How To Use Your Computer

by Lisa Biow
$22.95
400 pages

"How to Use Your Computer" is an exciting productivity guide for new computer users who are asking "Now What?". Hundreds of colorful graphics highlight the components of a computer and show how this complex machine actually makes life easier.

This section contains Chapters 2, 3, 5, and 6 from "How To Use Your Computer". To order the full book, call 1-800-688-0448, extension 199.

Click the turned up pages on the right and left corners to go forward and backward through the book. Click the sides of the almanac to jump a chapter at a time.

Chapter 2
Anatomy of a Computer

In this chapter, you will learn the fundamentals of computer hardware-what the essential parts are and how they interact with each other. I will start with the core elements of the computer-the parts that let you store and manipulate information, and allow you to communicate with your computer. Then at the end of the chapter I'll backtrack a bit and discuss the pieces that support and/or connect those core devices, turning the individual parts into a cohesive working system.

Probably the most important knowledge you will glean from this chapter relates to how your computer stores information. By the time you finish this chapter, you will know

>What happens when you load a program

>Where the data you type into your computer goes

>What happens when you save data (where the computer puts it, and how you get it back again)

Whenever possible, I'll use analogies to things you already know about and, for now, I'll give you just enough technical detail to get a feel for what's really going on inside your computer. You'll learn a bit more of the technical details in the next chapter.

The Brain of the Computer

At the core of every computer is a device roughly the size of a large postage stamp. This device, known as the *central processing unit*, or CPU for short, is the "brain" of the computer, the part that reads and executes program instructions, performs calculations, and makes decisions. The CPU is responsible for storing and retrieving information on disks and other storage media. It also handles moving information from one part of the computer to another. For this reason, some people compare it to a central switching station or control tower that directs the flow of traffic throughout the computer system.

In personal computers, the CPU (also known as the *microprocessor*) is comprised of a single *integrated circuit.* An integrated circuit, or IC, is a matrix of transistors and other electrical circuits embedded in a small slice of silicon. (*Transistors* are essentially microscopic electronic switches: tiny devices that can be turned on and off.) Like the dozens of other integrated circuits that inhabit your computer, from the outside, a CPU chip looks something like a square ceramic bug with little metal legs. These "legs" are designed to fasten the chip to a fiberglass circuit board which sits inside your computer, and to carry electrical impulses into and out of the chip. Inside the ceramic case is the chip itself, a slice of silicon about the size of a fingernail. At first glance, it's hard to imagine how this tiny device can run your entire computer. But under a microscope, the slice of silicon reveals an electronic maze so complex it resembles an aerial photograph of a city, complete with hundreds of intersecting streets and hundreds of thousands of minuscule houses. Most of the "houses" are transistors, and there are usually somewhere between one hundred thousand and three million of them on a single CPU chip.

The type of CPU that a computer contains determines its processing power-how fast it can execute various instructions. These days, most CPUs can execute on the order of millions of instructions per second. The type of CPU also determines the precise repertoire of instructions the computer understands and therefore which programs it can run.

The System Unit

The CPU resides inside a box known as the system unit, along with various support devices and tools for storing information. (You will learn about these other residents of the system unit later in this chapter.) For now, just think of the system unit as a container for the CPU.

The system unit case-that is, the metal box itself-can either be wider than it is tall, in which case it usually sits on top of your desk, often underneath the screen, or it can be taller than it is wide, in which case it generally sits underneath your desk and is referred to as a *tower case.*

In the PC world in particular, people often categorize computers by the model of CPU chip they contain, saying things like "I have a 386" or "My computer is a 486." The CPU chips currently used in PCs include the 8088 at the low end, 80286, 80386, 80486, and at the high end, the Pentium (the generation after the 486). People usually omit the first two digits when referring to 80286, 80386, and 80486 chips. So if someone tells you that they have a 386 computer, they mean an IBM-type computer with an 80386 chip. (You can tell it's an IBM computer or clone because they're the only computers that use 80386 chips.) The CPU chips used in Macintosh computers are, from slowest to fastest, the 68000, 68020, 68030, and 68040.

Within each class of CPU, speed is measured in terms of the cycle time at which the computer was designed to operate. All computers have built-in clocks that help regulate the flow of information from one part of the computer to another, rather like a metronome. Each pulse of this clock is known as a cycle, and a CPU can perform, at maximum, one operation per cycle.

Every CPU is designed to work with a clock that "ticks" at a particular rate. A CPU may be designed to run at 16 megahertz (MHz) or 33 megahertz, for example, meaning 16 or 33 million cycles per second. Bear in mind, however, that this measurement is relative to the class of processor. In the PC world, a 25MHz 386 is faster than a 16MHz 386, for example, but a 25MHz 486 CPU runs much faster than a 25MHz 386 CPU.

The other parts of the computer system-that is, the parts outside the system unit-are primarily used as a means of communicating with the CPU-of sending in instructions and data and getting out information. Devices used to communicate with the CPU are known, collectively, as *input and output devices,* or simply *I/O devices*. Input devices are all those things that allow you to "talk" to your computer-to pose questions and issue commands. Output devices are what allows the computer to talk back, providing you with answers, asking you for additional information, or, at worst, informing you that it has no idea what you are talking about.

Note *You may also hear the term peripherals applied to I/O devices. Technically the term peripheral means everything outside the CPU (including I/O devices).*

In personal computers, the most common input device is the keyboard (the part that looks like a typewriter). The second most common input device is a mouse. The mouse is a handheld pointing device that allows you to point to words or objects on the computer screen. The mouse sits on your desktop or sometimes on a rubber pad, called a *mouse pad*, that allows it to move more easily than on the bare desktop. Moving the mouse forward and back, or left and right, causes an arrow on the screen (known as the *mouse pointer*) to move as well. And pressing the buttons on the mouse (called *pressing* or *clicking*, depending on how fast you do it) lets you make selections on your screen. All Macintosh computer systems include a mouse, but the mouse is optional on PCs.

There are also lots of other input devices that you can use to communicate with your computers, including

>A trackball, which is a pointing device that resembles a ball nestled in a square cradle, and serves as an alternative to a mouse.

>A scanner, which allows you to copy an image (such as a photograph, a drawing, or a page of text) into your computer, translating it into a form that the computer can store and manipulate.

>A joy stick, which lets you manipulate the various people, creatures, and machines that populate computer games.

The most common output device is the display screen, which the computer uses to display instructions, ask questions, and present information. Computer screens go by many names, including monitor, VDT (video display terminal), and CRT (for cathode ray tube, the technology used in most desktop computer screens). (In laptop computers and their newer, smaller cousins the *notebooks*, both the screen and the keyboard are often built into the system unit itself.)

Almost all computer systems also include a printer for generating paper copies of your data. Like monitors, printers come in many shapes and sizes, and generate output ranging from the old grainy-looking computer printout to color printouts that rival the clarity of offset printing.

Finally, there is one type of device-called a modem-that serves as both input and output device. A *modem* is a gadget that allows computers to communicate with each other over phone lines. You can use modems both to send data and messages to your friends and coworkers who have computers, to tap into electronic information services such as Prodigy and CompuServe. Some modems even have fax capabilities built-in, allowing you to send and receive faxes via your computer.

Note *Some modems are separate devices that plug into the system unit. Others reside inside the system unit. You'll learn more about monitors, printers, and modems in Chapters 9 and 10.*

Storing Information

Now you know a little about the CPU of the computer and the devices that you use to communicate with that "brain." There is still one very large gap in our image of a computer system, however. There is no storage space.

Although the CPU is terrific at manipulating data and following instructions, it has almost no capacity for storing information. (Think of it as a brilliant but extremely absentminded professor.) In order to function, your computer therefore needs a place to store both programs (the instructions that tell the CPU what to do) and data. You need, in other words, the electronic equivalent of a closet or filing cabinet.

Note *In computerese, the term data refers to whatever type of information you are trying to manipulate. Data therefore includes far more than numbers; it includes any information that you type or otherwise input into the computer. You can also think of data as the raw material that is processed or manipulated by application programs. If you are using a word processing program, data means the document (letter, memo, poem, novel, legal brief, whatever) you are typing and/or editing. If you are working with a database program, it may be a set of names and addresses you are adding to your company mailing list.*

In most computers, the primary storage places are disks-flat, circular wafers that resemble undersized phonograph records. (You may be used to thinking of disks as square because they are always housed inside square plastic jackets. But the disks themselves are round.)

Like phonograph records or compact disks, they store information that can be "played" by devices specifically designed for that purpose. The device that "plays" computer disks is known as a *disk drive*. It is in several respects the equivalent of a turntable or CD player. Like turntables or CD players, disk drives have components designed to access the information on a specific area of the disk. These parts are called *read/write heads* and are equivalent to the phonograph needle on a turntable or laser in a CD player. Like turntables, disk drives turn around, thereby spinning the disk so that different parts of the surface pass underneath the read/write heads (just as records spin underneath the needle). Most disk drives have at least two read/write heads-one for each side.

Unlike record players or CD players, however, disk drives can record new information on disks as well as play existing information. (In this sense, they're more like cassette tapes than records.) In computer terminology, the process of playing a disk is called *reading* and the process of recording onto a disk is called *writing*. (Hence the term read/write head.)

Computer disks come in two basic types: floppy and hard.

>Floppy disks generally hold less information and are slower than hard disks. They can also be removed from their disk drives. In other words, you can "play" different floppy disks in the same drive by removing one and inserting another. The word floppy refers to the disk itself, which is a very thin, round piece of plastic on which information is magnetically recorded (much as music is recorded on the surface of plastic cassette tapes). This decidedly floppy disk is enclosed inside a sturdier, unfloppy plastic jacket to protect it from dust, scratches, liquid, and the oils on your fingers-all of which can erase or scramble the information recorded on the disk's surface. The disks used in personal computers are usually either 3½ or 5¼ inches in diameter.

>Hard disks hold more information and spin faster than floppies (about twenty times faster). They are also permanently enclosed within their disk drives. You can think of hard disks as records that are encased inside their record players so that the record and the record player function as a single unit. Contrary to what you may think, hard disks are not always larger than floppies; they're simply capable of packing information more tightly, and therefore can store more data in the same amount of space. Most hard drives contain multiple disks, often called platters, which are stacked vertically inside the drive. Typically, each disk has its own pair of read/write heads. Since you never remove hard disks, hard-disk drives do not contain doors or slots, as do their floppy counterparts. This means that the drive itself is completely invisible (and sometimes hard to locate) from outside the system unit. In most cases, the hard drive is adjacent to the floppy drive.

These days, most personal computers have one hard-disk drive and either one or two floppy-disk drives. In general, you'll use hard disks as the primary repository of data and programs-the place you store the information that you work with day to day. You'll use floppy drives mainly as a means of getting information into and out of your computer, by transferring information to and from floppy disks. Floppy drives are, in this sense, like doors to the outside world. In particular, you use floppy-disk drives to:

>Install new programs, by copying them from floppy disks to your hard disk.

>Make extra copies of programs or data for safekeeping by copying from the hard disk to floppies. This is known as making backups. If you are working on the great American novel, for example, you will keep your main working copy on the hard disk but keep an extra copy on a floppy disk, in case there is a mechanical problem with the hard-disk drive or you accidentally erase the original. If you want to be completely safe, you might even keep this duplicate copy in a safe deposit box or a fireproof safe.

>Archive data that you don't use regularly (and therefore don't want taking up space on your hard disk) but that you don't want to discard altogether.

>Transfer data from one computer to another, by copying information from one computer's hard disk to a set of floppies, taking the floppies over to the other computer, and copying from those floppies onto the hard disk.

There are several things that you need to know to work with floppy disks, including how to determine which type will work in your disk drive, how to prepare them for use, how to insert and remove them, and how to take care of them. (You'll learn to do all of these in the next chapter.)

Although disks and their disk drives are by far the most common means of storing data and programs, they are not the only ones. Other storage technologies currently in use include tape drives, CD-ROM drives, magneto-optical drives, and Bernoulli boxes.

Memory: The Electronic Desktop

Given what you've learned so far, you might assume that when you run a program, the CPU fetches instructions from the disk one at a time and executes them, returning to the disk drive every time it finishes a single step. If this were actually the way computers worked, they would be so slow as to be unusable.

Left to their own devices, most personal computer CPUs are capable of executing between one million and one hundred million instructions per second. But because it is mechanical-that is, composed of moving parts-the disk drive cannot deliver program instructions anywhere near that fast. Reading an instruction from the disk involves both rotating the disk so that the proper section is below one of the read/write heads and then moving the head closer to or farther from the center of the disk until it is positioned directly above the spot where the instruction is recorded. Even on a hard-disk drive, this process generally takes between 9 and 25 milliseconds (millionths of a second).

Now if the CPU can execute millions of instructions per second and the disk drive can only deliver, say, one hundred thousand instructions per second, you have the equivalent of an assembly line in which one person on the line (the CPU) is moving somewhere between 10 and 100 times faster than the previous one (the disk drive). If this were really the way your computer worked, the speed of the CPU would be wasted while it waited for the disk drive to deliver the next instruction.

For the computer to function efficiently, it therefore needs some repository of information that is capable of keeping pace with the CPU. This extra piece is called *random access memory*, usually referred to as RAM or simply memory for short.

Physically, RAM consists of a set of separate integrated circuits (each of which looks something like a small CPU chip) which are often mounted on fiberglass boards; in practice, however, memory is treated as a single, contiguous set of storage bins. One useful way to envision memory is as a set of mailboxes, like those inside a post office. Each mailbox holds a single character, and the entire collection of boxes is numbered sequentially. (In computer jargon, the mailboxes are called bytes and their numbers are known as *memory addresses*.)

Like the CPU chip, memory chips store and transmit information electronically. Sending an instruction from memory to the CPU is therefore a simple matter of transmitting electrical impulses. There is no waiting for a disk to spin or a read/write head to move to the proper position.

Because the CPU can move information in and out of memory so quickly, it uses memory as a kind of electronic desktop-the place it stores whatever it is working on this instant or plans to work with shortly. When you tell your computer that you want to use a particular program, for example, the first thing it does is find the program on your hard disk and copy it into memory. This process is known as *loading* a program. This gets the comparatively slow process of reading instructions from disk over with at the start. Once the entire program has been loaded, the CPU can quickly read instructions from memory as needed.

Although you can compare placing a program in memory to moving it from a file cabinet (the disk) to your desktop (memory), there is also one important difference. Placing something on your desktop entails removing it from its usual storage place in the file cabinet. In contrast, when you place a program in memory, you do not remove it from anywhere. Instead, you make the equivalent of a photocopy of the program, and place the copy in memory. The original copy of the program stays on the disk, ready to load again whenever you need it.

The first program that is loaded into memory in every work session is the operating system. In fact, just about the first thing your CPU does when you turn on your computer is hunt for and load the operating system program. This program then remains in memory until you turn off your computer. When you load application programs, they always share the electronic desktop with the operating system, and, in fact, application programs need to have an operating system around in order to function.

Saving Data

Programs are not the only thing that the CPU places in memory. It stores data there as well. As shown in the figure "Data Stored in Memory," each character that you enter, including any spaces, occupies a single "mailbox" of storage space.

There is one fundamental problem with housing data in memory, however: As soon as your computer is turned off, the contents of memory are erased. This means that if you accidentally kick your computer's power cord in the midst of typing a letter, for example, everything you have typed is lost. It also means that you cannot rely on memory if you want to return to a piece of work the next day or next week, since you will undoubtedly turn off your computer in between. Data is also erased from memory whenever you leave an application program, because the CPU assumes you'll need to use the space for the next task you choose to tackle. If you are using a spreadsheet program, for example, the CPU erases the spreadsheet you were using from memory as soon as you exit that program, just as you might clear your desk when you finish a particular project.

In order to protect your work and to store it for use in future work sessions, you need to copy it from memory to a more secure storage place: namely, a disk. This process is known as *saving* your data. Don't worry about how, exactly, you tell your computer to save data; the procedure may vary from one application program to another or, at minimum, from operating system to operating system. For now, just focus on when you need to save and why.

In general, you should save your data whenever:

>*You are done working with it and are ready to start on another project (if you finish typing one letter and want to start another one, for example)*

>*You are ready to leave an application program*

For your convenience, most programs automatically ask if you want to save when you give the command to exit or to close a document. You should also save to disk whenever you have been working for a while and want to protect the work you have done so far. This protects your data against power failures, kicked power cords, or drastic mistakes.

How often you should save depends on what you are doing and how easy it would be to do over. A good rule of thumb is that you should save your work every time you'd be unhappy if you had to do it over again. For some people, this will be once an hour; for others, it will be every two minutes. Bear in mind that when you save something, you are not removing it from memory. You are simply making a copy and then storing that copy on disk. You can then continue modifying the original if you like.

Note *Occasionally, you will enter some data that you have no need or desire to save. You may use a word processing program to type a short letter that you need to print but don't need to store for future reference. Or you may load your spreadsheet program to perform a few calculations and have no need to save the results. In such cases, simply leave the program without saving your data. (Most programs also have a command for throwing out the data in memory without leaving the program, in case you want to discard your work and start again.)*

Once something is stored on disk, you can always copy it back into memory when you want to use it again. This is known as *retrieving* data. (Again, don't worry about the exact command for doing this, since it may vary from one program to the next.) As soon as the data is copied back into memory, the data reappears on your screen and you can modify it if you like. In general, any data that appears on screen while you're using an application program is currently in memory, although not all the data in memory may fit on your screen at one time.

A Quick Review

So far, we have talked about three different types of information being stored in memory: the operating system, application programs, and data. Just to review:

>*When you first turn on your computer, the CPU automatically copies the operating system program from the disk into memory.*

>*When you tell your computer that you want to run an application program, the CPU copies that program from disk into memory (alongside the operating system).*

>*When you type in new data while working in an applications program or retrieve data from disk, the data is placed in memory as well, alongside the applications program and the operating system. The data remains in memory until you either issue a command to close the file (remove it from memory) or leave the application program. Whenever you leave an application program, both the program itself and any data that goes with it are removed from memory. The operating system stays put until you turn off your computer.*

You may find it helpful to think of these three different types of information as a set of layers, each of which depends on the one below. The operating system is at the bottom layer, followed by the applications program, followed by the data. If you unload any one layer, the layers above are erased as well. If you unload the operating system-for example, by turning off or restarting your system-memory is completely erased. If you unload an application program from memory, by issuing that program's exit or quit command, both the program and the data you were working on inside that program are erased. If you erase just the top layer-the data you are working with-the application/program and operating system remain loaded and ready to use.

What's So Random about RAM?

As mentioned, the type of memory we have been discussing-that is, the memory used to temporarily house programs and data-is often referred to as *random access memory* or RAM for short. In order to understand where this name comes from, you need to know more about how information is stored in memory.

As mentioned, the CPU treats memory as a set of numbered storage bins, rather like a collection of mailboxes, each one of which holds a single character. In older computers, the CPU had to access the mailboxes (bytes) in numerical order, starting from the first mailbox and moving forward until it reached the one that actually contained the desired information. This is known as *sequential access.* With the development of random access memory, the CPU can go directly to whichever mailbox it is interested in.

You can conceptualize the difference between random access memory and this older type of sequential access memory by comparing records or CDs to cassette tapes. If you want to listen to the fifth song on a cassette tape (sequential access), you have to start at the beginning of the tape and move past the first four songs, even if you fast forward the tape. With a record or CD (random access), you can go directly to song five.

Note *Disks are random access devices, too. Rather than starting from the outside of the disk and reading inward, or the inside and reading outward, the read/write head can jump directly to the spot where the desired data is stored.*

There is actually a second type of memory used in personal computers, in addition to RAM. This second type of memory is named *read-only memory*, or ROM (rhymes with Tom). Unlike RAM chips, ROM chips have software (program instructions) permanently etched into their circuitry. For this reason, ROM is often referred to as *firmware*-since it's kind of halfway in between hardware and software.

Unfortunately, the terminology here is a bit misleading, since both RAM and ROM allow random access. If the point is to distinguish RAM from ROM, then RAM would more properly be called read/write memory, meaning that you can not only retrieve (read) information from RAM; you can also record (write) information to it. In contrast, with read-only memory, instructions are frozen into the circuitry. The feature that sets RAM apart from ROM is its changeability: the fact that you can alter its contents at will.

The other difference between RAM and ROM is how long their memories last. RAM is short-term memory; it forgets everything it knows as soon as you turn off your computer. ROM is long-term memory; it remembers everything it has ever known as long as it lives. It's the elephant of the memory kingdom.

In personal computers, ROM is generally used to store some part of the operating system. In IBM-type computers, only a small part of the operating system is stored in ROM-just enough to get the hardware up and running and to tell the CPU how to locate and load the rest of the operating system from disk. In Macintosh computers, much more of the operating system is stored in ROM.

Note *Since ROM is not changeable, you will never have to deal with it yourself. It's just another part of your computer to know about, even if you never need to see it, touch it, or think about it much at all.*

What's in a Byte: How Memory and Disks Are Measured

As mentioned, the term *byte* means the amount of space required to represent a single character-a letter, a number, or even a space. (In our mailboxes analogy, it's a single mailbox.) This term is used regardless of whether you're talking about space in memory, on a disk, or on any other storage medium. Since many, many bytes are often required to accommodate an entire word processing document, spreadsheet, database, or program, computerese includes terms for several larger units of measurement.

>A kilobyte (often abbreviated as simply K) is 1,024 bytes. To a computer's way of thinking, 1,024 is a nice round number. (Computers "think" in units of two, and 1,024 is 2 to the 10th power.) To us, however, it's a little unwieldy so most people think of a kilobyte as "around 1,000" bytes. So when someone tells you that they have a computer with 640K of RAM, they mean a computer that has 640 kilobytes or approximately 640,000 characters worth of random access memory. A floppy disk that holds 360K of data can accommodate a little more than 360,000 characters.

>The term megabyte (abbreviated as MB or simply M) means a kilobyte squared (1,024 times 1,024), or approximately one million bytes.

>The term gigabyte means a kilobyte to the third power (1,024 times 1,024 times 1,024), or approximately one billion bytes.

While a few years ago, most computers had less than a megabyte of RAM, these days, many have as much as 4, 8, or even 16MB. Hard disks typically hold between 20MB and 600MB, although there are some that store gigabytes. Just to give you a measuring stick, a typical printed page of text, using single spacing, contains 2,500 to 3,000 characters. Therefore, 1MB holds close to 400 pages of single-spaced text.

So why do you care how much memory and disk space your computer has? Because it determines what kinds of work you can do. The amount of memory in your computer dictates which programs you can run. (Many Windows programs, for example, run best with 4MB or more of memory, and you cannot run them at all on computers with less than two megabytes.) The size of your hard disk is important because it defines how many programs and how much data you can store on your computer at once. In general, you will want enough room on your hard disk to accommodate all the programs and data that you work with regularly. Otherwise, you'll waste time copying data or programs to and from floppies. These days, each program you install requires from 1 to 15MB of disk space, not including room for data. The easiest way to find out if you have enough memory and/or disk space for a particular program is to read the program's packaging or user's manual.

You'll learn how to find out how much room you have in memory and on your disk when you learn about operating systems in Chapters 5, 6, and 7.

Note *Bear in mind that you can almost always add memory to your computer, by buying additional memory chips and having them installed. (If you are both brave and unintimidated by computer manuals, you may even be able to install them yourself.)*

Inside the System Unit

Now that you know what the CPU, memory chips, and disk drives do, you're ready to learn about where they reside and how they're connected. In most computer systems, all three of these components are housed inside the system unit. (Some computer systems have an external disk drive instead of or in addition to the ones inside the system unit.)

The centerpiece of the system unit is a printed circuit board, known as the *motherboard*, which holds the CPU chip and its support circuitry. (You may also hear the motherboard referred to as the system board or a planar board.) The motherboard generally lies face up at the bottom of the system unit.

The motherboard contains several other types of chips, in addition to the CPU, that help the CPU perform its job. These support chips include:

>The clock chip, which serves as the computer's metronome, setting the pace at which the various components function

>One or more ROM chips containing some part of the operating system software

>In some computers, a math coprocessor chip, which assists the CPU in performing

specific types of mathematical operations

Finally, the motherboard usually includes slots or sockets for memory chips. In most cases, memory comes in the form of small, plug-in boards call *SIMMs* (short for single in-line memory modules), each of which includes eight or nine memory chips.

Aside from the motherboard, the system unit includes disk drives (usually one or two floppy drives and one hard drive) and a power supply. The power supply brings in power from the wall socket and supplies it to the motherboard. It also contains your computer's on/off switch and a place to attach the power cord that connects the system unit to a power outlet. The power supply unit usually contains a fan, to prevent the various chips from overheating. If your system includes a tape drive, it probably resides in the system unit as well.

Most computers also contain additional circuit boards, commonly known as *expansions boards*, which fit into slots on the motherboard. (The slots themselves are known as *expansion slots*; think of them as parking spaces for circuit boards.) Expansion boards sit at the back of the system unit at a right angle to the motherboard itself. The purpose of most expansion boards is to allow an I/O (input/output) device-like a display monitor or a scanner-to communicate with the CPU.

The advantage of this design-a motherboard containing all the standard circuitry of the computer and a set of expansion slots that allow you to plug in additional circuitry as needed-is that it allows you to customize your system. Two people can buy essentially the same computer but add on very different sets of peripherals. This design also allows you to easily add new parts to your computer as your needs change or as new forms of computer paraphernalia are invented.

Expansion boards that are designed to serve as intermediaries between the CPU and some device outside the system unit have *ports* on one end. Ports are sockets that protrude from the back of the system unit. You can think of them as places where you can "dock" various external devices, plugging them into a circuit board that, in turn, connects them to the CPU. (Expansion boards that are designed for components inside the system unit-such as disk drives-do not include ports.)

Finally, all motherboards contain a bus: a set of circuitry designed to carry data and instructions back and forth between various devices on the board itself. You might think of the bus as a collection of elaborate, high-speed conveyor belts. The bus not only carries data and instructions back and forth between the CPU and memory (both RAM and ROM), it also connects the CPU and memory to any expansion boards that are plugged into the motherboard.

Chapter3
Your Computer's Filing System

-What Are Files?
-File Name Rules
-Saving, Retrieving, and Resaving Files
-File Formats
-Organizing a Hard Disk: Folders/Subdirectories

-Working with Floppies
-The Care and Feeding of Hard Disks
-Computer Viruses

In order to use your computer effectively, you need to understand a bit about its filing system, namely, how it stores and organizes information on disks. This chapter begins by discussing files, the repositories of programs and data on disks and other long-term storage media. You will learn about rules for file names, about file formats used by various application programs, and about the organization of files into groups known as folders or subdirectories. The second half of the chapter delves into detail about the disks themselves-the medium on which files are generally stored. By the time you're through, you will know how to choose the right type of disks for your floppy-disk drive(s), protect floppies from accidental damage, prepare new floppy disks for use, care for your hard disk, and ward off computer viruses.

What Are Files?

All data on disks is stored in files. A *file* is simply a named collection of information stored on a disk. There are two basic types of files: *program files*, which contain instructions to your computer, and *data files*, which contain data that you enter through an application program. (In the Macintosh world, data files are generally called documents.)

Unless you delve into programming at some point, most of the files you create will be data files. Every time you enter data in an application program-be it text, numbers, pictures, or anything else-and then save it for the first time, you are creating a new data file. And since all files, by definition, have names, the first thing that happens when you issue the command to save is that the program asks you to assign a file name.

Note *In some programs, data is saved to disk automatically. Many database programs work this way. When you first set up a database, you create a file to hold your data. Then as you enter the information on each customer (or product or order placed or whatever you are storing in your database), that information is automatically saved to disk as soon as you move to the next customer (or product or order).*

File Name Rules

The rules for file names are different on Macs and PCs. On Macintosh computers, file names are all one piece and can contain up to 31 characters. In DOS, file names consist of two parts: a primary file name (often called simply the file name), which can be up to eight characters long, and an optional suffix, called an *extension,* which can be up to three characters long. Macintosh file names can include letters, numbers, spaces, and any punctuation other than a colon (:). DOS file names (and extensions) can include letters, numbers, and the following punctuation characters:

` ~ ! @ # $ % ^ & ( ) _ - { } '

Since it's hard to remember which punctuation marks are acceptable, it's often easiest to just stick with letters and numbers. DOS file names *cannot* include spaces.

Note *Because Windows works in tandem with DOS, DOS file name rules apply in the Windows world as well.*

If you elect to add an extension to a DOS file name, you must type a period between the two parts of the file name whenever you refer to the file. If a file has a first name of LETTER and an extension (last name) of DOC, for example, the entire name would be LETTER.DOC.

Sometimes you don't have much choice in the matter of file extensions. Many application programs automatically add a preset extension for you whenever you create a file. If you create a letter in the Ami Pro word processing program, for example, every time you create a new file, Ami Pro assigns it an extension of .SAM.

Note *On Macintosh computers, file names that are entered in different mixtures of upper- and lowercase are considered different file names. You can therefore have files named Memo, memo, and MEMO stored on the same disk. DOS, in contrast, ignores the case of letters, considering Memo, memo, and MEMO to be identical. (DOS actually stores all file names in uppercase, regardless of how you enter them, but it lets you refer to files using any mixture of upper- and lowercase that you like. In other words, DOS may "think" of the file as MEMO, but it understands what you mean if you tell it to copy Memo to another disk.)*

In general, every file on a particular disk must have a unique name. (We'll explain the qualifications to this rule a little later in this chapter, after you learn about folders/subdirectories.) This way, when you tell your computer to find the file LETTER.DOC and copy it into memory, it knows exactly which file you mean. It doesn't need to determine *which* LETTER.DOC.

You must keep this law of unique file names in mind when copying files. If you copy a file named LETTER.DOC from disk 1 to disk 2, and disk 2 already has a file named LETTER.DOC, the old version of LETTER.DOC is completely and permanently replaced by the new one. If you are using a Macintosh or Windows, you will be asked to confirm that you want to replace the existing file. If you are working directly with DOS, however, you receive no warning at all. The file is simply replaced, and there is nothing you can do to rescue it afterwards.

Saving, Retrieving, and Resaving Files

As you learned in Chapter 2, once something is stored on disk, you can always copy it back into memory when you want to use it again (just as you can fetch a particular document from your file cabinet). This is known as *retrieving* data or *opening* a file.

Just as when you load a program from disk to memory, when you retrieve data, the original copy of that data remains in place and intact on the disk. If you then change the copy in memory, you end up with two different versions: an older version on disk and a newer version in memory. The same situation occurs when you save a new document to disk but continue working on it. You might, for example, get halfway through writing a letter and then save your data to disk. You then have two separate and independent versions of the same letter: one (the older one) on disk and another (the current one) in memory.This has a couple of implications:

>First, if you like the new version better than the old one, you must remember to save it before you leave the application program. Otherwise you'll have only the old version of the document (the one on disk) and your changes will be lost.

>Second, if you decide that you prefer the older version, you can close the document (remove it from memory) without saving it. When you do so, the version of the document currently in memory is erased. You can then retrieve the old version (the one on disk) and start amending it again. This can be extremely convenient when you completely bungle an edit and want to start all over.

Whenever you decide to save a file that has already been saved once, you need to decide whether to use the same file name as last time or a new file name. As mentioned, you cannot generally store two files with the same file name on the same disk. In fact, if you copy a file to a particular disk and the disk already contains a file of that name, the new one replaces the old one. The same issue arises when you resave a file. Suppose you create a budget in your spreadsheet program, for example. Halfway through the process, you save your data, then you revise it, and save it again. If you resave it under the same name that you used the first time, the new version will replace the old one on disk. Most of the time, this is exactly what you want. If you want to retain the old version of the file as well as the newly revised one, however, you must assign a different name to the new version.

In Windows and in the Macintosh operating system, there are separate commands for resaving a document under its existing name and for saving it under a new name. (You use the Save As option on the File menu rather than the Save command when you want to save something under a new name.) In some DOS programs, there's only one save command (often called File Save) and you need to remember that saving a file under the same name means replacing the old version. Many application programs also ask for confirmation when you try to save a file under an existing file name.

File Formats

Most application programs have their own unique format for storing data, a format that only makes sense to that one program. For example, the format in which the Lotus 1-2-3 spreadsheet program stores data is not the same as the format used by the Excel spreadsheet. The format that the Word for Windows word processing program uses is not the same as the format that WordPerfect uses.

In general, if you want to see what's inside a particular data file, you need to look at the file from inside the program in which it was created. For example, if you want to see what's inside an Excel spreadsheet file, you need to look at it from within the Excel program. If you try to look at it from within a word processing program or even from within another spreadsheet program, you will probably just see a lot of nonsense characters on your screen.

Special codes that tell the program how to arrange and format the data distinguish the file format used by one program from that used by another. Each word processing program has its own code for representing italics or page breaks, for example.

Many programs have commands for importing and exporting data in the formats used by other programs. Word for Windows can import WordPerfect files, for example, translating all the WordPerfect formatting codes to their Word for Windows equivalents. In the DOS world, some programs determine which file format is being used by looking at the file extension, and translate the data into the appropriate format automatically. (This saves you the trouble of telling the program about the file format or using a special import command.)

Occasionally, a program does not have any tools at all for interpreting the special codes used by another program. In this case, you can save the file in a generic format, without any of the special formatting codes specific to one particular program. The most commonly used generic format is one known as ASCII in the PC world and text-only in the Mac world. (ASCII stands for American Standard Code for Information Interchange, and is a set of standardized codes used to represent all the characters you can produce on a typewriter, plus a few others.) When you save a file in ASCII or text-only format, your program strips out any proprietary codes-that is, codes that only it knows how to read-leaving you with plain-vanilla text; any special attributes such as underlining or unusual typefaces will be lost in this translation process. Once you have saved a file in ASCII or text-only format, you can open it in almost any other program, although you may need to specify the file format when you open the file or issue a special import command. Consult your program documentation for information on saving and using ASCII text files.

Organizing a Hard Disk: Folders/Subdirectories

Hard disks often hold thousands of files. Rather than piling this entire collection of files in a single heap, most people organize their files into groups. These groups are generally known as *folders* in the Mac world and as *subdirectories* in the PC world. You can think of folders/subdirectories as manila folders in a file drawer, each one of which can hold several individual files.

Just as when you set up a manual filing system, when you organize a hard disk you decide for yourself how many folders you need and what to store in each. Often you will place each application program and its associated files in a separate folder (folder/subdirectory). For example, you might have one for your spreadsheet program, another for your word processing program, and a third for your accounting program. Many people prefer to create two separate folders for each application-one to hold the program itself and another to hold the data files created in that program. You might, for example, create one folder to hold your word processing program and another to hold your documents. This makes it easier to find and manipulate the data files. (In most operating systems, it is fairly easy to copy to a floppy disk all the files in a particular folder. Storing your data files (and nothing else) in one folder therefore makes it easy to create a backup copy of your data.)

If you have many different types of data, you might also create separate folders for each type. You might create one folder for business correspondence and another for letters to friends, or one for correspondence, another for reports, and a third for invoices.

You can create additional subgroupings if you like. For example, you might create a folder for word processing that contains your word processing program and then, within that folder, create another folder for the documents you have created and saved.

In short, the filing system you create on a disk is as individual as the one you create in a filing cabinet.

The process of actually creating and manipulating folders/subdirectories varies a bit from one operating system to the next, and is therefore covered in later chapters. For now, just be aware that creating a new folder/subdirectory always involves entering a unique name. The rules for naming folders/subdirectories are the same as the rules for naming individual files: On Macs, you get 31 characters to work with, including spaces; on PCs, you get 8 characters and an optional 3-character extension, although the extension is rarely used for subdirectory names.

Now that you know about folders/subdirectories, we can modify an earlier rule: Files in the same subdirectory/folder must have unique names. You can, however, have files of the same name in two different subdirectories/folders on the same disk.

Working with Floppies

As you learned in Chapter 2, there are two basic types of disks: floppy disks and hard disks. Hard disks serve as the primary repository of data. Floppy disks are mainly used for copying data to and from your hard disk, acting as a kind of gateway between your computer and the outside world.

Floppy disks come in two sizes: 5¼ inch and 3½ inch. Which size of disk you use depends on the type of floppy-disk drive you have in your computer. The floppy-disk drives in Macintosh computers always use 3½-inch disks. On PCs, drives designed to read 5¼-inch disks used to be the norm, but 3½-inch drives are becoming increasingly common and the 5¼-inch variety is gradually becoming obsolete. Some PCs have two floppy-disk drives, in which case one may take 5¼-inch disks and one 3½-inch disks.

As described in the last chapter, the part of the disk that you see and touch is the outside cover-the disk's jacket. Inside this jacket is a flimsy piece of plastic on which data is magnetically recorded. The reason that disks have jackets is that they are extremely sensitive: Scratches and spilled liquids are enough to permanently destroy them, and the oils on your fingers are enough to wreck the magnetic patterns used to record data on their surface.

On 5¼-inch floppies, the jacket contains an oblong read-write hole through which the surface of the disk itself is exposed. When you place the disk in the drive, the drive's read/write head (the part that reads and records data) is positioned right above this hole. When you handle 5¼-inch disks, you therefore need to be careful not to touch this exposed portion of the disk. To be safe, hold the disk jacket by the label. You should also avoid writing on a disk label with a ballpoint pen, since the pressure of the point can damage your data.

You don't need to worry about accidentally touching or scratching the surface of 3½-

inch disks, because the disk remains protected until it is actually inserted into the drive. When you insert a 3½-inch disk into a drive, the metal shutter is pushed to the side, exposing the disk's surface so that the read/write heads can read and record information.

With both sizes of disks, you do need to be wary of magnets. Any exposure to magnets can scramble or erase the information recorded on the disk.

Protecting your disk from magnets may take more vigilance than you realize. Magnets lurk in many unsuspected places, including many paper clip holders (that's why the clips stick to the rim of the holder) and various other office accessories, including some document holders. Since the coil for your telephone becomes magnetized every time your phone rings, it's wise to keep disks at least a few inches away from the phone. In addition, metal detectors in airports and government buildings sometimes use magnets to detect the presence of metal in your luggage, so you may want to remove disks from your luggage and show them to the attendant, just as you do with film.

You should also avoid storing both sizes of floppy disks in extreme heat. (Do not, for example, leave a disk on your dashboard on a hot summer day.)

In case you are interested, you need to keep disks away from magnets because of the way in which information is recorded on the surface of the disk. When your computer writes (records) information on a disk, electricity is sent through a coil of wire that is wrapped around a piece of iron within the read/write head. Now as you may remember from grade school science classes, whenever you happen to send electricity through a wire that is wrapped around a piece of iron, the piece of iron becomes magnetized. In effect, the read/write head is temporarily turned into a magnet. This magnet is then used to magnetize tiny particules of iron oxide (in a word, rust) on the disk's surface. As they become magnetized, different sets of particles are aligned in one of two configurations: one configuration represents ones, and the other, zeros. (Remember that computers store all types of information as numbers consisting entirely of zeros and ones.) Now since one magnet-a magnetized read/write head-is used to record the information on a disk, any other magnet has the capacity to realign the iron oxide particles and thereby scramble your computer's record of your data.

WARNING *When the read/write head is reading or writing to a disk, a small LCD (liquid crystal display) on the front of the drive lights up. Do not remove the disk until the this light goes off, indicating that the process is complete.*

Occasionally, you may want to guard the files on a disk against accidental erasure or overwriting. (You may find this particularly useful for disks that contain programs.) You can do this by *write-protecting* or *locking* the disk-making it impossible for anyone to copy new files onto the disk or erase files already on it. (The term "write-protect" is generally used in the PC world; "lock" is used in the world of Macs.)

The procedure for write-protecting a disk depends on the disk type. 5¼-inch disks contain a notch in the upper-right corner. To write-protect such disks, you simply put something over this hole. When you buy disks, the box includes write-protect tabs (small sticky labels) designed for this purpose. If you don't have any write-protect tabs, just use part of a gummed label to cover up the notch.

3½-inch disks have a tiny latch in the upper-right corner, which is usually closed. To write-protect such disks, turn the disk over and slide the latch downward so that a small window appears in the corner of the disk.

Note *Some program disks are permanently write-protected-that is, they do not have a write-protect notch or latch. This prevents you from accidentally erasing the disk; it also keeps you from using it for anything else.*

When you first buy floppy disks, they are essentially blank slates. Before you can store data on them, the surface of the disk needs to be subdivided in some way, so that data can be stored in specific, easily locatable spots. Otherwise, your computer would have an awful time finding files when you needed them. The process of subdividing a disk into sections by embedding magnetic codes on the disk's surface is known as *formatting* or in the Mac world, *initializing.*

Note *Hard disks also need to be formatted before use, but they're usually formatted when you buy them. If not, find an expert to do it for you; the process is a bit too complicated for a novice.*

When you format a disk, your computer subdivides the disk in two different ways: drawing concentric circles (known as tracks) around the disk, and drawing straight lines (known as sectors) that divide the disk radially.

You may find it helpful to think of your disk as a stadium. Just as people identify stadium seats by section and row number, your computer identifies where files are "seated" by sector (section) and track (row).

Note *The process of formatting a disk is quite simple, but it may take a minute or two. (On Macs, whenever you insert an unformatted disk, a message is displayed asking if you'd like to initialize it. On PCs, formatting requires issuing a single command.) In case you're perenially short on time, you can now buy disks that are already formatted for use on PCs. Since they cost a bit more than unformatted disks, you have to decide whether you are willing to pay extra to save yourself the trouble of formatting.*

To continue with the stadium analogy, when you "seat" data on a disk, you can't always get exactly the number of seats (bytes) you'd like. Instead, you need to "purchase" seats in blocks, usually known as clusters. A cluster is the minimum amount of space used to store a file. If the cluster size for your operating system is 256 bytes, even a 1-byte file takes up 256 bytes worth of space. It cannot share its space with other small files. (Remember that a byte is a character"s-worth of storage space.) Similarly, if a file is 300 bytes in size (a little more than one cluster), it takes up 512 bytes (2 clusters) on disk. The size (in bytes) of a cluster varies from one operating system to the next, but a cluster always occupies two or more sectors on a particular track. Strictly speaking, your computer keeps track of where files are seated on a disk by storing the starting cluster number for each file. It then uses that cluster number to calculate the corresponding track and sector number.

Different disk drives are designed to read different types of disks. The physical size of the disks is only part of the story. You also have to consider how closely the drive packs data on the disk. Some disk drives, known as high-density drives, store two or more times as much data on the same size disk as other, low-density drives.

You should be able to determine the type of disk drive you have by looking at your computer manual. If you can't find the manual, try getting a list of the files on a high-density disk. (You'll learn to do this in the chapter on your operating system. Chapters 5 through 7 discuss the DOS, Windows, and Macintosh operating systems, respectively.) If your drive can read the disk, it's a high-density drive. If it can't, you'll see an error message.

Note *Most new computers have only high-density drives. The whole issue of disk density is therefore slowly becoming moot as low-density drives become less and less common.*

When you buy disks, you need to be sure to get the right type for your disk drive-not only the right size, but the right density as well. If you have a low-density drive, you should buy low-density disks. (Such disks are generally labeled as double-sided double-density disks since, unlike the disks used in some earlier computers, they can store data on both sides, twice as densely as some older-model disks.) If you have a high-density drive, you should generally buy high-density disks (usually labeled as either double-sided high-density, or simply high-density).

It is possible to use low-density disks in high-density drives. This allows you to trade floppy disks with computers that have low-density drives, even if you only have a high-density drive. However, some older high-density drives are not as well-behaved in handling low-density disks as you would like: While they read low density disks beautifully, once you save any data on the disk, you may no longer be able to read the disk in a low-density drive. (In other words, as soon as you write to the disk using the high-density drive, you can only read it using a high-density drive.) If you have this problem with your high-density disk drive, try to limit your activity with low-density disks to reading rather than writing. Low-density drives cannot read (retrieve information from) or write to (record information on) high-density disks.

The Care and Feeding of Hard Disks

Hard disks are not as vulnerable to dust and liquids as floppies are because they are sealed inside metal cases. They are, however, far from indestructible (as many discovered during the 1989 California earthquake). Knocking a computer off your desk or dropping it on the floor is an almost sure fire way to lose some data if not permanently damage the disk. Some hard disks are more sensitive than others, and may respond poorly to being repeatedly moved from desk to desk.

Hard disks can be especially sensitive to brownouts-dips in electrical power that are common in big cities in summertime. This is because hard disks are designed to spin at a particular speed. If they receive too little current, they spin slower than they're meant to, often with disastrous results.

The basic rules of thumb for caring for a hard disk are

>*Don't drop your computer.*

>*Unless you have a laptop or other computer designed for travel, don't move it any more often than you have to.*

Most hard drives spin at a rate of somewhere between 3600 and 5400 revoluations per minute, generating the equivalent of gale-force winds at the edges of disk platters. While the disk is spinning, the read/write heads on hard disks do not actually touch the surface of the disk. Instead, they hover above or below the disk's surface, at a distance of millionths of an inch. (To make it a little more concrete, the distance between the read/write heads and a spinning disk is less than half the width of a particle of smoke.) When your computer is turned off, the read/write heads do come to rest on the surface of the disk, but only within a specified parking area that is reserved for this purpose and is never used for storing information.

If your hard drive malfunctions for some reason or you drop your computer on the floor, the read/write heads may fall onto the disk, permanently damaging it. This event, which is actually quite rare in modern-day hard drives, is known as a head crash. If someone tells you that their hard disk "crashed," they probably mean it underwent a head crash.

The small but essential gap that exists between the surface of the disk and the read/write heads explains why hard disks always live inside sealed containers, safe from such hazards as smoke, dust particles, and soda pop. Since the distance between the disk and the read/write head is half the size of a smoke particle, any encounter with such a particle would be like a high-speed go-cart running into a boulder.

>*If you're relocating to another office or building, try to pack the computer in its original packing materials to cushion it during the move.*

>*Most importantly, back up your data regularly. Operate on the assumption that sooner or later, your disk will crash (probably a day or two before some crucial deadline).*

I've been representing each file as occupying a single discrete area of a disk. In reality, files are sometimes stored in segments scattered across the disk. The first part of a file may be wedged in between two other files, for example-perhaps in the space previously occupied by a file that you later deleted. Since not all of the file can fit in that space, your computer simply makes a note of where the next piece is stored (rather like the next clue in a treasure hunt). If there isn't enough room for the rest of the file in that second spot, your computer makes a note of where the third piece is stored, and so on. Files stored in this way are said to be *fragmented.*

Over time, your hard disk will accummulate many of these fragmented files. Every time you load a fragmented file into memory, your computer has to jump from one part of the hard disk to another collecting all the file's different pieces. You can improve your computer's efficiency by periodically *defragmenting* your hard disk-that is, running a special program that rearranges data so that all the parts of each file occupy contiguous clusters on the disk. (You can compare this process to a bunch of people trading seats so a group of friends can sit together.)

Computer Viruses

Even people who've never touched a computer have heard ominous tales about computer viruses. A virus is a program, generally designed by a bright but maladjusted computer nerd, that in one way or another interrupts or undermines the normal workings of your computer. Viruses work by copying themselves into legitimate files, called *hosts.* From there, they often branch out, replicating themselves in more and more files on the disk. While some viruses infect almost every file in sight, others are more picky: Some viruses only infect application programs, other infect data files, and still others invade the operating system itself.

The level of destructiveness among viruses varies widely from one program to the next. Some viruses simply display pictures or messages on the screen periodically. Others erase or destroy both programs and data. They also wreak their havoc at very different speeds: for example, some viruses spread through your system fairly quickly but don't actually do anything for days or even months.

Viruses are usually passed via disk: you buy or are given a disk that already has the virus on it. Some PC viruses are only passed if you boot (start your computer) from an infected disk-meaning you turn your computer on with the disk already in the floppy drive. Others can infect your system when you copy a file from an infected disk or even when you attempt to erase an infected disk. Viruses can also be passed when you use a modem to download (copy) a file from a computer bulletin board or access data or programs on a network.

Before you get too paranoid about viruses, you should know that most computer users never actually encounter one. In my 11 years of working with personal computers, I have yet to contract a virus (although I have a few clients and colleagues who've been less fortunate). The chances of protracting a virus are particularly small if you only install commercially available, shrink-wrapped programs and rarely exchange disks with anyone else. The more computers your system comes in contact with-via modem or via floppy disks-the greater your risk.

There are several steps you can take to protect your system from viruses:

>Back up your data religiously, and don't discard or overwrite all of your older backups. (As mentioned in Chapter 2, you usually back up data by copying it to floppy disks.) It may take you days or weeks to notice and diagnose a virus, and many of your files may be damaged in the meantime. Backing up your data every day may not help in this case-you may just be backing up damaged files. What you need is an older copy of the data, a copy made before your computer was infected.

>Write-protect floppies whenever possible. Since viruses cannot infect write-protected (locked) disks, you should write-protect any disk that you don't need to copy files to. In particular, always write-protect your original copies of program disks before you insert them into your computer. That way, if you install the program and the copy that resides on your hard disk becomes infected, you can always reinstall from your write-protected floppies.

>Use antivirus programs. Some antivirus programs only detect and eliminate viruses on command. Others are what is known as "terminate-and-stay-resident" programs, or TSRs, meaning that they remain in memory throughout your work session, automatically hunting down viruses on every disk you insert into your computer. If you're very worried about viruses, you may choose to use the terminate-and-stayresident type, in which case, the program will check your hard disk for viruses as soon as you turn on your computer, as well as scoping out any floppies you use over the course of the day. If you don't like the idea of waiting for all this virus-checking, you may prefer a less aggressive approach. In this case, simply use a nonresident virus scanning program whenever you suspect that a disk might be

infected. One reasonable approach is to use such a program to scan any floppies you get from other people (as opposed to from computer stores) before you insert them into your floppy drive. For good measure, you might also scan your hard disk once a month and then create an infection-free backup copy of everything on that disk. You should create these monthly backups in addition to daily or weekly backups of any new or revised data files. That way if you discover that your hard disk has been infected for a week and that you have been backing up infected files, you will have the older, infection-free backup to fall back on.

Note *Some terminate-and-stay-resident antivirus programs conflict with some of your application programs. If you install an antivirus program and then start experiencing problems with your other programs, see if uninstalling the antivirus program solves the problem.*

One final note: Now that you know what viruses are, don't start blaming them for everything that goes wrong with your computer. Most of the problems you encounter on computers will be due to hardware problems, program bugs (mistakes within the program), or your typos and other "user errors." If your computer starts displaying messages about being stoned, or if you keep encountering little happy faces in your word processing documents, by all means, investigate virus protection programs. But consider some of the other possibilities first.

Note *At best, antivirus programs can prevent or repair damage caused by already known viruses. Since new viruses are invented all the time, however, antivirus programs need to be updated to deal with each new menace to your computer's health. In most cases, you can buy updates to your antivirus program for a fraction of the program's original cost.*

Chapter 5
Getting to Know DOS

-The DOS Prompt
-Determining Your DOS Version
-DIR: The Table of Contents Command
-Copying Files
-A Warning about Overwriting Files
-Deleting and Undeleting Files
-Renaming Files
-Formatting Disks
-Directories and Paths
-Loading a Program
-Logging onto Another Disk
-Learning about Your System with CHKDSK
-About Backups
-Other Commands to Explore

No one has ever accused DOS of being user-friendly. In fact, dozens of programs (including Windows) have been written to shield users from the stark and somewhat intimidating DOS environment.

DOS does have its virtues, however. Despite its gruff manner, DOS is a very efficient vehicle for performing disk housekeeping operations. If you like to get things done with as much speed and little fuss as possible, you may actually prefer DOS to more graphical operating systems. You need to memorize a few commands in order to use DOS, but once you do, you can usually tell the computer exactly what you want it to do with a single command. You don't need to spend a lot of time hunting through menus or windows or other paraphernalia on the screen. In short, DOS is a rather minimalist environment, offering plenty of power if little in the way of assistance or pretty pictures.

As with any operating system, you use DOS to do two things:

>*Start up application programs*

>*Manipulate and organize files on disks*

Most of this chapter is devoted to what I call "disk housekeeping" commands-that is, commands for managing files. The more you use your computer, the more files you are likely to accumulate. Learning to handle these files-be they word-processing documents, spreadsheets, or other types of data or program files-is an essential part of using your computer well. At a minimum, you'll need to know how to list the contents of a disk, to copy files from one disk to another (like from your hard disk to a floppy or vice versa), to delete files that you no longer need, and to figure out what happened to that file you created last week. You'll learn to do all this and a bit more in the following pages.

The DOS Prompt

Working in DOS is a matter of typing in commands next to a special symbol known as the DOS prompt. In most cases, when you first see the DOS prompt it looks like C:\>. I say in most cases because you can customize the DOS prompt to use any characters you like. (Some people change the prompt to display their name or the current date, for example.)

Think of the prompt as DOS's way of saying "Okay, now what?" Any time the last line currently on your screen shows a DOS prompt followed by a cursor (a little blinking line), DOS is ready and waiting for your next instruction.In general, your interactions with DOS will work like this:

>*A DOS prompt appears with the cursor blinking after it.*

>*You type one of two things: a DOS command or, if you wish to load an application program, the name of the main program file. (You'll learn more about loading programs later in this chapter.) If you make a mistake while typing, use the Backspace key to back up, erasing as you go, and then type the characters in again. If you want to throw out the entire line, press Esc to display a new DOS prompt.*

>*When you finish typing, you press the Enter key to tell DOS "Okay, I'm done typing. Now go ahead and do it."*

>*DOS does whatever you asked it to do, often displaying a message or two on the screen to let you know what it's doing.*

>*When DOS finishes executing the command, it displays another DOS prompt.*

One thing that people often find confusing at first is that DOS does not clean up after itself: Its responses to previous commands remain on the screen until the screen is full. Once the screen is full, the oldest lines start disappearing off the top of your screen. It might help to imagine the information on your screen as part of an adding machine tape: Even though the results of old calculations may still be visible on the tape, they are no longer relevant to the calculation at hand.

When you use DOS, the only line that's really relevant at the moment is the last one on the screen, the line on which the cursor appears. Everything above that line is simply flotsam and jetsam left over from previous commands.

DOS may be curt, but it's also very forgiving. If you type in a command name that's wrong or invalid, DOS will display the message "Bad command or file name" indicating that it has no idea what you're talking about. (Specifically, the message tells you that DOS knows of no commands or program files with names that match the first word you entered.) Then it immediately displays another DOS prompt, indicating that it is ready to accept another command.

In most cases, the DOS prompt provides you with two critical pieces of information:

>The name of the current disk drive. Remember that on PCs, disk drives are always represented by letters: The first floppy drive is drive A, and the first hard drive is drive C. Whenever you or DOS refers to a particular drive, the drive letter is followed by a colon. Drive C, for example, is represented as C:.

>The name of the current directory-that is, the directory that DOS assumes you want to use.

Returning to the filing analogy introduced in Chapter 3, the DOS prompt tells you which filing cabinet (disk drive) and which part of the file drawer (directory) you are currently in.

When you first turn on your computer, you are looking at the main directory on the disk, which is known as the *root directory*. In the filing analogy, the root directory is the file drawer as a whole. The root directory is always represented by a backslash (the slash that shares a key with the vertical bar character rather than with the question mark). When you see C:\ in either the DOS prompt itself or as part of a DOS command, it means "drive C, the root directory" or "the root directory on drive C."

If your disk is organized properly, the majority of files will be stored in subdirectories. When you get a directory listing for your root directory, you should therefore see only a handful of files and several subdirectories. On floppy disks, the situation is a bit different. Since floppy disks contain relatively few files, the entire collection of files usually resides in the root directory: there is no need to group them into subdirectories.

Note *Technically, there is only one directory on the hard disk-the root directory; all the subgroupings of files are subdirectories. In practice, the terms are used far more loosely. The root directory is always called a directory, but subdirectories are often called directories, as well.*

If you find that the root directory of your hard disk actually contains dozens of files, copy any files that you recognize as data files-such as word processing documents or spreadsheets-to a floppy disk or to one of the subdirectories on your hard disk. Then you can delete the originals from the root. (You'll learn to copy and delete files later in this chapter.) Be careful not to delete files with the names AUTOEXEC.BAT, CONFIG.SYS, or COMMAND.COM. AUTOEXEC.BAT and CONFIG.SYS are files that DOS looks for in the root directory as soon as you finish booting. COMMAND.COM is the main DOS program file.

As mentioned, DOS always assumes that you want to work with the directory that is named in the DOS prompt. When you first turn on your computer, the DOS prompt probably says C:\>. When you start issuing commands that deal with files on a disk, DOS will therefore assume that you are talking about files stored in the root directory on drive C unless you explicitly tell it otherwise. If you tell DOS to copy a file named LETTER.TXT, for example, and you don't specify where the file is located, DOS assumes that the file is in the drive and directory that appear in the DOS prompt. If DOS can't find the file in that location, it simply says "File not found." It doesn't try to track down the file elsewhere. Later in the chapter, you will learn to change drives and directories, a change that will be reflected in the DOS prompt itself.

Before you start learning individual DOS commands, let's take another quick tour of the keyboard, with an eye toward how various keys work in the DOS environment. The figure called "Keys You Can Use in DOS" explains how certain keys work in DOS.

Determining Your DOS Version

The first DOS command you will learn is the VER command, which tells you which version of DOS your computer is currently running. All successful programs are periodically updated and, hopefully, improved by their makers. The versions are identified by numbers. You may, for example, be using DOS 6.0 or DOS 3.3. The number to the left of the decimal point indicates a major revision and the number to the right of the decimal point indicates a minor revision. It's helpful to know which version of DOS (or any other program) you are using because the program's newer features will only be available in the most recent versions. In this chapter, I'll point out any features that were only recently added to the program.

You can use the VER command when you sit down at someone else's computer and want to know which of the new DOS features are available.

Note *DOS doesn't pay any attention to the case of letters. I'll generally be showing commands in uppercase, but you can enter them in any mixture of upper- and lowercase letters that you like.*

DIR: The Table of Contents Command

Often the first task in cleaning up a disk or moving files from one location to another is taking stock of the disk's contents. You can do this with the DIR command, which tells DOS to display a directory (list) of the files on a disk. You may find it helpful to think of this type of directory as a building directory-the roster of companies displayed in the lobby of many office buildings. You can also think of DIR as the table of contents command. I'm going to spend a bit of time on the DIR command, both because you'll use it fairly often and because it can give you a feel for how DOS commands work in general.

Note *You may find it a bit confusing that the word directory has two slightly different meanings in DOS lingo: namely, a group of files on a disk, and a list of files. To help keep things clear, I'll refer to the list displayed when you enter a DIR command as a directory listing.*

By default (meaning, unless you request otherwise), the DIR command produces a list of all the files in the disk and the directory that currently appears in the DOS prompt. At the moment, your DOS prompt probably says C:\>, so if you simply type DIR and press Enter, you will get a list of everything in the root directory on drive C. In most cases, this list will contain a few files and several directories (subdirectories of the root directory). The figure "A Directory Listing for the Root Directory" shows a sample directory listing; yours will undoubtedly look at least a little different.

Note *Most hard disks contain a subdirectory named DOS that contains the files for some of the more specialized, less frequently used DOS commands. Most of the other subdirectories of your root directory will probably contain application programs.*

If your computer is on and you are at a DOS prompt, go ahead and try it yourself: Type DIR and press Enter. The resulting list may be too long to fit on the screen, in which case, some text will scroll off the top. (You'll learn how to fix this problem in a moment.)

The figure "Anatomy of a DOS Directory Listing" shows all the different pieces of the directory listing. The centerpiece is a list of files and directories. The directories are the ones that say <DIR> after the name; the others are files (either program files or data files).

The date and time stamps for each file can prove very useful when you find two different versions of a file on two different disks and you want to determine which is the latest version. The number of bytes free displayed at the bottom of the directory makes the DIR command the perfect way to find out how much room is left on a disk. (This will help you determine whether you have room to copy a particular file onto a floppy disk, for example, or to install a new program on your hard disk.)

Using Command Parameters

Most DOS commands consist of several parts. There's the command name itself, which always comes first. Then there are usually one or more extra parts, known as *parameters,* that tell DOS what you want it to perform the operation on. For example, if you issue the COPY command, you have to tell DOS what you want to copy and to where. In some cases, as with the DIR command, the parameters are all optional. You only use them when you want DIR command to behave a bit differently than it does without such special instructions.

For example, when you enter DIR and press Enter, DOS assumes you want to see all the files in the current drive and directory (that is, the ones displayed in the DOS prompt). But sometimes you'll want to list the files on a different disk or in another directory. You can tell DOS to look in a particular place by including a drive and/or directory parameter in the DIR command. To find out what's on a floppy disk, for example, you could insert the disk in drive A and then enter DIR A: or put it in drive B and enter DIR B:. You'll learn to list the files in another directory near the end of this chapter.

Note *If you tell DOS to look on a floppy drive and the drive happens to be empty, you will see a message like "Not ready reading drive A. Abort, Retry, Fail?" If you simply forgot to put a disk in the drive or to close the drive latch, go ahead and do so and then enter R for Retry. If you don't really want to access the floppy drive and prefer to cancel the command, enter A for Abort.*

Looking for a Particular File

Next, suppose that you don't want to see a list of all the contents of a directory, you just want to know whether a particular file is present. To do this, you add a different parameter to the command: the name of the file you are looking for. To find out whether you have a file named BUDGET.WK1 in the current directory, for example, you would type

```
DIR BUDGET.WK1
```

If DOS finds the specified file, it displays a directory listing that includes only that one file. If it can't find the file, it displays the message "File not found."

You can also look for a particular file on a different disk. To look for BUDGET.WK1 on the disk in drive A, for example, you would enter

```
DIR A:BUDGET.WK1
```

Notice that there is no space between the drive name and the file name. You may find it helpful to think of the drive name as part of the file specification: You are telling DOS to look for a particular file, namely the file on drive A with the name BUDGET.WK1.

Using Wildcards

DOS provides two characters-the asterisk and the question mark-to help you manipulate groups of files at once. These characters are called *wildcard characters* because they stand in for other characters, just as wildcards can stand in for any card you like in a poker game.

The asterisk stands in for a group of characters of any length. The command DIR *.BAT, for example, would produce a list of all the files in the current drive and directory that have any first name, followed by a period and the characters BAT.

Although the asterisk is often used to represent the entire first names of a group of files, it can be used to represent either the end of the first name, the entire file extension, or the end of the file extension. Here are some examples:

```
W*.TXT
```

stands for files with names that start with the letter W and have an extension of .TXT. (This would include the files WANNABE.TXT, WILLIE.TXT, and W.TXT.)

```
A*.*
```

stands for files with names that start with the letter A followed by any other characters in the first name and any file extension.

```
LETTER.*
```

stands for files with a first name of LETTER and any file extension.

The other wildcard character, the question mark, stands in for any single character. Entering MEMO?.DOC would produce a list of all files with a first name of MEMO followed either by nothing or by a single character and an extension of DOC. File names with more than one character after MEMO would be excluded.

You can use wildcard characters in almost any DOS command that has a file name parameter.

File Naming Strategies

You can take advantage of DOS's wildcard characters by assigning similar names to similar types of files. If the names of all your office memos start with MEM, for example, you can display a list of all your memos with the command

```
DIR MEM*.*
```

As mentioned in Chapter 3, many programs automatically assign their own preferred extension to any data files that you create. If you create a spreadsheet in Lotus 1-2-3, for example, the program automatically adds the extension .WK1 or .WK2, depending on which version you are using. Many word processing programs, particularly DOS-based rather than Windows-based ones, let you assign any extension you like. In this case, you might use the extension to indicate the file type-using MEM for memos, INV for invoices, and LTR for letters, for example-and use the eight-character file name to indicate the file's contents. Then, whenever you want to list, copy, move, or delete all your memos, invoices, or letters as a group, you would just refer to those files as *.MEM, *.INV, or *.LTR when you issue the appropriate DOS command.

Switches

Switches are special command parameters that consist of a slash character usually followed by one or two letters. (The slash you want is the one that shares a key with the question mark as opposed to the backslash character, which shares a key with the vertical bar.) Unlike other parameters, switches are always optional and generally speak to issues of style rather than substance. The DIR command, for example, has several switches, all of which let you change the way that the directory listing is displayed on the screen. When you use one of these switches, you're not telling DOS what data, or files, or disks you want to know about (as you do with other parameters); youÕre telling it how you want the information displayed.

The most commonly used switch for the DIR command is /P (the P stands for pause). This switch is useful when a directory listing is too long to fit on the screen. Entering DIR /P directs DOS to pause the display as soon as the screen is full so that you can read the information before it scrolls off the screen. When you issue this command, DOS displays the message "Press any key to continue" at the bottom of the screen. As soon as you press a key, DOS displays the next screenful of information and, if necessary, pauses again.

Note *The same switch can (and usually does) have different meanings as part of different commands. The /P switch means Pause when used in the DIR command, for example, but it means something else in the delete command (which you'll learn about later).*

There are a few other switches that you can use with the DIR command. The /W (for wide) switch, for example, tells DOS to leave out the size, date, and time for each file and just display file names five across on the screen.

If you have DOS version 5 or later, you can control the sort order of the directory listing using the /O (for order) switch. When you use this switch, you follow the /O with another letter indicating the order in which you want the files and directories displayed. Your choices include:

>/ON, which sorts by file name

>/OD, which sorts by date and time, from oldest to most recent

>/OE, which sorts by extension (last name)

>/OS, which sorts by size, with smallest first

>/OG, which sorts so that directories appear first

You can reverse the order of a sort by entering a minus sign between the O and the letter that specifies the sort order. To display a directory in order by date from the latest to the earliest, for example, you would enter DIR /O-D.

Combining Parameters

You've now learned several different variations on the DIR command. You've seen what happens when you enter DIR by itself. You've learned to add parameters to specify a drive to look on and a file to look for. You've also learned to control the directory list with various switches (/P, /W, and /O).

You can mix and match these parts, if you like. For example, you could enter

```
DIR A: /W
```

to produce a wide listing of the files on drive A, or you could enter

```
DIR /P /ON
```

to sort your listing by file name, and had DOS pause each time the screen is filled.

You just need to make sure that the command name itself, in this case DIR, comes first.

Copying Files

One of the most critical tasks you'll perform with DOS is copying files. The basic form for the COPY command is

```
COPY from to
```

The *from* parameter refers to the name and possibly the location of the file you want to copy. (You only need to specify a location if the file is somewhere other than the current directory.) The to parameter refers to the name and/or location of the duplicate you are about to create. You only need to specify the file name if it differs from the name of the source file (that is, if you want to assign a different name to the copy). You only need to specify the location if the duplicate will be placed somewhere other than the current drive and directory. You can omit this entire part of the command if you are copying from a drive or directory other than the current drive/directory and you want the new file to have the same name as the original.

Note *Whenever I describe the format of a command, I'll use lowercase for placeholders-that is, parameter names for which you've supposed to substitute a drive directory, and/or file name.*

The most common purpose of the COPY command is copying from one disk or directory to another. To copy a file from your hard disk to a floppy, for backup purposes, for example, you use the format

```
COPY filename A:
```

or, if the floppy disk is in drive B rather than drive A,

```
COPY filename B:
```

To copy a file named DONDUCK from your hard disk to a floppy in drive A, for example, you would enter

```
COPY DONDUCK A:
```

You will also often copy files in the other direction: from a floppy disk onto your hard drive. This is what you do when you install new software, for example, or when your co-worker gives you a copy of a letter or spreadsheet to work with and you need to copy it from their floppy to your hard disk. The form for copying from a floppy disk to your hard disk is

```
COPY A:filename
```

or, if the floppy is in drive B

```
COPY B:filename
```

Notice that, in this case, you're actually omitting the third part of the command-the to parameter. When you do this, DOS assumes that you want to copy the file to the drive and directory that appear in your DOS prompt and that you want the duplicate file to have the same name as the original.

As with the DIR command, you can use wildcard characters in the file name if you want to copy a group of files that have similar names. To copy all files with an extension of .TXT to drive A, for example, you would enter

```
COPY *.TXT A:
```

To copy everything on drive A to your current drive/directory, you would enter

```
COPY A:*.*
```

To copy all the files in the current drive or directory to drive A, you would enter

```
COPY *.* A:
```

Note *The command COPY *.* A: is the simplest way to make a backup copy of all the files in the current directory of your hard disk. It only works, however, if the contents of that directory are small enough to fit on a single floppy. See the section entitled "About Backups" at the end of this chapter for info on the commands for backing up larger amounts of data.*

You may also occasionally want to copy a file from one floppy disk to another. If you have two floppy drives, you simply include the drive names when specifying the *from* and *to* parameters. To copy a file named KELLY.LTR from a disk in drive A to a disk in drive B, for example, you would enter

```
COPY A:KELLY.LTR B:
```

To copy the same file from B to A, you would enter

```
COPY B:KELLY.LTR A:
```

If you only have one floppy-disk drive, you would use the drive name B: as the destination. DOS will copy the specified file into memory and then ask you to insert a diskette in drive B and press Enter. Go ahead and put the second disk (the one you want to copy to) in the drive, close the latch if necessary, and press Enter. At this point, your computer "thinks" of your floppy drive as drive B. The next time you refer to drive A, it will ask you to put a disk in that drive and press Enter. Assuming the drive already contains a disk, just go ahead and press Enter, so that your computer goes back to thinking of the drive as drive A.

Note *Sometimes you'll want to make a copy of a file in the same directory of the same disk. In this case, you need to assign a different name to the duplicate. You might want to do this just to make an extra copy for safekeeping, or because you plan to modify one of the copies and leave the other unchanged. Your command would look something like this: COPY MYFILE.TXT MYFILE.OLD.*

A Warning about Overwriting Files

When you copy a file from one disk to another, DOS automatically overwrites any existing file with the same name. Suppose, for example, that you issue the command

```
COPY ALFRED.DOC A:
```

to copy a file named ALFRED.DOC from the current drive and directory to a disk in drive A. If drive A already contains a file named ALFRED.DOC, the new ALFRED.DOC (the one created by the COPY command) immediately and irrevocably replaces the old one. DOS does not ask if you really mean to replace the file, and doesn't even tell you that a file was replaced. It simply assumes this is what you want and acts accordingly.

Now, if you are simply copying the latest version of your data onto a floppy disk that you use for backup purposes, this doesn't pose a problem: You do, in fact, want to overwrite the old version of the file with the new one. If you are copying from a floppy onto your hard disk, however, you may wind up accidentally overwriting a critical file. Imagine that you copy a little sample file named REPORT.TXT from a floppy onto your hard disk, for example. If your hard disk happens to have a 3 megabyte customer list with exactly the same name, the little file will overwrite the big one and there will be nothing you can do to recover it. (This is when you better hope you have a backup copy of your data.)

The moral of this story is that whenever you copy from one disk to another, make sure that you will not accidentally overwrite existing files. In particular, if you are copying files from a floppy to your hard disk, use the DIR command to check for existing files of the same name before you issue the COPY command. It's particularly important to do this when you are using wildcard characters to copy a group of files from one disk to another. Otherwise you could accidentally overwrite several files at once.

Deleting and Undeleting Files

Just as you need to clean out your filing cabinets or closets periodically, you also need to clear old and unnecessary files from your disks. It's a good idea to perform such spring cleanings fairly often: The longer you wait, the harder it will be to remember what the files contain, and whether you really need to keep them. (Even if you are very diligent in assigning meaningful file names, when you see a file named JONES.LTR a year from now, you may not remember who Jones is, let alone what you had to say to her or him.)

The format for the DEL command is simply

DEL *filename*

To delete a file named MYFILE.TXT from the current drive and directory, you would enter

```
DEL MYFILE.TXT
```

You can also use wildcards in the file name, to delete a group of files that have common letters in their names. For instance, DEL *.TXT would erase all files with an extension of .TXT. When you include wildcards in the file name parameter, you may also want to add a /P (for Prompt) switch to the command. This tells DOS to prompt you for confirmation on each file. If you enter

```
DEL W*.* /P
```

for example, DOS will display a message like "C:\WP\WALLY.DOC, Delete (Y/N)?" for each file that begins with the letter W. You can then press Y (for Yes) if you want to delete that particular file or N (for No) if you don't.

To delete everything on a floppy disk, you would enter

```
DEL A:*.*
```

or, if the disk were in drive B,

```
DEL B:*.*
```

Whenever you tell DOS to delete everything in a particular directory (in this case, the root directory on drive A or drive B), DOS responds with the message "All files in directory will be deleted! Are you sure (Y/N)?" Press Y to confirm the deletion or N if you've changed your mind.

Note *The ERASE command does exactly the same thing as the DEL command. Use whichever command name you find easier to remember.*

When you erase a file in DOS, the data is not physically removed from the disk. Instead, DOS simply makes a note that the area occupied by that file is now available for reuse. It does this by erasing the first character of the file name in its internal table on files and their whereabouts. (This table is known as the File Allocation Table.) This method of handling deletions makes it possible for you to "undelete" a file, as long (and only as long) as DOS has not reused part or all of the file's disk space for another file.

The command for undeleting a file is UNDELETE, and its format is

```
UNDELETE filename
```

You can include wildcards in the file name parameter if you want to undelete files that have common characters in their file names.

When you issue the UNDELETE command, DOS tells you how many files it has that match your specifications, and how many of those can be recovered. (DOS can only recover files that have not already been overwritten.) Then it displays the name of the first file, with a ? where the first character would normally appear, and asks you to confirm that you want to undelete it. If you are undeleting a file named LETTER.SAM, for example, you will see ? ETTER.SAM, followed by the file's size, date, and time (as in a directory listing) and then "Undelete (Y/N)?" If you type Y (for Yes) to confirm the undeletion, DOS asks you to supply the first letter of the file name. Once you enter a letter, the file is undeleted.

Note *DOS has no way of knowing whether the letter you enter as the first name of the file is correct. It will accept any letter you enter.*

Renaming Files

As you know, you always assign a name to a file when you first create it. If you are creating the file by saving data in an application program, you supply a file name when you issue the save command. If you create a file by copying an existing file, DOS uses either the name of the original file or a name that you specify as part of the COPY command itself.

Occasionally, you'll need to change a file's name later. You might have accidentally entered LETTET instead of LETTER when saving a file, for example. Or you may name a file BUDGET.WK2 only to realize next year that you should have named it BUDGET94.WK2, to distinguish it from the next year's budget.

You can change the name of a file using the RENAME or REN command. (The commands work the same: RENAME may be easier to remember; REN's a little quicker to type.) The form for the command is

```
REN oldname newname
```

To change the name of a file in the current directory from ARDVARK to AARDVARK, for example, you would enter

```
REN ARDVARK AARDVARK
```

If the file is located somewhere other than the current disk or directory, specify the location as part of the *oldname* parameter. The command

```
REN A:MEMO MEMO64
```

would change the name of a file called MEMO, which is located on drive A, to MEMO64. You never include a location as part of the *newname* parameter, because at that point you're only specifying a name for a file. DOS already knows which file you mean and where it's located.

Formatting Disks

As discussed in Chapter 3, when you buy brand-new floppy disks, they are usually unformatted-meaning that they are blank slates without the track or sector lines that allow your computer to "seat" your data in an organized fashion. Before you can use such disks, you need to format them, directing your computer to draw the track and sector lines. If you format a disk that already contains data, that data is erased when you reformat the disk.

To format a disk using DOS, you use the FORMAT command. Assuming that you want to format a disk in drive A, you would issue the FORMAT command by typing

```
FORMAT A:
```

and pressing Enter. If you want to format a disk in drive B, you would type

```
FORMAT B:
```

and press Enter.

Note *Never tell your computer to format drive C (the hard disk) unless you are specifically directed to do so by an expert. Doing so will erase all the data and programs on the disk.*

As soon as you issue the FORMAT command, DOS will display a message like "Insert new diskette for drive A: and press ENTER when ready." (The message varies slightly from one version of DOS to another.) If you have not already placed your floppy disk in drive A, do so now. If the drive has a drive latch, go ahead and close it. Then press Enter to initiate the formatting process.

Note *If you change your mind at this point and decide that you don't really want to format a disk, press Ctrl+Break to interrupt the command.*

The formatting process takes a minute or more. While it's happening, DOS displays messages about what it's doing. Again, these messages vary a bit by DOS version. When the formatting process is complete, you may be given a chance to assign a name to the disk. The message will be something like "Volume label (11 characters, ENTER for none)?" If you want to assign a name to the disk, type it in at this point. You can use up to 11 characters, no spaces. (Most people just note the disk's contents on a sticky label attached to the disk jacket, rather than naming the disk itself.) Finally, you will see the message "Format another (Y/N)?" Press Y for Yes if you do want to format another disk; press N for No if you don't.

Directories and Paths

As you learned in Chapter 3, most hard disks contain several directories, or groups of files. When you first turn on your computer, the current directory is the root directory. If you think of a disk as a file drawer, the root directory is the drawer as a whole, within which there are multiple folders (subdirectories). When it comes to envisioning the many subdirectories on a hard disk, some people find it easier to think in terms of a tree rather than a file drawer. At the base of the tree is the root directory, out of which branch various subdirectories, some of which have branches (subdirectories) of their own.

Many programs, including DOS, represent directory structures in more or less this fashion, displaying diagrams known as directory trees or tree diagrams. (You'll learn to display a DOS directory tree shortly.) In most such directory trees, the root directory is shown either at the top of the diagram or in its upper-left corner, with the branches (subdirectories) extending downward or to the right. (Think of a tree turned on its side.)

As you have seen, whenever you issue a DOS command that involves files (like DIR, COPY, DEL, UNDELETE, and RENAME), DOS assumes that you are referring to files in the current directory. If you want to work with files in a directory other than the current directory, you have two choices:

>*You can specify the location of the other directory in the command.*

>*You can change the current directory.*

Let's start with the first alternative. You've already learned how to specify a disk drive in a DOS command. To get a list of files on the disk in drive A, for example, you enter

```
DIR A:
```

To copy a file from the current directory to drive A, you enter

COPY *filename* A:

To delete a file on drive A, you use

DEL A:*filename*

You use a similar form when you want to tell DOS to manipulate files in another directory rather than (or in addition to) another disk drive. But instead of naming a drive, you state the location of a directory.

Note that I said the location of a directory, rather than the name of the directory. In order to tell DOS to look in a particular directory on the disk, you need to give it a set of instructions on how to get there. This set of instructions is known as a path. Although you can sometimes give directions starting from the current directory, it's usually safest to give instructions starting from the root directory. (DOS always knows how to get back to the root.)

As an example, suppose all your word-processing files are stored in a directory named WP which is a subdirectory of the root directory. The most generic instructions you can give DOS for finding this directory are "Wherever you are now, go back to the root directory and then look for a directory named WP." The way you say this in DOS-speak is \WP.

To tell DOS to look for files in the WP directory, you would enter

```
DIR \WP
```

(This is very much like telling DOS to display a list of files on drive A by entering DIR A:, only instead of specifying a drive, you specify the directory path.) To tell DOS to copy a file named MYLETTER to the WP directory, you would enter

```
COPY MYLETTER \WP
```

To tell DOS to copy all the files in the WP directory to a disk in drive A, you would enter

```
COPY \WP\*.* A:
```

Let's take a slightly more complex example. The path for a directory named LETTERS, which is a subdirectory of the WP subdirectory, which is a subdirectory of the root, would be \WP\LETTERS. This means start at the root directory, then look for the WP directory, and, once you get to the WP directory, look for the LETTERS directory. (It's a little confusing, but you use a backslash both to represent the root directory and to separate the names of subdirectories in case there are more than one in a path. In the latter case, it serves as a kind of punctuation mark, separating one directory from the next.)

When you want to refer to a file that is located in a particular directory, you enter the directory's path, followed by an extra backslash and then the name of the file. If you want to tell DOS to copy a file named SALES.WK1 to drive A from a directory whose path is \LOTUS, for example, you would enter

```
COPY \LOTUS\SALES.WK1 A:
```

If you think of the second part of this command, \LOTUS\SALES.WK1, as a set of directions for finding the source file, you have the following:

>*Start at the root directory.*

>*Within that directory, look for a subdirectory named LOTUS.*

>*Within that subdirectory, look for a file named SALES.WK1.*

Whenever a DOS command has a file name parameter, you can include a directory path as part of that parameter. This is known as specifying a file's *path name.* (The path name for a file named SALES.WK1 located in the LOTUS directory is \LOTUS\SALES.WK1.)

It is all too easy to misplace files on a hard disk-either by neglecting to specify the desired directory when you save a file from inside an application file, or by specifying the wrong one in a COPY command. If you know that you have a file with a particular name, but you can't remember where you put it, you can use the command DIR \filename /S to tell DOS to start looking in the root directory and then search through all its subdirectories as well. (It's the /S switch that tells it to include subdirectories in the search.) In effect, this tells DOS to search through every directory on the current drive. (If you want to search a different drive, just enter the drive name right before the backslash.)

The second means of getting to and manipulating files in other directories is by actually changing the current directory. You do this by means of the CD (change directory) command. You can think of this command as a way of moving to a different part of your file cabinet or, if you prefer to think of the directory structure as a tree, as moving from one branch to another. The format for the CD command is simply

CD *path*

If your hard disk contains a directory named WP, for example, you can change to that directory (open up that hanging folder) by entering

```
CD \WP
```

(The space immediately after CD is actually optional; CD\WP has the same effect as CD \WP.) As soon as you press Enter, the DOS prompt will change to reflect your new location. If you changed to the WP directory, for example, you will see C:\WP> instead of C:\>. Since DOS always assumes that you want to manipulate files in the current directory (that is, the directory named in the DOS prompt), you will get a list of files in the WP directory if you enter DIR at this point. Similarly, if you issue a COPY, DEL, or REN command, DOS will assume that you mean to copy, erase, or rename a file in the WP directory, unless you specify otherwise. (If you want to try this yourself, you need to change to one of the directories on your own hard disk. You may or may not have a directory named WP.)

To change back to the root directory, you enter CD \ or CD\.

Note Just because the DOS prompt changes to display the name of a particular application program does not mean that you've left DOS. If your DOS prompt says C:\LOTUS>, for example, it means that you're on drive C, in the LOTUS directory-that is, the place where the Lotus files are stored. This is not the same thing as being "in" the LOTUS program itself. (You'll learn how to load programs in a moment.) It just means that you're looking at the LOTUS "folder." You might find it helpful to think of this as being at the Lotus program's front door, with the DOS prompt as a kind of signpost indicating that Lotus "lives here."

There are two reasons to change directories. If you want to work with the files in a particular directory for a while, changing directories can save you the trouble of specifying the directory path in every command. Suppose, for example, that you want to get rid of some of your old word processing files and the directory in which they are stored is named \WP60\DATA. Instead of entering commands like this

```
DIR \WP60\DATA\*.* /ON
DEL \WP60\DATA\JONES.*
DEL \WP60\DATA\SMITH.TXT
DEL \WP60\DATA\STOCK.DOC
DEL \WP60\DATA\TODO.LST
```

you could enter commands like this:

```
CD \WP60\DATA
DIR *.*/ON
DEL JONES.*
DEL SMITH.TXT
DEL STOCK.DOC
DEL TODO.LST
```

You may also want to change directories to load a program that is stored in a different directory. You'll learn more about this in the next section.

Displaying a Directory Tree

As mentioned, many programs, including DOS itself, represent directory structures as trees rather than file drawers. You can display a directory tree for your own hard disk using the DOS TREE command. Unless you add a path parameter to the command, the resulting tree diagram will start with the directory that appears in the DOS prompt. In other words, you will see a diagram that includes the current directory and any subdirectories of that directory. If you want to display the directory structure of the entire disk and you're not currently in the root directory, enter

```
TREE \
```

meaning "display a directory tree starting with the root directory." On many hard disks, the directory structure is too complex to fit on a single screen. If you find that the tree scrolls by too fast for you to read, enter

```
TREE | MORE
```

The | MORE part of this command is somewhat complex, but it is known as a *piping*. Its effect is to make the listing pause when the screen is full, similar to the /P switch on a DIR command.

If you want to print a copy of the tree diagram, make sure that your printer is turned on and then enter

```
TREE > PRN
```

You will probably need to eject the last page from the printer manually.

Loading a Program

As mentioned, there are basically two things that you use DOS for: per- forming disk housekeeping operations and loading application programs. Whenever you want to load an application program, you must state your request to DOS. You might find it helpful to think of DOS as a waiter or waitress-that is, as the "person" who "serves" you various application programs. Whenever you want to use a particular program, you place your order with DOS. Alternatively, you can think of DOS as a reference librarian who, in most cases, only lets you check out one reference book at a time. Whenever you want to use a program, you ask the librarian (DOS). Whenever you "turn in" (exit) a book (program), you are back at the librarian.

Note *A few special programs, including Windows, allow you to "check out" multiple application programs at once. Unless you are running one of those special programs, however, you can only use applications one at a time on a PC.*

In most cases, the first step in loading an application program is changing to the directory in which the program is stored. (This is like saying "Go to where the *x* program is stored" or "Move to the *x* program folder.") Once you have moved to the proper directory, you enter the first name of the program's main program file. If the name of the main program file is QPRO.EXE, for example, you type QPRO and press Enter.

Most programs consist of many different disk files. Some of them are program files (that is, sets of instructions). Some are supplementary files. For example, word processing programs that allow you to check for spelling errors need dictionary files that contain the proper spellings of words. Programs that perform fancy printing operations need files called printer drivers that contain the data that your computer needs in order to "talk to" various printers.

Every program has one main program file, the file that you must load into memory to start the program. If you forget the name of a program's main program file, here's a tip: Its extension is either .EXE, .COM, or .BAT. (EXE stands for executable, COM for command, and BAT for batch, as in a whole batch of commands.) So to determine the name of the program file, first change to the directory that contains the program's files. Then try entering DIR *.EXE and see if the list contains any likely candidates. If not, try entering DIR *.COM, and finally DIR *.BAT.

Logging onto Another Disk

You've already learned how to change the current directory-so that a different directory path appears in the DOS prompt. You can also change the current disk drive. To do this, simply enter the name of the drive (don't forget the colon after the drive letter) and press Enter. This is known as logging onto a different drive. To log onto drive A, for example, you would type A: and press Enter.

As with changing directories, changing drives can occasionally save you some typing. If you plan to issue several commands involving files on drive A, for example, logging onto that drive will eliminate the need to type A: before every file name. The other reason for logging onto drive A is to load a program that is stored on a floppy disk. You will often need to do this when installing new programs on your computer. In many cases, the instructions for installing will tell you to first log onto drive A, and then enter INSTALL. This tells DOS to load the program named INSTALL into memory. Since you are already logged onto drive A, DOS automatically looks for the INSTALL program on your floppy disk.

You can always log back onto drive C by typing C: and pressing Enter.

Learning about Your System with CHKDSK

The CHKDSK command (pronounced "check disk") serves two functions:

>It provides you with information about your system, including the amount of disk space and memory you have. This can be extremely useful when you sit down at an unfamiliar computer and want some information about the equipment you are using.

>It looks for and, if instructed, fixes certain types of problems on the disk. It's a good idea to issue the CHKDSK command as a precautionary measure any time you lose power or crash when you're in the middle of running an application program.

When you issue the CHKDSK command, DOS checks your disk for unreadable areas or misplaced data. In general, DOS is very good at filing data in an orderly fashion. If you leave a program in the wrong way-by accidentally turn off the computer without leaving your application program, for example-DOS may not get a chance to store everything where it belongs. In this case, you may end up with lost clusters: chunks of data that are not stored in any particular file on the disk. (Think of papers accidentally misfiled, so that they're in between the file folders. Or, if you're a Star Trek fan, think of a transporter malfunction, scattering some unfortunate soul's molecules across the galaxy.)

If you include the /F switch when you issue the CHKDSK command (so the whole command is CHKDSK /F), DOS groups all such misplaced data into new files. You can then either attempt to edit those files or, more likely, simply erase them from the disk. All files created by CHKDSK are located in the root directory and have names starting with the letters FILE and extensions of .CHK. You can therefore delete all of them at once with the command

```
DEL \*.CHK
```

Varieties of Memory

In PCs, RAM is typically divided into three basic types:

>Conventional memory (sometimes known as normal memory) means the first 640K of RAM in your computer. DOS itself only knows how to run programs and store data within this area of memory.

>Upper memory (also known as high memory) is the area between 640K and 1MB of memory. Upper memory is generally reserved by DOS for managing various input and output devices, and other operating system functions.

>Extended memory is memory beyond the first megabyte. If your PC has four megabytes of RAM, for example, you have 640K of conventional memory, 384K of upper memory (for a total of 1,024K or exactly 1MB) and three megabytes of extended memory. DOS itself cannot directly access extended memory, but other programs including Windows can. However, in some cases those programs require the assistance of a special additional program known as an extended memory manager.

If you have DOS 5 or DOS 6, you can find out how much of each type of memory your computer has by issuing the MEM command. (Just type MEM at the DOS prompt and press Enter.) DOS will respond by listing each of the types of memory, noting how much is installed in your system, how much is currently in use, and how much remains available. (In this listing, the upper memory category is broken down into two parts, identified as upper memory and adapter RAM/ROM.)

There is one last type of memory-called *expanded memory*-that you should know about. In contrast to the other types of memory mentioned, expanded memory is not so much a region of memory as a particular strategy for accessing memory, using an expanded memory manager. The CPUs in the first PCs-namely the 8088 and 80286-were only capable of using up to a single megabyte of memory. Expanded memory was developed as a strategy for breaking this one-megabyte barrier. The expanded memory manager loads programs and data to the extended memory area and then reserves a region within the first megabyte of memory as a special swap region. Whenever you need to access the programs or data in the extended memory area, the memory manager temporarily "swaps" (moves) it into the swap region-the area that DOS knows how to use.

As more and more programs become capable of directly accessing extended memory, expanded memory is becoming obsolete. There are, however, some DOS programs that can only utilize extended memory if it's treated as expanded memory-that is, if it's swapped in and out of a region inside the first megabyte.

About Backups

Creating regular backup copies of your data is one of the most important habits you can develop (right up there with flossing your teeth). Sooner or later, something bad will probably happen to your computer. The hard disk will fail, a power surge will damage some of your data, your office will fall prey to thieves, or your computer will come down with a virus. With luck, such a disaster will only happen to you once in a lifetime's worth of computer use. But given Murphy's Law, it's most likely to strike at the worst possible moment. When it does, having a backup copy of your data can keep an inconvenience from becoming a disaster.

The COPY command is fine for backing up small amounts of data. If you need to back up more data than will fit on a single floppy disk, however, you will need a different tool for the task. If you are using DOS version 5.0 or earlier, you use the BACKUP command. If you are using DOS 6.0 or later, you use a command named MSBACKUP. Rather than simply copying data to a disk, both BACKUP and MSBACKUP store the data in a special backup format. If something happens to the original copies of your data and you need to use your backup copy, you can't simply copy the contents of the backup disk back onto your hard disk. In the case of BACKUP, you need to use a RESTORE command. In the case of MSBACKUP, you need to select the Restore option on the main MSBACKUP screen.

MSBACKUP is significantly more sophisticated than the BACKUP command provided with earlier versions of DOS. It not only creates a backup copy of your data, it compresses it in the process-meaning that it shrinks it down considerably in size by removing blank spaces. (Don't worry, the spaces are put back in when you restore the files.) This both speeds up the backup process and allows you to fit more data on your floppy disks. MSBACKUP also allows you to perform what are known as *incremental backups*-backing up only those files that have changed since the last time you performed a full backup. If you perform a full backup of 20 files one day and then change two of them the next day, for example, the incremental backup will copy only the two that have changed.

The BACKUP, RESTORE, and MSBACKUP commands are beyond the scope of this book. (MSBACKUP, in particular, is easy to use, but it leads to a menu-driven program in itself, that takes a little time to set up and get used to.) Refer to your DOS manual or a book on DOS for information on how to use them.

Other Commands to Explore

Although the material in this chapter will give you a good start with DOS, sooner or later you will probably want to buy a book on the subject. (For a thorough, easy-to-follow introduction, try *PC Learning Labs Teaches DOS 6.0*, published by Ziff-Davis Press.) In particular, you may want to explore the following commands:

>XCOPY copies files in a way that is often more efficient than the COPY command and supports several additonal switches (including one that lets you copy files that changed on or after a specified date).

>MOVE lets you both move files and rename directories.

>MD (make directory) lets you create new directories on a disk.

>RD (remove directory) lets you eliminate empty directories from a disk.

>DELDIR (delete directory) lets you erase the contents of a directory and then remove the directory from the disk.

Chapter 6

Using Windows

As you learned in Chapter 1, Windows is an operating system that works hand-in-hand with DOS, providing a more graphically oriented alternative to the DOS command line. This chapter introduces the Windows environment including how you start and exit Windows, how you load application programs within Windows, and how you use Windows to perform disk housekeeping operations like copying and deleting files. It also explores many of the features common to all Windows application programs as well as Windows itself-things like windows, dialog boxes, and menu bars. Since installing Windows is no task for a beginner, it is beyond the scope of this book. I will therefore assume that Windows has already been properly installed on your computer. (Many computers are now sold with Windows already installed.)

Note *In order to run Windows, you need a fairly powerful PC, one with both a reasonably fast CPU (an 80286 is the minimum, an 80386 or 80486 is better) and a substantial amount of memory (preferably at least 4 megabytes). You also need a monitor and video adapter card capable of displaying graphics. (A video card is an expansion board that lets your computer "talk to" your monitor.)*

What's So Great about Windows?

For the uninitiated, Windows is an operating environment created by Microsoft. It provides users of DOS-based personal computers with the kind of "graphical user interface" (GUI) long cherished by Macintosh users. In other words, the Windows environment is organized primarily around pictures rather than around text. While issuing orders in DOS is a matter of typing in commands, in Windows programs it's a matter of clicking pictures on the screen or making selections from menus (on-screen lists of options). The advantage of this way of interacting with the computer is that it's less intimidating to new users and there is less memorization required. If you forget how to perform a particular task, you can often refresh your memory simply by poking around on the screen, seeing which menu options and icons are available.

Another advantage of using Windows is that most programs designed to run within Windows look and behave similarly. Most Windows application programs are populated with entities such as icons (pictures that represent data files, programs, or groups of programs), dialog boxes (frames that display information and/or ask you questions) and, as you might guess, windows (rectangular frames in which programs and data are displayed). Almost all Windows programs also feature similar menu systems and at least a few of the same commands. To leave any Windows application, for example, you choose the Exit option from a menu named File. Once you have mastered one Windows program, it's fairly easy to learn the next because Windows programs have so much in common.

Windows also allows you to run multiple programs at once. If you're just getting started with computers, this feature may seem of limited value: Why would you want to run two programs at once when you're still feeling overwhelmed by the first one? Once you get a little more comfortable with your system, however, you may find this feature invaluable. Imagine getting a phone call about your latest sales figures when you're in the middle of typing a letter in your word processing program, for example. If you are using Windows, you can easily open your spreadsheet program and find the necessary information without leaving your word processing program. When you're done, a single keystroke will take you back to your word processing document, and you can pick up exactly where you left off. Windows also includes a feature known as the Clipboard that lets you easily copy or move data from one program to another. This means that you can copy those sales figures directly from your spreadsheet into your word processing document without having to use any special importing or exporting commands, as you would in DOS.

Windows not only allows you to open two (or more) programs at once, it also allows you to carry out work in both programs simultaneously. If you need to perform a time-consuming task in one program, you can simply start the process and then switch to a different program. The task you started in the first program will continue unattended while you continue working in the second one. People refer to this ability to work on two things at once as *multitasking*. It's the computer equivalent of patting your head while you rub your stomach.

Starting Windows

If your computer has been set up to run Windows automatically, you will probably see a screen with the words "Program Manager" at the top shortly after you turn on your computer. (It may take a few moments for the program to load, so be patient). If you know your computer has been set up to run Windows, but you see a DOS prompt (something like C:\> followed by a blinking cursor) rather than the Program Manager screen, you'll need to start Windows yourself. To do this, type WIN and press the Enter key.

If DOS responds with an error message, you need to switch to the subdirectory of your hard disk that contains the Windows files first, by typing CD \WINDOWS and pressing Enter. With luck, the last line on your screen will now read C:\WINDOWS>. Next, go ahead and type WIN and press Enter. You will briefly see a Windows sign-on screen. Then the screen will clear and you will probably see the Windows Program Manager. Once you see the Program Manager screen, you know that Windows is up and running.

Note *If none of these strategies work, you'll need to check with the person who set up the computer or with some other accessible computer expert to make sure that Windows has been installed on your computer and, if so, where.*

What Are Windows?

The Windows program gets its name from its most ubiquitous feature: the rectangular frames, called *windows*, that you encounter at every turn. Just about everything that happens in the Windows environment takes place inside a window.

When you first start Windows, you usually land in a window known as the Program Manager. The reason I say that you "usually" land in the Program Manager is that the Program Manager is actually one of two possible *shells* (master programs) available in Windows. The other is the File Manager, which you'll learn about later in this chapter. You can designate the File Manager as your shell program; if you do, you will see the File Manager window when you first start Windows and you will launch application programs from there.

The Program Manager is the Windows control center: the place from which you launch (start) other programs. Not only is it the first screen you see when you start Windows (after the Windows sign-on screen), it's also the last window you see before you exit. And itÕs the place from which you install new programs or remove old ones.

There are two types of windows in the Windows environment. An *application window* is a window that houses a program. A *document window* is a window that you open inside an application window. It contains data specific to that program. If you load the word processing program Word for Windows, for example, a Word for Windows application window appears on the screen. Then, every time you open or create a document, it is displayed in its own document window within this main window. Document windows always reside inside of application windows and cannot exist without them. If you close an application window, any document windows it contains are closed as well.

Note *In case you're wondering where the two types of windows get their names, in Windows-speak, programs are usually called applications, and document means any set of data, regardless of whether it contains text, pictures, numbers, or anything else. The term "document" is therefore roughly equivalent to "file," except that a document has not necessarily been named and saved to disk. If you are using a spreadsheet program, each spreadsheet is a document. If you are using a drawing program, each drawing is a document.*

Both Windows itself and most Windows applications let you arrange your electronic desktop in any way you like. You can work with one window at a time or several. You can move the windows around, shrink them, expand them, arrange them side by side, pile them on top of each other, and so on. In short, you can keep your electronic desktop as spartan or as cluttered as your other desk. (As with a normal desk, however, the more chaotic your Windows desktop, the harder it becomes to find what you need when you need it.)

The Program Manager

As mentioned, the first window you are likely to encounter when you load Windows is the Program Manager window. This window remains open as long as you stay in Windows, even if it is completely obscured by other application windows. As soon as you close the application windows, the Program Manager window will emerge from the background.

It's difficult to say what, exactly, your Program Manager window will look like because this depends on which programs have been installed and how you (or whoever set up Windows for you) chose to arrange elements on the screen. But it will probably contain a handful of items known as *program groups*.

You can think of program groups as little baskets used to organize program icons. As you'll learn later in this chapter, you use these icons to launch application programs. It's entirely up to you how many program groups you have and which icons they contain. Program groups can either appear as windows with one or more icons inside or as icons themselves.

At a minimum, you probably have program groups entitled Main, Accessories, and Games. (Again, I say probably because this, too, can be customized. If some grinch in your office set up Windows and didn't want you to waste company time having fun, she or he may have deleted the Games program group, for example.)

Each icon inside a program group represents an individual application. For instance, the Main program group contains icons for the Control Panel and File Manager, among others, which are little application programs that are built into Windows. The Games program group generally contains icons for the games Solitaire, Reversi, and Minesweeper.

All windows contain several common elements, which are shown in the figure "Anatomy of a Window."

Having Your Way with Windows

All the windows you'll encounter in Windows are very malleable: You can expand, shrink, push, pull, and rearrange them to your heart's content. When you first open a window, it may occupy less than the entire screen or, in the case of a document window, less than its entire application window. If you want more room to work in, you can expand the window as far as possible by either clicking its Maximize button or opening its Control menu and selecting Maximize. (You can open the Control menu either by clicking the Control menu box or by holding Alt while you press the spacebar in an application window or holding Alt while you press the hyphen in a document window.) Bear in mind that even when you maximize an application window, other application windows, including the Program Manager, remain open on the Desktop; they are just not visible at the moment. (You will learn to switch from one application window to another later in this chapter.)

Note *When you maximize a document window, you get two sets of Control menu boxes and sizing buttons, which can be a little confusing. The Control menu box and sizing buttons at opposite ends of the title bar affect the application window. The ones on the next row control the document window.*

Once you have maximized a window, the Maximize button itself is replaced by a Restore button, which contains one arrowhead pointing up and another pointing down. Clicking on this button restores the window to its previous size-that is, the size it was just before you maximized it.

You can also maximize and restore windows by double-clicking their title bars. If the window is not maximized, double-clicking the title bar expands it to full screen; if it is already maximized, double-clicking the title bar restores it to its previous size.

To minimize a window (temporarily reduce it to an icon), you click its Minimize button. When you minimize a window, it disappears from the screen and is replaced by an icon, usually at the bottom of the screen. Minimizing a window is rather like placing it in a corner of your desk, somewhere in your "to do" pile. You're not putting it away completely; you're just removing it from the center of your attention, with the aim of returning to it later. (In more technical terms, when you minimize a window, you shrink it on the screen but leave it in memory.) You can restore a minimized window in an instant just by double-clicking its icon.

If you prefer using your keyboard, you can also maximize, restore, and minimize a window by opening the window's Control menu and selecting the appropriate option.

Note *When you're first getting used to Windows, it's sometimes hard to distinguish between the types of icons. Any icon that's "loose" in the Program Manager window, rather than enclosed in a program group window, represents an entire program group rather than an individual program. In addition, program group icons always look like little windows with icons inside. (Some people say they look a bit like waffles.) The icons for individual application programs are less standardized.*

Moving, Resizing, and Closing Windows

You can move a window (assuming it is not maximized) by dragging its title bar with the mouse. If you prefer using your keyboard, open the Control menu, select Move, and then press the appropriate arrow keys. Press Enter when you reach the desired spot.

To resize a window, simply drag its borders with your mouse. Start by slowly moving the mouse pointer across the border until you see the pointer change to a double-headed arrow. Then press the left mouse button, drag the border until the window is the desired size, and release the mouse button. If you want to change both the height and width of a window, drag one of the window's corners. To resize a window using the keyboard, open the Control menu and select Size. Then use your arrow keys to move the window's borders, and press Enter when you're done.

There are four ways to close a window:

>*You can double-click the Control menu box in the window's upper-left corner.*

>You can make sure the window is active and press Ctrl+F4. (This only works in document windows.)

>You can open the window's Control menu and select Close.

>You can open the File menu and select Exit. (This only works for application windows and closes the application itself.)

Bear in mind that closing a window means removing its contents from memory. In the case of document windows, this just means removing data from memory. In the case of application windows, it means leaving the program.

Using Windows's Menu Systems

As you have seen, there are many operations that you can perform in Win- dows just by clicking, double-clicking, or dragging with your mouse. But others require you to work with menus, telling the computer what you want it to do by selecting options from on-screen lists.

There are three main types of menus in Windows and in most Windows applications: menu bars, pull-down menus, and cascading menus. Some applications also feature what are called *pop-up menus*, which pop up from the bottom of the screen and only remain on screen as long as you hold down the mouse button. And a few feature *object menus*, which appear whenever you point to an object on the screen and press the right mouse button.

In many cases, selecting an option from one menu evokes another menu with a more specific set of options. For example, in most Windows programs, selecting File displays a pull-down menu of options related to managing files on your disk, including options for creating new files and opening existing ones. Many of the options on the File menu lead to cascading menus that contain more detailed sets of options. In some cases, you may need to select options from three or four menus before anything actually happens. When you have to select options from more than one menu to issue a command, the command's full name consists of all the various options. The command for leaving an application, for example, is File, Exit-meaning first select File and then select Exit. (You may also sometimes see the options separated by vertical bar characters, as in File|Exit.)

To select an option from a menu bar, you can either:

>Click the option with your mouse. For example, to select File and thereby open the File menu, you simply click on the word "File."

>Hold down the Alt key and press the underlined letter in the option name. To select File, for example, you would press Alt+F. You can also press F10 to activate the menu bar and then press either the Right Arrow or Left Arrow key until the desired option is highlighted and press Enter, or type the underlined letter in the option name.

Once a pull-down or cascading menu is open, you can select options in one of three ways:

>You can click on the option with your mouse.

>You can use the Up Arrow or Down Arrow key to highlight the option and then press Enter.

>You can type the underlined letter in the option name.

To close a menu, simply click anywhere outside the menu or press Esc.

Loading an Application Program

Windows's main purpose is to serve as a platform for applications software, a launch pad for more goal-specific database, spreadsheet, and/or word processing programs. The first step in loading an application program is finding and, if necessary, opening its program group. (Most programs belong to program groups that bear either the program's or manufacturer's name.) If you don't see the program group you are looking for, select the Window option on the menu bar at the top of the Program Manager window either by clicking the option or pressing Alt+W. A complete list of program groups will appear at the bottom of the menu and you can select the one you want.

If you see the program's program group but it is minimized (appears as an icon), open the group either by double-clicking the icon or clicking the icon once and then selecting Restore from the resulting Control menu. Once the desired program group is open, you can start the program itself by double-clicking on its icon or by selecting the icon and pressing Enter.

A program does not need to be designed for Windows in order to run inside the Windows environment. You can run regular DOS programs under Windows as well. (The main reason for doing this is to run more than one program at once, and switch back and forth between them.)

To launch a DOS application from inside of Windows, open a DOS window by double-clicking the MS-DOS icon (usually located in the Main program group) or by selecting the MS-DOS icon and pressing Enter. Then, when you see the DOS prompt, follow the usual procedure for starting up the program. It's also possible to create an icon for a DOS application so that you can launch it directly from the Program Manager, but it's a little early to learn the specifics here. (Once you get comfortable with Windows, refer to your Windows documentation for instructions on creating a new program item.)

When you first open a DOS window, it occupies the full screen. If you prefer to display it inside a window on the Desktop, press Alt+Enter. (You can press Alt+Enter again if you want to switch back to full-screen mode.)

Switching Applications

As mentioned, one of the benefits of using Windows is that you can run several applications at once. In fact, whenever you are working with an application program in Windows, you are always running at least two programs: the application itself and the Program Manager.

No matter how many applications you open at once, however, only one of them is active at one time. That program is sometimes called the *foreground application*, because it always sits on top of the stack of application windows-"in front" of all the others. (Depending on whether the window is maximized, you may or may not be able to see part of the other application windows underneath.) The foreground application is also the one that receives most of your CPU's attention. Other open applications are said to be running *in the background*, meaning that they continue to plod along at any tasks you have assigned to them, but they do so slowly, since they are receiving a relatively small part of your CPU's "brain power."

There are several methods of switching from one program to another while inside Windows:

>If any part of the other program's application window is visible, click in that window.

>If the desired application has been minimized and its icon is visible, double-click on the icon.

>Hold Alt and press Tab to switch from one application to the next. As you cycle through the programs, Windows displays only the border of each application window, including the title bar. When you reach the application you want, release both keys. The Alt+Esc key

combination works in almost the same way, but automatically switches you to a full-sized window rather than displaying just the window title.

>Hold Ctrl and press Esc to open the application window's Control menu. Then select Switch To to display the Task List, which shows all the programs that are currently running. Choose the desired program from the list and then select the Switch To button, or else double-click on the appropriate program name.

Note *Bear in mind that if you open too many programs at one time, your system may slow to a crawl. The more programs you are running, the more time your computer needs to spend swapping things into and out of memory, and the less attention each program receives from your CPU.*

Using the Clipboard

Windows has a feature, known as the Clipboard, that allows you to copy or to "cut and paste" (move) data from one place to another. Think of the Clipboard as a temporary holding pen, a place that you place data that you want to transport from one spot to another.

You can use the Clipboard to:

>Move or copy data to another location within the same data file. You might copy a paragraph from the third page to the fifth page of a report, for example, or move a set of numbers from one part of your spreadsheet to another.

>Move or copy data to a different data file within the same application, copying a sentence or two from one letter to another, for instance.

>Move or copy data to a data file in another application. You might copy part of a spreadsheet into a report you are creating in your word processing program, or copy a picture from a graphics program to a newsletter you are creating in a desktop publishing program.

Bear in mind that you can only copy or move the data into a file or area that suits it-that is, in which that type of information makes sense. You can't just drop a photograph into the middle of a word processing document, for example, although if your word processing program supports graphics, you can create a special graphic box and place the picture inside of that.

Moving or copying data via the Clipboard involves four steps:

>First you must select the data that you want to move or copy. (More on this in a moment.)

>Then you either copy or "cut" (move) that data to the Clipboard, using the Edit, Copy or Edit, Cut command. (To do this, open the Edit menu by either clicking the word Edit on the menu bar or by holding Alt and pressing E. Then select either Cut or Copy.) If you have trouble figuring out whether to cut or copy, think about what you want to happen to your original copy of the data. If you want the original to remain where it is, use Copy. If you want the original to be erased, use Cut. In either case, the data will be copied to the Clipboard for pasting somewhere else.

>Next, you move to the desired location-be it in the same file, a different file, or a file in another application.

>Finally, you paste the data from the Clipboard using the Edit, Paste command.

Note *When you paste data from the Clipboard, you do not erase it from the Clipboard. You* *simply copy it to a designated spot. Anything you place in the Clipboard remains there until you either place something else in the Clipboard or leave Windows. This means that you can easily paste the same set of data into several different locations. (If you have a very long company name, for example, you might type it once, copy it to the Clipboard, and then paste it in from the Clipboard every time you want to use it again.)*

Selecting Data for the Clipboard

As mentioned, the first step in copying or cutting data to the Clipboard is selecting the data that you want to manipulate. In most Windows programs, you can select data by dragging across it with your mouse. You can also often select data by clicking at one end of the set of data you want to select and then holding Shift while you click on the other end. To select discrete items on the screen, you select one by clicking, and then hold Ctrl while you click others. If you prefer to select data using your keyboard, try any of the following:

>Shift+End selects from the insertion point to the end of a line

>Shift+one of the arrow keys selects from the current position of the insertion point to wherever you move the insertion point. If you hold Shift while you press the Down Arrow key three times, for example, you'll select from your original spot to a spot three lines down.

Keyboard Shortcuts

Most Windows programs let you issue commands using key combinations (often known as keyboard shortcuts or menu command shortcuts). To determine the shortcuts for the cut, copy, and paste commands in a particular program, open the Edit menu (by clicking on Edit or pressing Alt+E) and then see which shortcut is listed to the right of the option name. The ^ character is used to represent Ctrl, so ^C, for example, means hold Ctrl and press C.) The shortcut for Edit, Copy is usually either Ctrl+C (meaning hold Ctrl while you press C) or Ctrl+Ins (hold Ctrl while you press Ins). The shortcut for Edit, Cut is either Ctrl+V or Shift+Del. The shortcut for Edit, Paste is either Ctrl+V or Shift+Ins.

If you are running a DOS application under Windows, you can copy data from that program to the Windows Clipboard, but the procedure's a little different. (You cannot cut data from a DOS application to the Clipboard.) If you are running the application "full screen" rather than inside a window on the Desktop, press Alt+Enter to switch to a window. Then open the window's Control menu, select Edit, and then choose Mark. Next, press the arrow keys while you hold Shift to highlight the data that you wish to copy to the Clipboard. Finally, open the Control menu again, and choose Edit and then Copy to copy the data to the Clipboard. Then you follow the usual procedure for switching to another application and pasting in the data.

Managing Files with the File Manager

Chapter 5 explained how to use DOS to perform disk housekeeping operations like copying and deleting files. In Windows, you can perform these same operations using a program named File Manager.

You start the File Manager just as you would any application program. First you need to find the appropriate program icon. (In most cases, the File Manager icon resides in the Main program group.) Once you find it, either double-click it, or select it and press Enter.

The File Manager window always contains at least one document window, which I'll refer to as a *directory window*, that displays information about the contents of a particular disk. At the top of the directory window, just below the title bar, are a set of buttons representing various disk drives, with a box around the drive that is currently selected. (Recall that A and B represent floppy drives, C and higher are hard drives or other mass storage devices like CD-ROM drives.) To view the contents of a different drive, just click the appropriate drive letter.

Unlike the other windows you encounter in Windows programs, directory windows are divided into two sections, known as *panes*. The left pane, called the *tree pane*, contains a graphic representation of the directory structure of the currently selected disk. (As mentioned in Chapter 5, such representations are commonly known as *directory trees*.) At the top of the directory tree is a folder representing the root directory (the main directory on the disk). Below and to the right of that folder are folders representing the subdirectories of the root directory. By default, the directory tree includes only the root directory and its subdirectories; it does not include another level of subdirectories. To display the subdirectories of a particular subdirectory (if any), either double-click its folder icon, or select it by clicking or moving to it using the arrow keys. Then press the plus sign (+). To display all the levels of subdirectories below a particular subdirectory, select the subdirectory and press the asterisk (*). To display all the levels of subdirectories on the entire disk, hold down the Ctrl key and press the asterisk. To hide all levels below a particular subdirectory, either double-click its folder icon or select the subdirectory and press the minus sign (-).

The right pane of the directory window is known as the *directory pane* or *file pane*. It displays a list of the files and subdirectories in the currently selected directory. To select a different directory, click the directory's name in the tree pane. The contents of the directory pane will immediately change to reflect the contents of that directory. Whenever you select a directory, its name is highlighted and its icon changes to an open file folder. (The folders for all the other directories are closed.) The current directory's name also appears in the directory window's title bar.

You can use the scroll bars to scroll through either the tree pane or the directory pane, and you can move down and up a windowful at a time by pressing the PgDn and PgUp keys. The End key takes you to the bottom of the list and Home takes you back to the top.

The first step in manipulating (copying, moving, renaming, or deleting) a file is selecting its name in the list of files within the directory pane. There are a few techniques you can use to select files.

>To select one file, just highlight its name by clicking or using the arrow keys.

>To select multiple files, highlight the first one's name and then hold down Ctrl while you click on others.

>If the files you want to select happen to be contiguous, highlight the first (highest) one you want to select and then hold down Shift while you highlight the last (lowest). (You can also highlight the last file and hold down the Shift key while you click the first file.)

*>If the group of files that you want to manipulate can be specified using DOS wildcard characters, issue the File, Select Files command (select File from the menu bar and then choose Select Files). Windows will display a little dialog box in which you can enter your file name specification. (If you want to select all files with an extension of .TXT, for example, enter *.TXT. To select files with a name that starts with Z, enter Z*.*.) Next, click on Select to select the specified files. (You will see lines above and below the names of all the files that match your specifications.) Finally, click on Close to close the dialog box and highlight the files.*

To deselect a group of files, click the mouse anywhere else, without holding down Ctrl or Shift.

Once you have selected one or more files, you can use one of two strategies to copy or move them: You can drag one of the selected files icons to the icon representing the destination, or you can issue a copy or move command, using either the menus or menu command shortcuts. Let's start with the dragging techniques.

To copy file(s) from your hard disk to a floppy, simply drag any one of the selected file names over to the button representing that drive and then release the mouse button. To move a file or files to a floppy disk, press the Shift key and hold it down while you drag.

To copy or move files from a floppy to your hard drive:

>In the directory tree at the left side of the File Manager window, select the directory on the hard drive into which you want to copy or move the files.

>Click the icon for the floppy drive that contains the file(s) you want to manipulate.

>To copy, just drag the icon for any one of the selected file(s) from the directory pane to the icon for drive C. To move, hold down the Shift key while you drag.

To copy or move files from one directory on your hard disk to another:

>In the directory tree at the left side of the File Manager window, select the directory on the hard drive into which you want to copy or move the files.

>Click the icon for the drive that contains the file(s) you want to manipulate.

>To move, just drag the icon for any one of the selected file(s) from the directory pane to the folder in the tree pane that represents the desired directory. To copy, hold down the Ctrl key while you drag.

In each case, when asked whether you're sure you want to copy or move the selected files or directories, select Yes.

As mentioned, you can also copy or move files using menu commands or menu shortcuts. To copy files, issue the File, Copy command (select File from the menu bar and then choose Copy) or press F8. To move files, issue the File, Move command, or press F7. Regardless of which command you choose or how you initiate it, Windows will display a dialog box in which you can specify the destination. If you want to copy or move files to a different directory, be sure to include the path when you enter the file name. (Paths were explained near the end of Chapter 5.) For example, to copy or move a file to LOTUS, which is a subdirectory of the root directory, you would enter \LOTUS.

To delete one or more files, start by selecting the files and then either press Del or issue the File, Delete command. Windows will display a dialog box entitled Delete that names the file(s) to be deleted. Click OK to proceed, or click Cancel if you've had a change of heart. Next, another dialog box asks you to confirm the deletion of the first file you selected. Click Yes if you want to delete that particular file and then be asked about the others. Click Yes to All to delete all the selected files. Click No if you've decided not to delete that particular file. Click Cancel to cancel the entire delete operation.

If you want to undelete a file that you've deleted by accident, select the file and then issue the File, Undelete command.

To rename one or more files, select their file names and then issue the File, Rename command. When Windows displays a Rename dialog box, enter the new name you want to use in the box labeled "To" and click on OK.

To format a floppy disk, place it in the drive and, if necessary, close the drive latch. Windows will display a Format Disk dialog box. If you want to format a disk in drive B rather than drive A, click the little arrow at the right side of the Drive A box. This will drop down a list of floppy drives. Click on Drive B. Leave all the other options as is and select OK to initiate the formatting process.

Getting Help

Windows features a very extensive and easy to use Help system-a series of informational screens and a set of tools for navigating them and finding the information you need. When you are in an application program, the Help system provides information on that particular program. When you are at the Program Manager, the Help system provides information on Windows itself.

There are two ways into the Windows Help system:

>You can open the Help menu on the menu bar by clicking it or pressing Alt, H and then select the desired option.

>You can press the function key F1, which is usually located in the upper-left corner of the keyboard, to the right of the Esc key.

The options on the Help menu depend on which application you are in. The first option is always Contents, which leads to a kind of table of contents for that program's help screens. The last option is always named either About or About *x* (with *x* being the name of the program), and usually displays the program's serial number and the name of the person to whom it is registered. In the case of the Program Manager, this option also provides information on the available memory and system resources (meaning both memory and processor power). The other options on the Help menu vary from one program to the next.

The F1 key usually leads to what is known as *context-sensitive help,* meaning a screen with information about whatever operation you are performing or object you are manipulating at the moment. If you open a menu, highlight an option, and then press F1, for example, you will see a screen with information about that menu option.

The figure "The Program Manager Help Screen" shows the window displayed when you select Contents from the Help menu option in the Program Manager. In all help screens, words or pictures that are underlined are called keywords. (On color monitors, keywords appear in a contrasting color, such as green.)

There are two types of keywords:

>Topics, which have solid underlines, are links to additional help screens on the specified subject.

>Terms, which have dotted underlines, invoke dialog boxes with definitions of the underlined word(s).

Whenever the mouse pointer moves across either type of keyword, it changes from an arrow to a hand. To select a keyword, you either click it, or press Tab until it is highlighted and then press Enter. (You can move backwards through the keywords on a screen by holding down Shift and pressing Tab.)

All the screens in the Help system contain both a menu bar and, in most cases, a bar of mouse buttons. To use one of the buttons, click it or type the underlined letter in the option name. The Program Manager's Help screen includes these buttons:

>Contents always takes you back to the Contents screen.

>Search leads to a Search dialog box that you can use to locate information on particular topics. (Sometime the Help menu itself contains a Search option that opens the same dialog box.) To use this dialog box, you can select the list of topics by clicking it or tabbing to it, and then move down to the topic you are interested in, using either the arrow keys or the scroll bar. Alternatively, you can type in a word or words and, as you type, the highlight will move to the item that is closest alphabetically to the word or words you typed. Once you have an interesting item highlighted, click the Show Topics button to display a more specific list of topics, highlight one of those, and then click the Go To button.

>Back takes you back to the last help screen you viewed.

>History displays a list of help screens you have viewed. To return to one of those screens, either double-click that option in the list or tab to it and press Enter.

>Glossary (only available in the Program Manager) displays a list of terms for which help screens exist. These are all defined terms, so you can just click them or tab to them and

press Enter to display a definition.

Help is really its own little program. It runs in its own application window, and can be minimized like other application windows. When you are learning a new application, you may want to keep the Help system running but minimized so that you can easily access it when you need it. (Remember that you can restore a minimized application by double-clicking its icon.)

When you are done using the Help window, close it just as you would any other application window. (It's usually easiest to double-click the Control menu box or use the File, Exit command.)

For more information on using the Help system, open the Program Manager's Help menu and select How to Use Help. The Help menu also includes an extremely useful option entitled Windows Tutorial. Select this option for an interactive tour of Windows in which you'll get a chance to practice your mouse skills and manipulate windows.

Leaving Windows

You should always leave Windows and return to the DOS prompt before you turn off your computer. To leave Windows, simply return to the Program Manager and then close the Program Manager window, using any of the usual window-closing techniques. (Double-click the Control menu box, issue the File, Exit command, or click the Control menu box once and choose Close.) You will see a dialog box informing you that this will end your Windows session. Press Enter or click OK to proceed. If you have any open application windows with open and unsaved documents, Windows will ask if you want to save your data. Once you have either saved or discarded any unsaved documents, you will be returned to the DOS prompt.

Note *If you are running any DOS (non-Windows) applications within Windows, you will need to close them before you can leave Windows. Otherwise, you will see a dialog box informing you that an application is still active.*