full pathname is the pathname that begins at the top of the XENIX file system with the root (/) directory. For example, /demo/publishing/inprogress/mystery is the full pathname of the *mystery* file discussed in this book. *See also* relative pathname.

hardware Hardware is the keyboard, screen, circuit boards, and so forth that make up the physical parts of the computer. *See also* software.

home directory Each XENIX user has a personal directory called the home directory. This is the directory in which you usually keep your own files and directories.

insert mode When using the Vi editor, you are in either insert mode or command mode. In insert mode, the characters you type are put into your file, just as typing at a typewriter puts characters on the page. *See also* command mode.

log in When you log in, you open a line of communication between you and the computer. This is done by typing your login name and, usually, a password.

log out When you log out, you close the line of communication between you and the computer. In XENIX, this is done by pressing CTRL-D.

login name Your login name is the name you use when you log in, to connect to the computer. Your login name may be your own name, the name of your department, or some other name your system manager told you to use. Usually, your login name is also the name of your home directory.

mail *See* electronic mail.

mailbox Your mailbox is the file in which messages you have received from other users are stored. You can read the messages in your mailbox by using the **mail** command. *See also* electronic mail.

memory Memory is the place in your computer where information is stored.

message A message can be a communication you send or receive using XENIX's electronic mail feature. It can also be a note that appears on your screen giving you information about the system, for example, "Login incorrect."

monitor The monitor is your computer's screen.

multi-level Multi-level refers to the XENIX file system. It describes the fact that directories can be stored in other directories. For example, in the sample *demo* directory you used in this book, the *inprogress* directory was stored in the *publishing* directory, and the *publishing* directory was itself stored in the *demo* directory.

multi-tasking Multi-tasking describes the ability of a system such as XENIX to work on more than one task at a time. For example, you can be printing one of your files at the same time that you are editing another file. *See also* background.

operating system An operating system is the set of instructions (programs) that have been placed in the computer to make it work for you. The XENIX Operating System tells the computer hardware how to do the tasks you request.

option An option is a letter or number that is added to a command to make the command work in a slightly different way. For example, the **−l** option to the **ls** command lists the contents of the directory in long format. This gives more information than using just the **ls** command alone.

parent directory The parent directory is the one in which a file or directory is stored. For example, in the /demo/publishing/inprogress/mystery pathname, *demo* is the parent directory of *publishing*, *publishing* is the parent directory of *inprogress*, and *inprogress* is the parent directory of the *mystery* file.

password A password is the word you type in response to the password prompt when you log in. Your password identifies you to the computer, so be sure you are the only one who knows your password. If you type the wrong password, the connection to the computer will not be made.

pathname The pathname is the set of directions XENIX follows from one directory to the next until it reaches the file or directory you want to use. For example, the pathname for the *mystery* file in the sample file system you used in this book is /demo/publishing/inprogress/mystery. This means that to get to the *mystery* file, XENIX goes to the root (/) directory, then to *demo*, then *publishing*, then *inprogress*. It finds the *mystery* file in the *inprogress* directory.

permissions Permissions control which files and directories on the XENIX system can be used by which people. Permissions keep unauthorized people from using your files and directories.

process In this book, process is used to mean a task or command that the computer is running. For example, if you print a file using the **lp** command, this is called a process that the computer is running.

program A program provides the software instructions to the computer that cause it to perform a certain task. For instance, the **mail** program can be used to send and receive messages between computers, and the **more** program allows you to view a file.

prompt A prompt is the word or symbol XENIX shows on the screen to tell you it is ready for you to type something. For example, the password prompt asks you for your password. *See also* system prompt.

relative pathname The relative pathname is the shortcut version of a file or directory pathname. The relative pathname instructs XENIX to follow the shortest path from the working directory to the requested file or directory. For example, the relative pathname to a directory just above any directory is two periods (..). *See also* full pathname.

root directory The root directory is the highest-level directory in the entire XENIX file system. The root directory is shown by the forward slash (/), and is the first character listed in a file or directory pathname.

sample file system The sample file system you use in this book consists of the *demo* directory plus all of the directories and files contained in it.

save When you save information, you store it in the computer's memory. For example, you save the editing changes you make in a file. *See also* write.

screen The screen refers to the video display where you can see the characters you type and any messages the computer generates.

scrolling Scrolling is the movement of text on your screen as it rolls up and off the top of the screen.

slash In this book, slash means the forward slash (/), although your keyboard may also have a backwards slash (\). At the beginning of a pathname, the slash indicates the root directory. In the rest of the pathname, the slash is used to separate the files and directories in the pathname. In the following pathname, you can see both uses of the slash: /demo/publishing/inprogress/mystery.

software Software provides the instructions in the computer that make it perform tasks. Software includes the operating system, as well as any application programs you might put into the computer, such as Microsoft Multiplan. *See also* hardware.

subdirectory A subdirectory is any directory that is stored in another directory in the file system. For example, in the pathname /demo/publishing/inprogress, *publishing* is a subdirectory of *demo*, and *inprogress* is a subdirectory of *publishing*.

system In this book, system is used as an abbreviation for the XENIX Operating System. *See also* operating system.

system manager Your system manager is the person in charge of your XENIX system. This is the person you should go to when you have questions about your computer that you can't find answers to.

system prompt The system prompt is the symbol you see on the screen that tells you the computer is waiting for your instructions. The normal XENIX prompt is the dollar sign ($).

terminal A terminal is the machine you use that is connected to the main computer in your XENIX system.

text editor A text editor is a program that you use to make changes to the text of a file. Text editors are generally referred to simply as editors. The two editors included with your XENIX system are Vi and Ed.

user A user is a person who uses the computer.

user name This is the name that is used to send a **mail** message to a user. This name is usually the person's login name.

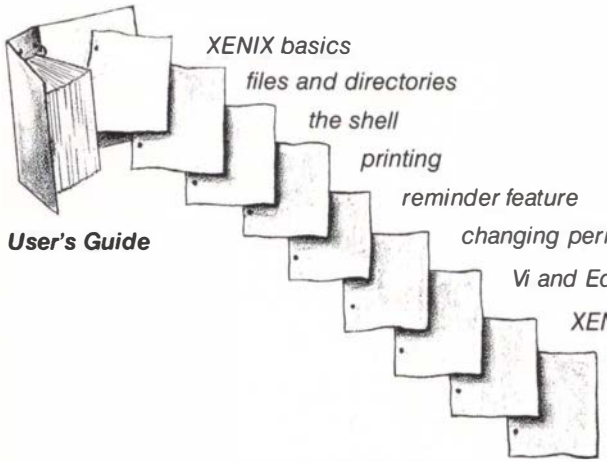
Vi Vi is the primary text editor included with XENIX. *See also* Ed.

working directory Your working directory is the directory you are currently working in. For example, when you are working in your home directory, it is your working directory. If you move to the *demo* directory, it becomes your working directory.

wrap When you are typing text and you reach the end of a line and the cursor automatically moves to the next line, you say the text "wraps" to the next line. You may need to check with your system manager to have this feature set up for you.

write Write is another way of saying you save information in the computer. *See also* save.

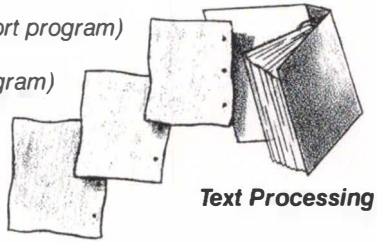
A Quick Guide to XENIX



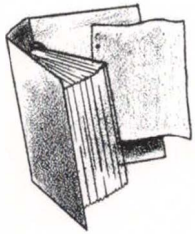
XENIX basics
files and directories
the shell
printing
reminder feature
changing permissions
Vi and Ed text editors
XENIX mail system (mail)
XENIX calculator (bc)

User's Guide

alphabetizing lines in files (sort program)
checking your spelling (spell program)
formatting a document



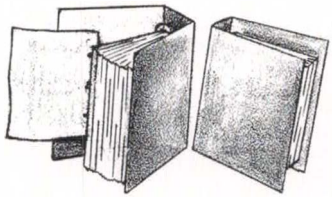
Text Processing Guide



choosing tasks from a list of commands

Visual Shell User's Guide

descriptions of commands and how to use them



Reference Manual

Index

MICROSOFT®

Introducing XENIX®

This book tells you everything you need to know to get up and running on a computer running the Microsoft® XENIX operating system.

Written for people who have little or no experience using computers, *Introducing XENIX* gives you an overview of the Microsoft XENIX manuals. It also teaches you all the basic XENIX commands, including:

- How to log in.
- How to create and change files.
- How to use the XENIX calculator.
- How to use XENIX mail.

1185 Part No. 091-192-001

Introducing XENIX®

MICROSOFT®