JUST ENOUGH

UNIX

PAUL K. ANDERSEN

JUST ENOUGH UNIX

PAUL K. ANDERSEN
PURDUE UNIVERSITY

WCB Wm. C. Brown Publishers
Dubuque, Iowa • Melbourne, Australia • Oxford, England

Preface

What is UNIX?

Mention computers, and most people tend to think of computer *hardware*—the physical device, consisting of circuit boards, a central processing unit (CPU), memory chips, etc. Equally important, however, is the *software*—the programs that tell the hardware what to do. Without software, a computer is just a box with wires attached to it.

An *operating system* is an important kind of software that manages the resources of the computer. You might think of the operating system as the master control program for the entire computer system, hardware and software.

In this book you will learn about the UNIX operating system. UNIX is fast becoming the standard computer operating system in industry, government, and education. It is especially popular in academia: according to AT&T, where UNIX was developed, every major university in the United States now has at least one computer system running under UNIX.

Which Version of UNIX?

Although UNIX originated at the AT&T Bell Laboratories, much of its subsequent development has occurred at the University of California, Berkeley. Computer manufacturers, too, have gotten into the act, producing their own variations on the UNIX theme. Examples include AIX, from IBM; A/UX, from Apple Computer; HP-UX, from Hewlett-Packard; SunOS, from Sun Microsystems; ULTRIX, from Digital Equipment Corporation; and XENIX, from Microsoft.

These versions of UNIX are quite similar. Most can trace their ancestry to either AT&T UNIX or Berkeley UNIX; some are amalgams of both. This book presents features that are found on almost all UNIX systems, with special emphasis on those that are common to AT&T System V and Berkeley System Distribution (BSD) 4.3 UNIX.

Who Should Read This Book?

This book is intended for anyone who wants to acquire a working knowledge of UNIX without having to become a UNIX expert. It is especially appropriate for students of science, engineering, or business who are taking their first computer programming course.

What Does This Book Cover?

This book covers the basics of the UNIX operating system. It has eight main parts:

I. INTRODUCTION TO UNIX

II. THE UNIX FILE SYSTEM

III. THE VISUAL EDITOR

IV. UNIX COMMUNICATIONS

V. THE UNIX SHELL

VI. SHELL SCRIPTS

VII. PROGRAMMING UNDER UNIX

VIII. DOCUMENT PREPARATION

INTRODUCTION. In Part I, you will find an overview of the UNIX operating system, and you will learn what you will need to start using it.

THE UNIX FILE SYSTEM. UNIX organizes information in collections called *files*. You will learn how to create, name, rename, copy, and delete files in Part II. You will also learn how UNIX keeps track of your files.

THE VISUAL EDITOR. You can create or modify UNIX files using a utility program called an *editor*. The most popular UNIX editor is called **vi** ("vee-eye"), which is discussed in Part III.

UNIX COMMUNICATIONS. One of the most useful utilities is *electronic mail* (or "E-mail"), which allows you to communicate with other users on the system. The UNIX electronic mail utilities **mail** (Berkeley UNIX) and **mailx** (AT&T System V) are discussed in Part IV.

THE UNIX SHELL. The part of UNIX that interprets user commands and passes them on to the computer is called a *shell*. Many different shells have been written for UNIX; the three most prevalent are the **Bourne Shell** (standard on AT&T System V UNIX), the **C Shell** (standard on Berkeley UNIX), and the **Korn Shell**. These shells are considered in Part V.

SHELL SCRIPTS. The UNIX shell is also a sophisticated programming language. A file containing a program for the UNIX shell is called a *shell script*. Shell scripts are described in Part VI.

PROGRAMMING UNDER UNIX. Almost every UNIX system includes the programming languages C and FORTRAN. Most also include Pascal, and your system may have other languages such as BASIC, Lisp, COBOL, etc. UNIX also offers various software tools called *debuggers*, which are used to find errors ("bugs") in programs. UNIX programming is discussed in VII, with emphasis on C, FORTRAN, and Pascal.

DOCUMENT PREPARATION. The UNIX *text formatters* **nroff** and **troff** allow you to produce attractive memos, reports, and other printed documents. The **ms** *macro package* is a collection of predefined formatting instructions that make the formatters much easier to work with. You can include tables and mathematical equations in your reports by using the table formatter **tbl** and the equation formatter **eqn**. These programs are presented in Part VIII.

How to Use This Book

Anyone who is just starting with UNIX should read straight through Parts I, II, III, IV, and V. The remaining parts may be read in any order. If you would like to learn about shell scripts, read Part VI. If you are interested primarily in using UNIX to program in C,

FORTRAN, or Pascal, you should read Part VII. If you want to use UNIX for document preparation you will want to read Part VIII.

Each part of this book begins with a chapter that explains the material without requiring the use of the computer. Other chapters are called "tutorials." These are intended to be read at the computer terminal. You should plan to spend about an hour at the terminal to cover each tutorial.

At the end of each section, you will find some short exercises. Some exercises are intended to be done at the terminal; these are marked with an asterisk (*). To derive the maximum benefit from this text, be sure to work through all of the exercises.

Notation Used in This Book

This book uses **boldface** to refer to UNIX utilities and commands, and ***bold italics*** for file and directory names. Lines that you are supposed to type into the computer, as well as the computer's responses, are shown in this type:

```
This is what you type
This is how the computer responds
```

Acknowledgments

Many persons helped in the preparation of this book. The following reviewers read the manuscript at various stages in its development and provided helpful comments and suggestions:

Eric P. Bloom
Boston University

John Carroll
San Diego State University

Richard J. Easton
Indiana State University

Charles Frank
Northern Kentucky University

Donald L. Greenwell
Eastern Kentucky University

Joe Hagarty
Raritan Valley Community College

Vasant G. Honavar
Iowa State University

Alexander Stoffel
Mayville State University

John Slimick
University of Pittsburgh—Bradford

This book could not have been produced without the invaluable assistance of Dr. M. G. Scarbrough, colleague and friend.

Table of Contents

Part I
INTRODUCTION TO UNIX

1. Introduction to UNIX

In this chapter, you will get an overview of UNIX—what it is, how it works, what it can do for you. You'll also learn some necessary computer terminology, and you'll find out what you will need to begin working on the computer.

Computer Hardware

Computers come in a bewildering range of shapes, sizes, and types. Despite their differences, almost all have the following four essential components (see Figure 1.1):

- ***Central processing unit (CPU).*** The CPU performs calculations and manipulates data.
- ***Main memory*** (a.k.a. ***primary memory***, ***internal memory***). This is the place where the CPU looks for instructions and data to process. Main memory is fast but limited in how much it can hold.
- ***Input/Output (I/O) device.*** I/O devices are used to move information to and from the computer. The most common I/O devices are keyboards, video displays, and printers.

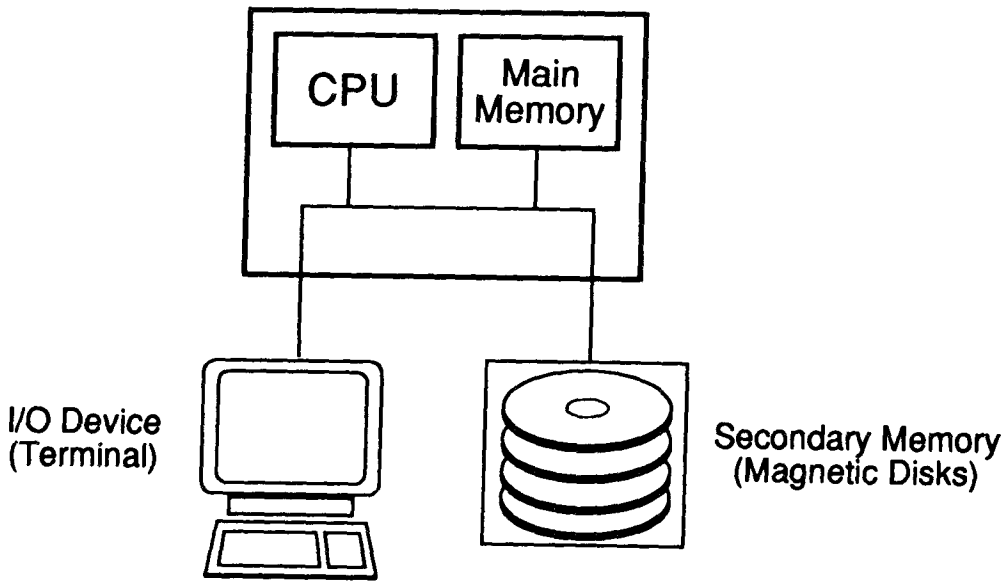

Figure 1.1. A typical computer system. The four parts of a single-user computer are shown. The CPU and main memory are almost always placed together in the same box; the disks and terminal may be separate units.

- ***Secondary memory*** (***external memory***, ***mass storage***). Information that is not immediately needed by the computer is placed in secondary memory. Secondary memory is slower than main memory but can hold much more. The most common secondary storage devices are magnetic disks.

The CPU and the main memory are usually found close together in the same unit; this is what many people mean when they talk of "the computer." Other devices, such as terminals, printers, disk drives, etc., are often called *peripherals*, because they attach to the unit containing the CPU and main memory.

One User or Many?

The computer represented in Figure 1.1 is a *single-user* system. Because it has one terminal, it is able to serve only one person at a time. This is a common arrangement with smaller, *personal* computers.

Large computer systems, on the other hand, often have more power than one person can profitably use. These computers are commonly set up as *multiuser systems*, as depicted in Figure 1.2. Note that the multiuser computer has the same four basic parts as the single-user computer: CPU, main memory, secondary memory, and I/O devices. The main difference is that the multiuser system has multiple I/O devices. (It may also have larger secondary memory.)

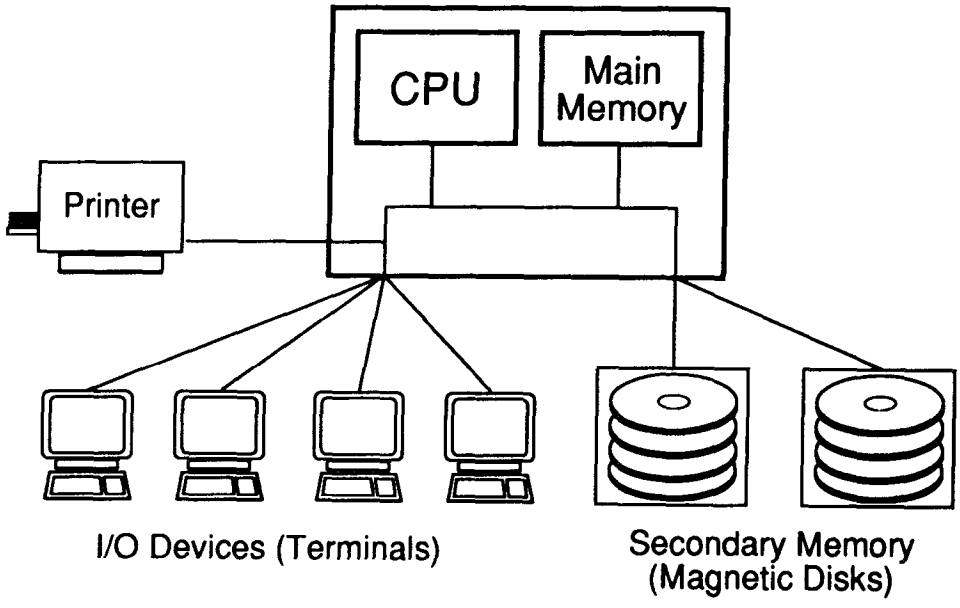

Figure 1.2. A multiuser computer system. This system is set up for four users, who share the CPU, the memory, and other resources such as the printer.

Another way to accommodate multiple users is to link two or more computers together to form a *network*. Figure 1.3 shows a network consisting of three single-user machines (called *hosts* or *workstations*), a printer, and another computer (called a *server*) that has no terminal but does have magnetic disks. In this case, none of the three hosts has its own secondary storage but instead relies on the magnetic disks attached to the server. (The server *serves* the workstations by providing secondary storage for them.)

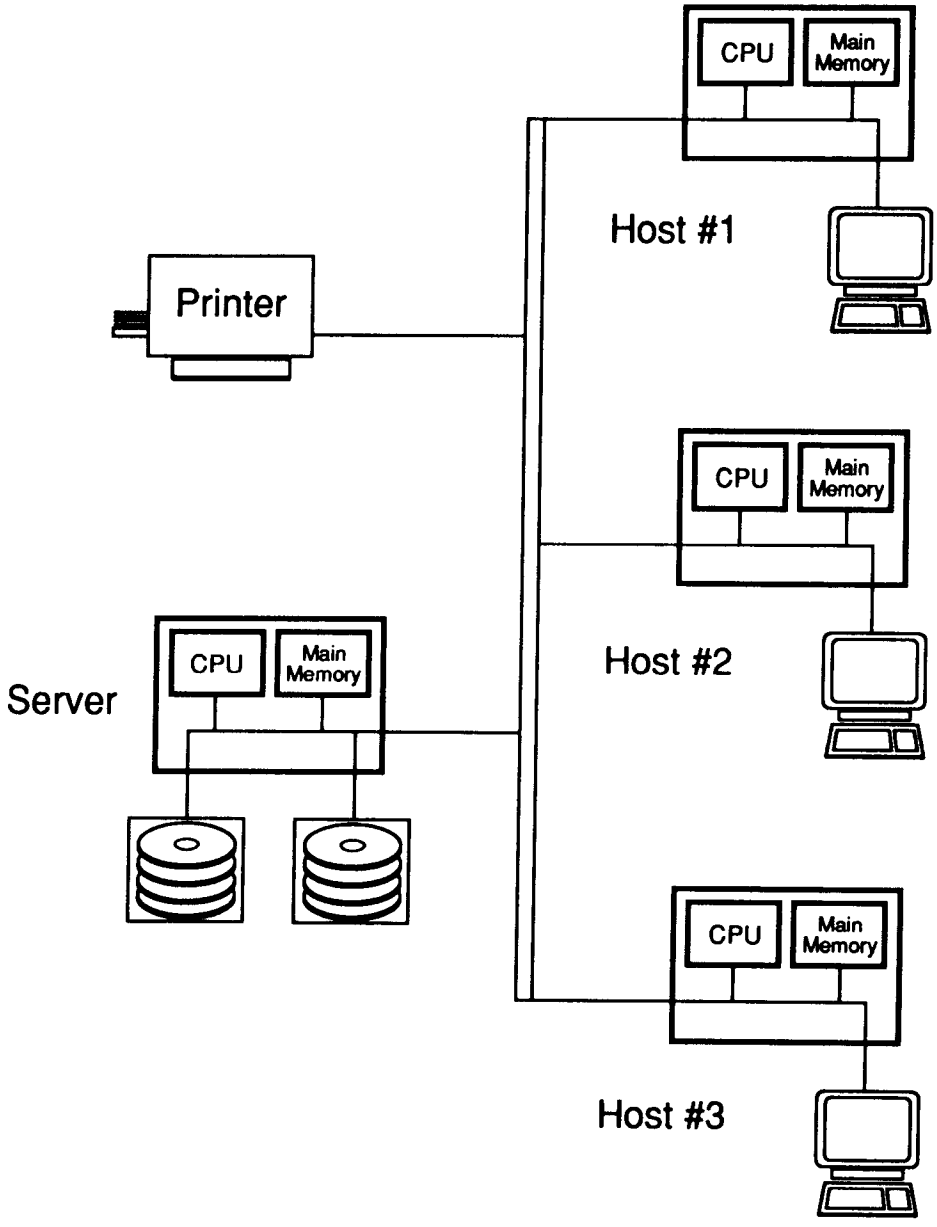

Figure 1.3. A networked computer system. This network includes four computers, one of which is used as a server providing disk storage for the other three.

Operating Systems

As important as the computer hardware is, it can do nothing without *software*, the coded instructions that tell the CPU what to do. An especially important type of software is the *operating system* which performs three vital functions. First, it controls all of the various peripheral devices—printers, disk drives, terminals, and so on—that are attached to the computer. Second, it manages the way other programs are stored and run. And third, it handles communications between the user and the computer, passing commands from the user to the computer, and returning messages from the computer to the user.

Multitasking and Time-Sharing

UNIX is a *multitasking* operating system, meaning that it enables the computer to work on more than one task at a time. The computer can run several of your programs "in background" while you work on another task "in the foreground."

How does the computer do more than one task at a time? The answer is, it doesn't; it only appears to. The computer's CPU is only able to do one thing at a time. However, by switching rapidly back and forth between tasks, performing a little here and a little there, it creates the illusion of being able to do everything at once. This method of working on many different tasks is called *time-sharing*. It is feasible only because (a) the computer is very fast, and (b) UNIX takes care of scheduling what is to be done and when.

UNIX is also a *multiuser* operating system, meaning it can serve more than one person at a time. Again, the computer's CPU can only work on one task at a time, but by switching rapidly back and forth between users, it appears to be interacting with many users simultaneously. Most of the time the computer works so quickly that each user is unaware that the computer is working with anyone else.

Not surprisingly, UNIX can also serve as a single-user operating system for personal computers and workstations. Even as a single-user operating system, however, UNIX retains its multitasking ability.

Versions of UNIX

The two most widely used versions of UNIX are AT&T System V UNIX and Berkeley Software Distribution (BSD) UNIX. Many computer manufacturers offer their own versions of UNIX. These are usually based on either Berkeley or AT&T UNIX. For example,

Version	Company	Based on
AIX	IBM	AT & T and BSD
A/UX	Apple Computer	AT&T System V
HP-UX	Hewlett-Packard	BSD 4.3
SunOS	Sun Microsystems	BSD 4.3
ULTRIX	Digital Equipment Corp.	BSD 4.2
XENIX	Microsoft	AT&T Version 7

Keep in mind that these UNIX versions are very similar; if you learn to use one, you should have little trouble with any of the others.

Major Components of UNIX

The UNIX operating system consists of four main parts:

- ***Kernel.*** The *kernel* is the master control program of the computer. It resides in the computer's main memory, and it manages the computer's resources. It is the kernel that handles the switching necessary to provide multitasking.

- ***Shell.*** The part of UNIX that interprets user commands and passes them on to the kernel is called the *shell.*

- ***File System.*** UNIX organizes information into collections called *files.* You can put just about any kind of information into a file—a program you have written, a memo, data waiting to be analyzed, the manuscript for your next novel, even a letter to your mother. Files may be grouped together into collections called *directory files*, usually called *directories.*

- ***Utilities.*** A *utility* is a useful software tool that is included as a standard part of the UNIX operating system. Utilities are often called *commands.* UNIX provides a rich set of utilities for word processing, programming, database management, communications, etc.

UNIX Shells

Three different shells are commonly used today. The Bourne Shell was developed at AT&T by Steve Bourne; it is standard on AT&T System V UNIX. The C Shell was written by Bill Joy for Berkeley UNIX. The Korn Shell was written by David Korn at AT&T; it is based on the Bourne Shell, but it includes many of the features of the C Shell. Most UNIX installations offer all three of these shells; some have other shells as well. This book examines the Bourne, Korn, and C Shells.

Your Terminal

You will communicate with the computer using a *terminal*, which consists of a typewriter-like keyboard attached to a video display device. (You may still occasionally run across an old teletype terminal, which prints on a roll of paper instead of a video display.)

UNIX operates in what is called *full-duplex* mode. This means that any characters that you type on your terminal keyboard are first sent to the computer, which then echoes them back to your terminal to be displayed on the screen. For this to work correctly, you must tell the system what kind of terminal you are using. Many different terminals are available on the market; each type is identified by a code, called a *termcap* or *terminfo* code. These codes are typically abbreviations for the terminal model designation. For example,

Manufacturer	Model	Code
Applied Digital Data Systems	Regent 20	reg20
Digital Equipment Corp.	VT100	vt100

Hewlett-Packard	2621	hp2621
Lear-Siegler	ADM 3	adm3
Liberty Electronics	Freedom 100	f100
Perkin-Elmer	550	pe550
Sanyo	55	sanyo55
Televideo	925	tvi925
Tektronix	4015	tek4015
Wyse Technology	50	wyse50
Zenith Data Systems	19	z19

Your Printer

You'll want a way to print your files. (Paper output from a printer is usually called *hardcopy*.) A large computer installation may have a variety of printers. Individual printers are identified by a code, which usually indicates the type and location of the printer. To give you an idea of what these codes may look like, here are some of the hardcopy output devices available at Purdue University:

Location	Type	Code
Computing Center	Versatec Plotter	puccvp
Math Building	C. Itoh	mathci
Matthews Hall	Dataprinter	mthw
Owens Hall	Apple Laser Writer	owenlw

Your UNIX Account

Most UNIX systems require that you set up an account before you can use the computer; see your instructor, consultant, or system administrator about this. When your account is created, you should receive the following information:

- the name of the computer you'll be using
- an account or user name
- a password

Your account name is often called your *login*, because you use it to identify yourself when you "log in" to use the computer. (*Logging in* is the process of gaining access to the computer.) Your login will typically be some variation of your real name. It may consist of as many as eight letters or numbers; however, it must begin with a letter.

Your *password* tells the computer that you are who you say you are—passwords are meant to prevent unauthorized use of the computer. *Never share your password with anyone.* It is a good idea to change your password frequently—you will see how in the next chapter.

While you are setting up your account, it is a good idea to ask a few questions:

- Which variety of UNIX will I be using—AT&T or Berkeley?

- Which UNIX shell will I be using?
- What kind of terminal will I be using? What is its code name?
- Which printer will I use? What is its code name?
- What is the login procedure? (May I have that in writing please?)

Exercises

(1) Define: (a) hardware; (b) software; (c) CPU; (d) main memory; (e) internal memory; (f) mass storage; (g) I/O device; (h) network; (i) terminal; (j) peripheral; (k) operating system; (l) multiuser; (m) multitasking; (n) kernel; (o) shell; (p) file; (q) directory file; (r) directory; (s) utility; (t) login; (u) password; (v) host; (w) server.

(2) Large computer installations typically have several public terminal rooms. How many terminal rooms does your installation have? Where are they located?

(3) Name the two most popular versions of UNIX. Which one runs on your machine?

(4) Name the three most common UNIX shells. Which shell will you be using?

(5) Who can you ask for help if you run into problems with your UNIX system?

(6) How can you print your files? Where is the printer located? What is the code for the printer you will use?

(7) What kind of terminal will you be using? What is its code?

2. Tutorial: Getting Started

In this chapter you will learn how to gain access to the computer. Unfortunately, UNIX installations do differ, so it is a good idea to have someone standing by to help with the specifics, especially the first time.

If you haven't done so already, ask your instructor, system administrator, or consultant about setting up an account. You should already know the name of the computer you'll be using, your account or login ("log-in") name, and your password.

The Terminal

Your next destination is the computer terminal room. Find a terminal and get comfortable. The terminal should have a keyboard, similar in many ways to the keyboard of a typewriter (Figure 2.1). However, the terminal keyboard has some important keys that are not found on a typewriter. See if you can locate the following five keys on your terminal:

(1) RETURN—Also called NEWLINE or ENTER key, the RETURN key is almost always located on the right hand side of the keyboard.

(2) CONTROL—This is usually on the left hand side of the keyboard. It is generally used in combination with another key, both keys being pressed simultaneously.

(3) ESCAPE—The usual position for this key is near the upper left corner of the keyboard.

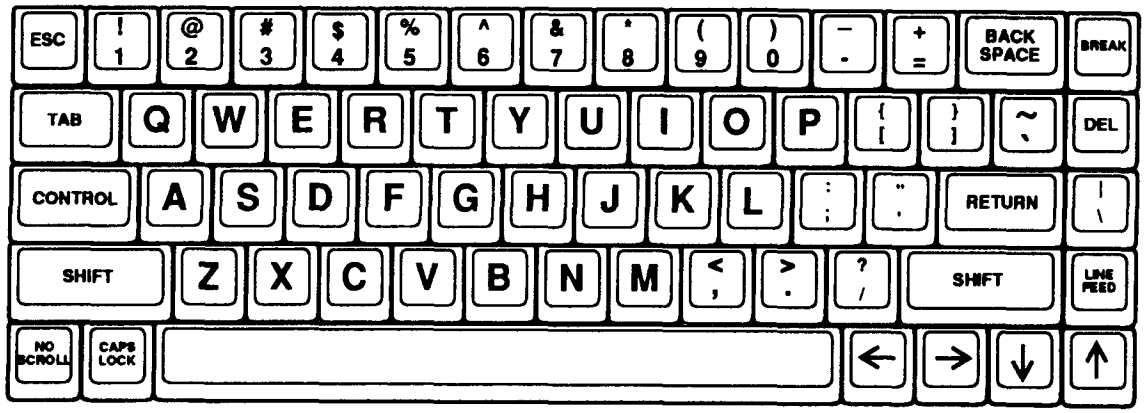

Figure 2.1. A typical computer keyboard. Note the locations of the ESCAPE (ESC), BREAK, and CONTROL (CTRL) keys. Also note that the keyboard depicted here has both a BACKSPACE key and a DELETE (DEL) key; one of these will usually serve as the erase key. Your computer keyboard may differ somewhat in the names and locations of the keys.

(4) ERASE—This may appear as the BACKSPACE, DELETE, or RUBOUT key. Some keyboards do not have a separate erase key but instead use a combination of keys such as CONTROL+h. (Hold down CONTROL, and press the *h* key.)

(5) BREAK—On some terminals this key is labeled RESET. It is used to alert the computer that you want to use it.

Now find the "on" switch. This may be hidden under the front edge of the keyboard or on the back panel of the terminal. (See if you can find the power cord—the switch may be close to the point where the cord enters the terminal.) Turn on the terminal, and allow it to warm up.

After a while, you should see a small, blinking line or rectangle on the terminal screen. This is the *cursor*. It shows where the next typed character will appear.

Connecting to the Computer

Now you must get the computer's attention. This is usually very easy if your terminal is directly wired to just one computer. Simply press the RETURN key:

RET

Some systems require that you alert the computer first by pressing RESET, BREAK, or CONTROL+BREAK. (For the latter command, press the BREAK key while holding down the CONTROL key.)

Things are a bit more complicated if you are using a modem to communicate with the computer over a telephone line. This requires that you have the computer's telephone number and that you know how to use your modem. The telephone number you can get from your instructor or system administrator; instructions on the use of the modem should have been included with the modem itself.

If several computers are tied into your system, you will be asked to specify the one you want. You may be asked

```
Which computer?
```

Or perhaps something like

```
Dial:
```

This is where you need to know the name of the computer on which you have an account. Just type in the computer name you were given, and press the RETURN key.

Logging In

You will know when you have reached your computer because it will identify itself and ask you to log in:

```
login:
```