Microsoft® Operating System/2

Beginning User's Guide

Microsoft Corporation

Document Number 510830001-200-O01-1087

Contents

Welcome

Welcome to Microsoft® Operating System/2 (MS® OS/2). MS OS/2 is a powerful operating system that lets you take full advantage of the Intel 80286 processor in your personal computer.

If you are new at working with a personal computer, you will want to learn a few basics before you go on to learn about the advanced features of MS OS/2. The *Microsoft Operating System/2 Beginning User's Guide* was written to help you understand the fundamentals of using an operating system such as MS OS/2 with your personal computer.

In this short book, you will learn what an operating system does. You will also learn how information is stored on a computer and how MS OS/2 works with your personal computer.

Also see these manuals

After you have read this manual or if you are already an advanced user of personal computers, see the *Microsoft Operating System/2 User's Reference* to learn in detail about MS OS/2 features and commands. If you want to learn how to install MS OS/2 or how to configure your system, see the *Microsoft Operating System/2 Setup Guide.*

In addition, three manuals discuss topics of interest to programmers:

- The *Microsoft Operating System/2 Programmer's Guide* introduces programmers to MS OS/2.
- The *Microsoft Operating System/2 Programmer's Reference* describes all MS OS/2 system calls.
- The *Microsoft Operating System/2 Device Drivers Guide* describes how to use device drivers with MS OS/2 and explains device-driver commands.

Notational Conventions

About this manual

Throughout this manual, the following conventions are used to distinguish elements of text:

Text	Use
bold	Distinguishes commands, options, switches, and literal portions of syntax that must appear exactly as shown.
italic	Distinguishes variables and placeholders that represent the kind of text to be typed in by the user.
`monospace`	Distinguishes sample command lines, program code and examples, and sample sessions.
CAPITALS	Distinguish filenames, directory names, programming languages, and acronyms.
SMALL CAPITALS	Distinguish keys and key combinations.

A plus sign (+) used between two keynames indicates that those keys must be pressed at the same time. For example, "Press CONTROL+S" means that you should press and hold down the CONTROL key while you press and release the S key. Then release the CONTROL key.

How to Use This Guide

The following is a quick overview of the topics covered in this manual:

Turn to	To find out about
Chapter 1	Terms you'll need to know; special keys you use with MS OS/2
Chapter 2	How MS OS/2 works with your personal computer
Chapter 3	How MS OS/2 organizes floppy and hard disks; how to format a disk
Chapter 4	How to name files; how to copy, rename, or delete files; how to use the **print** and **type** commands; how to use wildcard characters
Chapter 5	How to create and use directories
Chapter 6	How to keep your data files safe

1 Learning About MS OS/2

Often when you are introduced to a new or different subject, you must learn a new set of words to understand that subject. The Microsoft Operating System/2 is no exception.

In this chapter, you will learn the following:

- Some terms you will need to know so that you can understand and use MS OS/2
- How MS OS/2 uses some of the keys on your computer's keyboard

Terms You Should Know

Introducing MS OS/2 terms

The terms described in the following sections will help you understand and use MS OS/2.

Program

A program, often called an application program, application, or software, is a series of instructions written in a computer language. These instructions are stored in files and tell your computer to perform a task. For example, a program might tell your computer to sort a list of names alphabetically.

File

A file is a collection of related information, like the contents of a file folder in a desk drawer. File folders, for instance, might contain business letters, office memos, or monthly sales data. Files on your disks could also contain letters, memos, and data.

Filename

Just as each folder in a file cabinet has a label, each file on a disk has a name. This name has two parts: a filename and an extension. A filename can be from one to eight characters and can be typed in uppercase or lowercase letters. MS OS/2 automatically converts filenames to uppercase letters.

The filename extension includes a period followed by one, two, or three characters. Although filename extensions are optional, you should use them, since they are useful for describing the contents of a file. For instance, if you want to be able to quickly identify your report files, you can add the filename extension .RPT to each one. Here's an example of a filename with this extension:

```
progress.rpt
```

When you look at the files on your MS OS/2 program disk, you will see many that have the extension .EXE or .COM. The extension .EXE means "executable," and .COM means "command." These extensions tell MS OS/2 that the file contains a program that can be run. Some

files will have other kinds of extensions, such as .DOC and .TXT, and these may contain text. Another common program-file extension is .BAS for BASIC programs.

Directory

A directory is part of a structure for organizing your files into convenient groups. You can think of a directory as being similar to a drawer in a file cabinet. A directory can contain both files and subdirectories.

Memory

Memory is the place in your computer where information is actively used. There are two types of memory in your computer. One is called read-only memory (ROM). This is what your computer reads automatically when you turn the power on. There are just enough instructions in ROM for your computer to know that it must look further for more instructions. The other, larger part of your computer's memory is called random-access memory (RAM). This is the area to which data is written as a program has need for it. When the program no longer needs that data, or when RAM is full and needs to move in new data, it writes the old data back to disk storage.

Disk

Your computer typically includes drives for one or more types of disks. MS OS/2 supports three types of magnetic-disk storage:

- 5 1/4-inch floppy disks
- 3 1/2-inch disks
- Hard disks

Because the 5 1/4-inch disks are flexible, they are often known as floppy disks. These disks are enclosed in protective covers. The front of the cover is smooth, while the back has visible seams. A high-density floppy disk stores up to 1.2 million characters.

Like 5 1/4-inch floppy disks, 3 1/2-inch disks are portable magnetic disks. Data on 3 1/2-inch disks are more compressed, so depending on the style, a single 3 1/2-inch disk can store as much or more data as a high-density floppy disk. These smaller disks, sometimes called

microfloppies, have rigid plastic covers with metal shields that guard the disk from dirt and fingerprints. When you place the disk into the disk drive, the computer automatically moves this shield aside to read the disk.

A hard disk, sometimes called a fixed disk, is often built into the computer (as opposed to the removable 5 1/4- and 3 1/2-inch disks). Hard disks store much more information than removable disks. Computers take less time to find information stored on a hard disk than on a removable disk.

Volume Label

When you use a new disk, you'll probably put a label on the outside of it to help you identify its contents. You can also give each of your disks an internal name, called a volume label.

You can look at the volume label on a disk by displaying its directory listing. Some programs may check the volume label to see if you are using the correct disk, so be sure to label your disks.

Disk drive

To use the programs or data files that are on a floppy disk, you must first insert the disk into a floppy-disk drive. Floppy-disk drives are commonly referred to as drive A and drive B. The first hard-disk drive, normally installed inside your computer, is usually referred to as drive C. Check your computer manual to see which drive is A and which is B (or C).

Drive name

A complete drive name consists of a drive letter and a colon. When using a command, you may need to type a drive name before your filename to tell MS OS/2 where to find the disk that contains your file. For example, suppose you have a file named FINANCES.DOC on the disk in drive C. To tell MS OS/2 where to find this file, you would type the drive name before the filename:

```
c:finances.doc
```

If you don't specify a drive name when you type a filename, MS OS/2 automatically searches for the file on

the disk in the default drive (the drive you are using). MS OS/2 displays a prompt that contains the default-drive letter (the prompt lets you know that MS OS/2 is ready to receive a command). For example, when your prompt is A>, MS OS/2 searches only the disk in drive A for files and programs—unless you tell it to search in another drive.

You can change the default drive by typing the letter of the desired drive followed by a colon. For example, if you will be working primarily with files on drive C, it would be easier to change the default drive to C, so that you wouldn't have to type **c:** with every command and filename.

Device

Whenever you use your computer, you supply information—input—and expect a result—output. Your computer uses pieces of hardware called devices to receive input and send output.

For example, when you type a command, your computer receives input from your keyboard and disk drive, and usually sends output to your screen. It can also receive input from a mouse, or send output to a printer. Some devices, such as disk drives, both receive input and send output.

Devices are connected to ports on your computer. There are two kinds of ports—parallel and serial (asynchronous). The computer uses the name of the port to identify a device. For example, if a lineprinter were connected to the first parallel port, LPT1, the computer would use "LPT1" as the device name.

When you add a new device, such as a mouse, to your computer, you must tell MS OS/2 that the device is there by setting up (configuring) your computer for that device. For more information on setting up your computer for devices, see the information supplied with your device, or the *Microsoft Operating System/2 Setup Guide.*

Command

Just as you will run application programs to create and update files containing your data, you will also need to run some special programs, called MS OS/2 commands, that let you manipulate entire files.

When you type MS OS/2 commands, you are asking the computer to perform tasks. For example, when you use the **diskcopy** command to make a copy of your MS OS/2 program disk, you are running a program named Diskcopy that copies the files on the MS OS/2 disk.

Other MS OS/2 commands perform tasks such as the following:

- Copy and display files
- Format and label disks
- Run your own programs, as well as programs supplied with MS OS/2, such as Edlin
- Set printer and screen options

For a detailed description of each MS OS/2 command, see the *Microsoft Operating System/2 User's Reference.*

Multitasking

When you give your computer a command, it performs a task. MS OS/2 allows your computer to perform multiple tasks — that is, MS OS/2 lets you run more than one program at the same time. This is called multitasking.

For example, you might want to run a word-processing program at the same time you are printing a file. With multitasking, you could perform both operations at once on one computer.

Keys You Use with MS OS/2

In addition to the keys you'd find on a typewriter, your computer keyboard has some keys that have special meanings to MS OS/2.

Keys that look alike

First, note that there are two important differences between a typewriter keyboard and a computer keyboard:

[illus. 1 key & l key on left]

A computer understands the difference between a one (1) and a lowercase L (l). Be sure you don't type a lowercase L when you mean a one.

[illus. O key & and 0 key]

Capital O and zero (0) may look alike, but they have different meanings to a computer. Many computers display a zero with a diagonal line through it (Ø). Make sure you type the correct letter or number when you give commands to MS OS/2.

The ENTER Key

ENTER key
[illus. of ENTER key]

Press the ENTER key after you type commands. When you press the ENTER key, MS OS/2 performs the command.

Special Keys

DIRECTION keys [illus. DIRECTION keys]

DIRECTION keys move the cursor right, left, up, and down. They do not affect the characters that are displayed. Some programs ignore these keys or do not use them. These keys are also called the RIGHT, LEFT, UP, and DOWN keys.

SPACEBAR
[illus. SPACEBAR]

The SPACEBAR moves the cursor to the right, deleting any characters already on the line. To move the cursor along a line without deleting any characters, use the DIRECTION keys.

BACKSPACE key
[illus. BACKSPACE key]

The BACKSPACE key deletes characters as it moves the cursor to the left. Use the BACKSPACE key to correct typing mistakes on the current line.

CONTROL-Key Combinations

CONTROL key
[illus. ctrl key]

The CONTROL key lets you give complex commands to your computer by pressing only two or three keys. You must hold down the CONTROL key while you press another key. That is, you use the CONTROL key as you would the SHIFT key.

CONTROL+S
[illus. ctrl + s keys]

When you press the CONTROL key and the S key at the same time, you can stop the scrolling of the screen display. To resume scrolling, press CONTROL+Q.

CONTROL+C
[illus. ctrl + c keys]

When you press the CONTROL key and the C key at the same time, you can stop a command before it finishes executing.

CONTROL+ALT+DELETE
[illus. Control+ Alt+Del keys]

If you want to restart MS OS/2, press the CONTROL, ALT, and DELETE keys at the same time. This is also called rebooting MS OS/2.

Function Keys

[illus. Function keys]

Your computer has ten function keys, labeled F1 through F10. They are grouped together (often at the top or left-side of the keyboard). These are programmable keys that allow you to do specific functions with a single keystroke. MS OS/2 uses these keys to perform editing functions.

For more information about MS OS/2 editing and function keys, see the *Microsoft Operating System/2 User's Reference*. Applications that you run may use these keys for different purposes.

Summary

The main points of this chapter included the following:

- Data and programs are kept in files. Each file is given a unique identifier, called a filename.
- Files are stored on disks. Disks have unique identifiers called volume labels.
- MS OS/2 commands are special programs that help you work with your files and disks.
- MS OS/2 gives special meanings to several keys on your keyboard, including the following:

Key	Purpose
ENTER	Enters commands after you type them
CONTROL	Combined with other keys for special purposes
DIRECTION keys	Move the cursor up, down, left or right
Function keys	Used with MS OS/2 as editing keys

In this chapter, you've seen what the keys on your keyboard can do, and learned some of the MS OS/2 vocabulary. In Chapter 2, you'll learn how an operating system works with the personal computer to run application programs.

2 Learning About Operating Systems

Operating systems such as Microsoft Operating System/2 have two main functions. One is to act like a translator between you and your computer. The other is to set standards. These standards provide a framework that allows a variety of application programs to run on your computer.

In this chapter, you will learn the following:

- How an operating system translates between you and the computer
- How the operating system enables programs to run on the computer

How Operating Systems Work

There are three basic components to a computer system:

- **Hardware** This includes equipment such as a disk drive, keyboard, monitor, mouse, printer, plotter, and modem.
- **Application software** This includes, for example, word processing applications, spreadsheets, database programs, and computer games.
- **Operating-system software** This group of programs sets the standards for what software will run on the computer and acts as a translator between you and your computer.

An Operating System Sets Standards

MS OS/2 set standards

The operating system acts as an intermediary between application software and hardware configurations. A configuration is the way a personal computer is set up. One personal-computer configuration may include a high density floppy-disk drive, a hard-disk drive, a standard United States keyboard, and a screen with black-and-white display. Another configuration may include four disk drives, a French keyboard, a color display, a light pen, a mouse, a plotter, a printer, and an image scanner.

The operating system sets standards that allow different hardware configurations to work with a variety of application programs. For example, when you buy an application, the software comes on floppy disks. These disks are formatted according to standards established by the operating system. If the software you buy is compatible with MS OS/2, this means that the floppy disk containing that software is formatted by MS OS/2 standards. Any personal computer that uses MS OS/2 can then read that software from the floppy disk by using a compatible floppy-disk drive. Because the disk conforms to the standard MS OS/2 format, the computer knows where to begin reading on the disk. The computer also knows how the information is organized on the disk.

An Operating System is a Translator

MS OS/2 translates your commands

The operating system also acts as a translator between you and your computer. An operating system such as MS OS/2 is the common link between the language your computer understands and the language you speak. The operating system lets you communicate with your computer, your disk drives, and your printer, letting you use these resources to your advantage.

Computers understand only two things—basically, yes and no. Either a switch is on or it's off. This logic is the basis of binary mathematics. In binary, the language of computers, there are only two digits—0 and 1.

Learning a language of only two words would be simple, but using it to communicate complex ideas would be tedious. Fortunately, MS OS/2 allows you to communicate in more natural language than binary; it lets you use words like "copy," "print," and "label" to communicate ideas. These words are used to identify MS OS/2 commands. When you type an MS OS/2 command, MS OS/2 translates it into a machine language understood by your computer. Your computer is then able to perform the function related to that command.

For example, if you wanted to print a file called SEASON.NEW, you would type the following command:

```
print season.new
```

MS OS/2 translates this command into machine language for your computer. When your computer receives the message, it prints the file on the printer connected to your computer.

Programs Work with the Operating System

Just like MS OS/2 commands, application programs get "permission" from MS OS/2 to run on the computer. For example, an application program must ask MS OS/2 to retrieve a data file from the disk. If the application program creates new data, the program must ask MS OS/2 to write that data on disk and to give the data file a name.

As an example, suppose you have a program called Roadrunner that keeps track of the calories you burn when you follow a particular exercise routine. Most programs have a short, easy-to-remember command that starts them. Let's assume that the command to start Roadrunner is **runner**.

The following steps illustrate how this program might interact with MS OS/2 when you run it:

1. Type in the command to start the program:

   ```
   runner
   ```

2. MS OS/2 interprets the command for the computer, letting the computer know that it needs to find the Roadrunner program on the disk.
3. When the computer locates the program, it copies the program into a work area called memory.
4. If the program needs any files, it asks MS OS/2 to copy them from disk into memory so that the program can use them. Since Roadrunner keeps track of your progress, it may ask MS OS/2 for a history file of the calories you've burned each day.
5. When the computer has all the data and the program in memory, it performs whatever the program tells it to do.

   The Roadrunner program is an interactive program, meaning that it waits for you to enter information or to ask to see information. When you type commands for Roadrunner, MS OS/2 translates what comes through the keyboard for the program.
6. The last command that you need to type is one that will end the interactive program. Each application program dictates what command is used to end it.
7. When the program is completed, the computer copies any new or revised data files back to disk storage. If you entered any new statistics through the Roadrunner program, Roadrunner asks MS OS/2 to update the appropriate file and copy it back to the disk.

Summary

The main points of this chapter included the following:

- An operating system sets standards that let application programs run on your computer.
- The operating system translates between you and your computer.

- When programs run on your computer, MS OS/2 uses a special work area called memory.
- When programs and data files are not in use, they are stored on disk, or on some other form of storage used by the computer.

In the next chapter, you will learn more about how MS OS/2 works with floppy and hard disks. You will also learn how to format a disk by using the **format** command.

3 Learning About Disks

In Chapter 2, you learned that disks are used to store the data files you work with using Microsoft Operating System/2. There are at least two kinds of disks your computer uses — floppy disks and hard disks. Floppy disks are small and portable. Hard disks are often built into your personal computer and can store more data than a floppy disk.

In this chapter, you will learn the following:

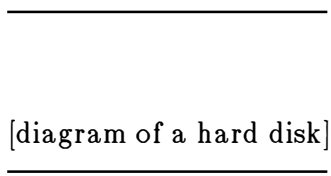
- How floppy and hard disks are organized
- Some tips for using floppy disks
- How to format and label disks

How Disks are Organized

Sides, tracks, and sectors

Disks are divided into data storage compartments by sides, tracks, and sectors. Single-sided floppy disks store data on only one side. Double-sided floppy disks store data on both sides. The flexible magnetic disk is protected by a plastic cover. There are 40 concentric rings, called tracks, around each side of a double-density floppy disk. Tracks on a floppy disk are similar to the grooves on a record. A high-density floppy disk has 80 tracks per side.

Hard disks consist of two or more platters stacked on top of one another. Thus, hard disks have four or more sides. Hard disks also have more tracks per side than floppy disks. (On hard disks, tracks are called cylinders.) A 20-megabyte hard disk, for example, has 615 cylinders per side.



Sectors further compartmentalize the disk. Imagine several lines that look like spokes on a bicycle tire, radiating from the center of the disk to its outer edge. On each track, the area between these imaginary lines is called a sector. Depending on the floppy disk, MS OS/2 allows 8, 9, or 15 sectors per track. A high-density disk, for example, contains 15 sectors per track. Ten- and 20-megabyte hard disks generally have 17 sectors per cylinder.

Each sector is a storage compartment for data from one file. If a file is larger than 512 bytes, the file's contents will be written to several sectors. Sectors are not necessarily filled to capacity.

Using Floppy Disks

When you use a floppy disk, you should label the outside of the disk to identify what files it contains. Place the label on the front of the floppy disk's protective cover, at the top, so that the label doesn't touch the magnetic surface of the disk. You should use a soft, felt-tip pen when writing on labels—a pencil or ballpoint pen can damage the disk if you press too hard.

You should store floppy disks in a safe place, away from dust, moisture, magnetic objects, and extreme temperatures.

[do's and don'ts of floppy disks]

Write-protected disks

Some floppy disks are write-protected, letting you examine the information on them but not change it. Floppy disks can be write-protected in one of two ways. Some have a small piece of tape, called a write-protect tab, covering a notch on the right side of the disk. You can copy information onto a disk protected this way by first removing the write-protect tab; however, you should consider why the disk was protected before you change its contents. It is always a good idea to replace the write-protect tab after you have copied or changed a write-protected disk.

If a disk does not have a write-protect notch, it is permanently write-protected. Many programs, including MS OS/2, come on write-protected floppy disks to protect their files from being destroyed accidentally.

How MS OS/2 Formats a Disk

When you purchase new disks, they are blank and unformatted. You must format them before MS OS/2 can use them. Formatting structures a disk with tracks and sectors so that MS OS/2 can find and store information on it. Formatting also checks the disk for defective spots.

In addition to creating sectors on the disk, MS OS/2 creates three special areas on the disk. First, MS OS/2 reserves an area for a boot record. If MS OS/2 is copied onto the disk, this area is used to store the program that starts MS OS/2. Next, MS OS/2 creates a file-allocation table. This area becomes a map for the disk. When MS OS/2 formats a disk, it also writes address marks on the disk. Then, when a file is written to the disk, its address is logged in the file-allocation table. The third item MS OS/2 creates on the disk is a directory. This area tracks the name, size, and creation date for each file on the disk. The rest of the disk is available for data storage.

[Formatted disk, incl: boot, FAT, dir]

Using the Format Command

Formatting your disks

You can format a disk by using the **format** command.

Warning The **format** command destroys any information that is on a disk. You should view the directory listing of a disk that contains files *before* you format it; then you can make sure that you aren't destroying important files. You can display a disk's directory listing by using the **dir** command. See Chapter 5, "Learning About Directories," for more information about the **dir** command.

To format a blank disk in drive A, you would use the following command line:

```
format a:
```

Remember that if you want to format a high-density floppy disk, you must place it in a high-density drive.

Formatting a system disk

You can also format a blank disk so that some important MS OS/2 system files are copied to it during formatting. These files are necessary if you want to use the disk to start MS OS/2. To format a blank disk in drive A and

include these special MS OS/2 files, you would type the following command line:

```
format a: /s
```

If you have a disk and don't know whether you can use it to start MS OS/2, put the disk into drive A and press CONTROL+ALT+DELETE to reboot MS OS/2. If the disk does not contain the system files, MS OS/2 displays an error message.

Important Remember to copy the system files with the **format** command when you make a copy of your MS OS/2 program disk.

Formatting a disk

As a detailed example of how to use the **format** command, suppose you need to create a new data disk to hold some tax records, but you don't want to copy the special MS OS/2 files when formatting the disk. To format and label a blank disk (in drive A) without including the special MS OS/2 files, follow these steps:

1. At the MS OS/2 prompt, type the following command line and press the ENTER key:

   ```
   format a:
   ```

2. When you are prompted to do so, insert a blank disk in drive A.
3. Press the ENTER key to start the formatting process.

   When formatting is complete, MS OS/2 displays the following prompt:

   ```
   Type in an 11-character volume label,
   or press Enter.
   ```

4. Type the computer-readable volume label that you want to use to identify this disk (for example, **TAX DISK**), and press the ENTER key. MS OS/2 then displays a message like this one:

   ```
   1213952 bytes total disk space
   1213952 bytes available on disk

   Format another?  (Y/N)
   ```

5 Type **n** (for No) to end the formatting process.

Now your disk is formatted and ready to use. Be sure to label it on the outside cover, and remember to include the volume label that you used in step 4. The label will remind you that you have formatted the disk, and will help you identify the disk's contents.

Adding Labels to Formatted Disks

Using the Label and Vol commands

If you want to add a computer-readable volume label on a floppy disk that is already formatted, use the **label** command. For example, if you type the following command line, MS OS/2 reads the label on the disk in drive A:

```
label a:
```

MS OS/2 displays the current label, if any, and prompts you to type a new label:

```
volume in drive A has no label
volume label (11 characters, ENTER for none)?
```

To read but not edit the volume label of a disk, you can use the **vol** command. For example, if you want to look at the volume label of a disk in drive C, type the following command line:

```
vol c:
```

MS OS/2 displays a message like this in response:

```
Volume in drive C: is 20-MEG DISK
```

For more information about MS OS/2 commands, see the *Microsoft Operating System/2 User's Reference.*

Summary

The main points of this chapter included the following:

- Personal computers primarily use two kinds of magnetic disk storage: floppy disks and hard disks.
- You should take certain precautions when using floppy disks:
  - Use only soft-tipped writing instruments, like a felt-tip pen, to write on external labels of floppy disks.
  - Keep floppy disks away from dust, magnetic objects, moisture, and extreme temperatures.
- You use the **format** command to format disks.

In the next chapter, you'll learn about some MS OS/2 commands that help you to manage your files.

4 Learning About Files

Microsoft Operating System/2 keeps data and programs in files on disks. With MS OS/2 commands, you can give your data files names that help you remember what each file contains. You can also use a number of MS OS/2 commands to help manage your data files.

In this chapter, you will learn the following:

- How to name your files
- How to use wildcard characters to search for files
- How to copy, rename, and delete files
- How to use the **type** and **print** commands to look at a file's contents

Naming Your Files

In Chapter 1, you learned that a file is a collection of related information given a unique filename. The filename that you give a file should help you to identify the file's contents.

For example, a file that contains a script might be named EPISODE.ONE. Other examples of filenames might include the following:

BUDGET.88
TAKEOVER.BID
FINANCES.RPT
SCHEDULE

Important You can type filenames in uppercase or lowercase letters; MS OS/2 automatically converts lowercase letters into uppercase letters.

Valid filename characters

Many of your filenames will contain only letters and numbers. But you may also use any of the following symbols in your filenames and extensions:

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

Warning Some applications may not let you use all these symbols. If in doubt, use only letters and numbers.

Note also that many applications assign their own extensions to your filenames unless you specify otherwise. For example, if you create files using Microsoft Word, your files will be given the extension .DOC unless you specify another extension.

Filenames that MS OS/2 Reserves

Before you begin creating your own data files, you should be aware of the filenames that MS OS/2 reserves for its own use. These names are used for specific devices that your computer uses. The following names are reserved:

COM1	LPT1
COM2	LPT2
CLOCK$	LPT3
CON	PRN
SCREEN$	NUL
KBD$	POINTER$

These filenames are not reserved as extensions, so you can use any of the three-character names as extensions, for example, .PRN.

Using Wildcards in Filenames

Wildcards are special characters that take the place of other characters in a filename. They are useful in MS OS/2 command lines because they give you flexibility when you are specifying directories and files. The two wildcard characters you can use with MS OS/2 are the asterisk (*) and the question mark (?).

The ? wildcard

A question mark (?) in a filename or filename extension means that any character can occupy that position. The following command, for example, lists all filenames on the default drive that begin with SCHD, have any characters in the next two positions, end with PM, and have a .AUG extension:

```
dir schd??pm.aug
```

Here are some examples of files that might be listed by the preceding command:

```
SCHD08PM.AUG
SCHD09PM.AUG
SCHD10PM.AUG
```

The * wildcard

An asterisk (*) used in a filename or filename extension means that any character can occupy that position or any of the remaining positions in the filename or extension. For example, the following command renames all files with a .TXT extension on the default drive as the same filenames with the new extension .LST:

```
ren *.txt *.lst
```

If you include the * wildcard as the first character of a filename in a command line, MS OS/2 will not recognize any other characters in the filename. For example, if you type **dir a:*1***, all the files in the working directory on drive A will be listed, not just those that contain the number "1."

To MS OS/2, the wildcard name *.* means all files in a directory, so be careful when you use this abbreviation in your commands. For example, if you type **del *.***, MS OS/2 asks if you want to erase *all* the files in the directory. If you type **y** (for Yes), MS OS/2 permanently deletes the files in the directory.

Duplicating a File

Using the Copy command

You can use the **copy** command to make a duplicate of one or more files, either on the same disk or from one disk to another. For example, suppose that you want to copy a file named RATINGS.FEB on drive C and that you want to call this new copy RATINGS.TV.

Note The filenames used in the following examples are for illustrative purposes only—to use these commands, you would substitute the name of an actual file.

To copy the RATINGS.FEB file and call the new copy RATINGS.TV, you would follow these steps:

1. At the MS OS/2 prompt, type the following:

```
copy ratings.feb ratings.tv
```

2. Press the ENTER key.

You cannot give the new copy of a file the same name as the original. You can, however, copy a file from one disk to another, or from one directory to another, and keep the same filename. For example, to copy a file from drive C to the disk in drive A, you would type a command line similar to the following:

```
copy c:newshow.hit a:newshow.hit
```

Note In the example above, if C is the default drive (that is, if the prompt is C> or [C:\]), you don't need to type **c:** before the first filename. Also, by default, the copy will have the name of the original file if you do not specify a new name. For example, the following command lines all copy the file NEWSHOW.HIT to a disk in drive A:

```
copy c:newshow.hit a:newshow.hit
copy newshow.hit a:newshow.hit
copy newshow.hit a:
```

Deleting a File

Using the Del command

Just as you may want to make copies of files, you may also want to remove old or unnecessary files. When you want to erase a file from a disk, you can use the MS OS/2 **del** command. But remember that the **del** command *permanently* erases the file.

For example, to delete a file name PILOT.SHOW from the disk in drive A, you would type the following command line at the MS OS/2 prompt:

```
del a:pilot.sho
```

If this file were on the disk in the default drive, you would follow these steps to delete it:

1. At the MS OS/2 prompt, type the following command line:

   ```
   del pilot.sho
   ```

2. Press the ENTER key.

Here are some guidelines for using the **del** command:

- Use wildcard characters sparingly with the **del** command to avoid unintentional deletions.
- The **del** command does not work if you type **delete**. You can, however, use the **erase** command instead of the **del** command.

Renaming a File

Using the Rename command

Occasionally, you may want to change the name of a file. For example, suppose you have a file named SPYSHOW.NEW on a disk. When you add other monthly reports to your disk, you may want to change the name to something more specific. To change the name to SPYSHOW.9PM, for instance, you would type the following command line:

```
rename spyshow.new spyshow.9pm
```

You can rename files only on the same disk, so you cannot type **rename a:spyshow.new c:spyshow.new**. Also note that you cannot rename a file to a filename that already exists. Thus, if the files DETECTIV.PLT and POLICE.PLT both exist on drive C, MS OS/2 will not allow you to rename DETECTIV.PLT to POLICE.PLT. This ensures that you don't lose files.

As an example of how to use the **rename** command, suppose there is a file on drive C called ADVERTIS.DOC that you want to rename to SPONSORS.DOC. To do so, you would simply follow these steps:

1. At the MS OS/2 prompt, type the following:

   ```
   rename c:advertis.doc sponsors.doc
   ```

2. Press the ENTER key.

Shortcut Some MS OS/2 commands can be abbreviated so that you don't have to type them in their entirety to use them. The **rename** command can be abbreviated to **ren**.

Printing a File

Using the Print command

If you have a printer attached to your computer, you can print files with the MS OS/2 **print** command. For example, suppose that you have a file named REMINDER.LST that you want to print. You would use the following command line to do so:

```
print reminder.lst
```

As another example, suppose that you have a file that contains a list of advertisers and their phone numbers and that you want to print this file and keep it near your phone. The file is named ADVERTIS.MNT and is on the disk in drive A. Drive C is the default drive (C> or [C:\] is the prompt). To print the ADVERTIS.MNT file, you would follow these steps:

1. At the MS OS/2 prompt, type the following:

```
print a:advertis.mnt
```

2. Press the ENTER key.

   MS OS/2 prompts you for the name of the printing device connected to your computer (this name is usually the name of the communications port that the printer cable is connected to, for example, LPT1).

3. Type the name, or press the ENTER key to print on the default printer.

In the event that you have not copied the MS OS/2 commands to a hard disk but are using a floppy-disk copy instead, MS OS/2 may prompt you to insert the MS OS/2 disk as needed.

Displaying a File's Contents

Using the Type command

If you want MS OS/2 to display a file that contains text (often called a text file) on the screen, use the **type** command. For example, if you have created a file named PHONE.LST on the disk in drive C, and you want to display the file on the screen, you would type the following command line:

```
type c:phone.lst
```

As another example, suppose you want to check the production costs for several projects, so you decide to look at a file named PRODUCTN.CST on the disk in the default drive. To display this file, you would follow these steps:

1 At the MS OS/2 prompt, type the following:

```
type productn.cst
```

2 Press the ENTER key. MS OS/2 then displays the PRODUCTN.CST file on the screen.

If the file is too long to fit on one screen, you can press CONTROL+S to prevent it from scrolling off the screen. Press CONTROL+Q when you want the file to resume scrolling.

MS OS/2 displays only text files on the screen. If you try to display a program file (a file with an extension of .COM or .EXE), you will see only symbols on the screen.

If you have an application program that creates files, you may need to run the application to view them.

Summary

The main points of this chapter included the following:

- Files must be given unique filenames and/or filename extensions. Filenames are a maximum of eight characters in length. Filename extensions consist of a period followed by one to three characters.
- You should give your files names that describe their contents. Examples of descriptive filenames are BUDGET.87 and PHONE.LST.
- The wildcard characters * and ? can be used with some MS OS/2 commands in place of one or more characters in a filename.
- You can use the following MS OS/2 commands with files:

Command	Action
copy	Makes a duplicate copy of a file.
del	Deletes a file.
rename	Renames a file.
print	Sends a file to the printer.
type	Displays a file's contents on the screen.

In the next chapter, you'll learn how to use directories to organize your files.

5 Learning About Directories

If you're like most people, you will soon have more files on your hard disk than you can easily keep track of. Microsoft Operating System/2 lets you use directories to organize your files into convenient groups.

In this chapter, you will learn the following:

- How to list a directory's contents
- How to change to a new working directory
- How to create new directories
- How to plan the directories you will need
- How to remove directories that you no longer need

Keeping Track of Your Files

Using directories

MS OS/2 stores the names of files in directories. When you format a disk, MS OS/2 automatically creates an empty directory, called a root directory, on the disk. Then, as you create files on that disk, each filename is added to the disk's directory.

The directories, along with the disk's File Allocation Table, enable MS OS/2 to recognize and organize the files on your disk.

If you want to know what files are in a directory, you can use the **dir** command. This command tells MS OS/2 to list all the filenames in a specific directory. For example, to display the contents of the root directory on a disk in drive A, you would use this command line:

```
dir a:
```

If you use the **dir** command without a drive letter, MS OS/2 lists the contents of the directory on the disk in the default drive.

Using the Dir command

For example, suppose you want to see how many files are in the root directory of the disk in the default drive. To display this directory, follow these steps:

1. At the MS OS/2 prompt, type the following command:

   ```
   dir
   ```

2. Press the ENTER key. If necessary, you can stop the list from scrolling by pressing CONTROL+S. To view the rest of the display, you simply press CONTROL+Q.

   Your screen should look similar to this:

MS OS/2 directory

You can also get information about a specific file on your disk by typing the **dir** command followed by a filename. For example, to display directory information for a file named SCHEDULE.TXT, you would use the following command line:

```
dir schedule.txt
```

The **dir** command then would display the filename, the file's size in bytes, and the date and time it was created or last changed:

```
SCHEDULE TXT      3698      8-5-87     4:11p
```

Using Multilevel Directories

If you share your computer with other people, or if you are working on several different projects, the number of files in the root directory can become large and unwieldy. To deal with this large number of files, you may want to keep your files separate from a coworker's, or organize your files into convenient categories.

In an office, you can separate and organize files that belong to different people or that relate to specific projects by putting them in different file cabinets. For example, you might put your accounting records in one file cabinet and your letters in another. You can organize information the same way with MS OS/2 by putting your files in different directories.

Directories let you organize your files into convenient groups. These directories, in turn, may contain other directories (subdirectories). This organized file structure is called a multilevel or hierarchical directory system.

The root directory

The first level in a multilevel directory is the root directory. This is the directory that is automatically created by the **format** command. The root directory is represented by a blackslash (\). You can create directories and subdirectories within the root directory.

As you create new directories for groups of related files, the directory system grows. And within each new directory, you can add new files or create new subdirectories.

You can move around in the multilevel system by starting at the root and traveling through intermediate subdirectories to find a specific file. Conversely, you can start anywhere within the file system and travel toward the root. Or you can go directly to any directory without traveling through intermediate directories.

Your working directory

On a given disk, the directory that you are working in is called the working directory. When you start your computer, you begin in the working directory. Similarly, when you create a file, you create it in the working directory.

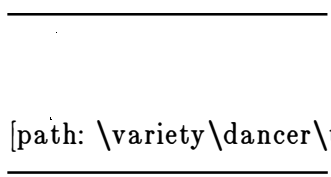
Using More than One Directory

Because your hard disk can hold so many files, it's a good idea to create multiple directories. You'll want to think about how your files can be organized into manageable groups. For example, you may want to separate your home finance files from your correspondence files. So you could put the finance files in a directory called BUDGET and your correspondence in another directory called LETTERS.

Hint Although the size of each directory will vary, you may want to try to keep each one to about 20 files initially. This way, when you use the **dir** command, the directory listing of all 20 files will fit on one screen.

Often, you will find that the files in a directory could be further divided into subgroups.

You can move around in the multilevel directory system by starting at the root directory and traveling through intermediate subdirectories to find a specific file. The route that you take to find a file is called a path. For example, the following figure shows the path \VARIETY\DANCER\TAP (the first backslash represents the root directory).



Changing Directories

Using the Chdir command

If you want to work in a different directory, you can use the **chdir** command to change directories. For example, if you were in the root directory and wanted to work in the \VARIETY\DANCER\TAP directory, you would type this command line:

```
chdir \variety\dancer\tap
```

As another example, if you wanted to change from your current working directory to a subdirectory called RERUNS, you would do the following:

1. Type the following command line:

   ```
   chdir reruns
   ```

2. Press the ENTER key.

Shortcut The **chdir** command can be abbreviated to **cd**.

Planning Directories

Directories created by the Install program

When you install MS OS/2, the Install program creates several directories for MS OS/2 system files. The following list describes the files found in each directory Install creates:

Directory	Files
\OS2	Command-processor files
\OS2\BIN	Executable (.EXE) commands that run in both real and protected modes.
\OS2\RBIN	Executable commands that run in real mode only
\OS2\PBIN	Executable commands that run in protected mode only
\OS2\DEV	Installable device-driver files
\OS2\LIB	Dynalink library (.DLL) files

The Install program also creates these four files and places them in the root directory:

- CONFIG.SYS
- AUTOEXEC.BAT
- INITENV.CMD
- STARTUP.CMD

For information about the Install program, see the *Microsoft Operating System/2 Setup Guide.*

Suggestions for other directories

Here are some suggestions for other subdirectories you can create on your hard disk:

- Create one directory for each application you want to run. Use short names that resemble the names of the applications.
- If you are working on more than one major project, each of which will require several files, create a directory for each project.
- If you are sharing the computer with other people, create one "home" directory for each user. Use the person's first name as the name of his or her directory.

Creating a Directory

Using the Mkdir command

You can create directories with the **mkdir** (make directory) command. For example, if you wanted to create a directory named GOURMET, you would type the following command line:

```
mkdir gourmet
```

The GOURMET directory then would exist as a subdirectory of your working directory. If your working directory were your root directory, the path for the new directory would be \GOURMET. MS OS/2 automatically creates the "." and ".." entries in the new directory. These two entries are shorthand notations for the working directory (.), and the parent directory (..). A parent directory is any directory that contains subdirectories. In this case, the root directory is the parent directory.

You now have two directories, your root directory and the new directory called GOURMET. Now you can create and copy files to your new directory. You can also change your working directory to your new directory.

As another example, suppose you want to create a subdirectory called COMEDYHR in the root directory. To do so you would follow these steps:

1. Make sure the root directory is your working directory by typing this command line:

```
cd \
```

2. Type the **mkdir** command, followed by the name of the directory you want to create:

```
mkdir comedyhr
```

A directory name, like a filename, may be up to eight characters long. However, directory names cannot have extensions.

Shortcut The **mkdir** command can be abbreviated to **md**.

Removing a Directory

Using the Rmdir command

When you find that you no longer need a directory, you can use the **rmdir** command to remove it. Before you remove a directory, though, the directory must be empty. Then, from a directory other than the one you're deleting, type **rmdir** followed by the directory name. For example, the following command would remove an empty directory called FALL86 from the disk:

```
rmdir fall86
```

As another example, if you wanted to remove a directory called BADPLOTS from your root directory, you would do the following:

1. Remove all files from the BADPLOTS directory. Use the **copy** or **del** command as appropriate.
2. Make sure that the BADPLOTS directory is empty except for the working directory (.) and parent directory (..) entries. (These entries cannot be deleted.) If

a directory contains any files or subdirectories, MS OS/2 will not remove it. Type the following command line to view the directory listing:

```
dir \badplots
```

3 Change your working directory to the root directory by typing the following:

```
cd \
```

4 Remove the BADPLOTS directory by typing this command:

```
rmdir badplots
```

Shortcut The **rmdir** command can be abbreviated to **rd**.

Listing a Disk's Directories

Using the Tree command

The **tree** command tells MS OS/2 to display the names of all directories and subdirectories on your computer. To see which directories and subdirectories are on your system, simply type the following:

```
tree
```

The **tree** command displays a report that looks similar to this:

[Tree command: list of directories]

For more information about directories, see the *Microsoft Operating System/2 User's Reference.*

Summary

The main points of this chapter include the following:

- MS OS/2 uses a system of multilevel directories to help you organize your files into logical groups.
- The first directory on a disk is called the root directory. Every other directory created on the disk is a subdirectory of the root directory.
- You can use these commands with directories:

Command	Action
dir	Lists the files contained in a directory.
chdir	Changes the working directory.
mkdir	Creates a new subdirectory in the working directory.
rmdir	Removes an empty directory from the disk.
tree	Lists all the directories on a disk.

Once you've created files on your computer, you'll want to protect them. In the final chapter, you will find some instructions for keeping the files on your computer safe.

6 Protecting Your Files

You can store hundreds of pages of information on a single floppy disk. If that disk is destroyed, all of your information could be lost—unless you are prepared. This chapter recommends some simple techniques that help keep your data safe.

In this chapter, you will learn the following:

- How to develop a regular routine of backing up the files on your hard disk
- How to make copies of often-used files and floppy disks
- How to use the **chkdsk** command to make sure you have enough disk space to run all the programs you want to run

Making Backups

Using the Backup command

In case something should happen to the files on the hard disk of your computer, you should make backup copies of the files on a floppy disk by using the **backup** command. For example, if you wanted to make a backup of a file called VARIETY.SHO on drive C to a floppy disk in drive A, you would type the following:

```
backup c:variety.sho a:
```

The **backup** command also lets you back up everything in your working directory. To make backup copies of all the files in your working directory on drive C to a floppy disk you place in drive A, type this command line:

```
backup c: a:
```

If you need more than one floppy disk to hold the backup copies, MS OS/2 prompts you to insert another disk when the first disk is full.

As another example, suppose you want to back up every file in every directory on your hard disk (drive C). To do so, you would follow these steps:

1. Make sure you have plenty of formatted floppy disks. If you are not sure how many floppy disks you'll need, use the **chkdsk** command. This command will show you how much space is used by the files on drive C. Plan on using a little more space to back up these files. For information on **chkdsk**, see the section called "Checking for Disk Space" later in this chapter.
2. Place the first formatted floppy disk in drive A.
3. Type this command line:

   ```
   backup c: a: /s /l
   ```

4. Press the ENTER key.

The **/s** switch tells MS OS/2 to back up data in all of your subdirectories. The **/l** switch creates a file called BACKUP.LOG in the root directory of your source disk (drive C in this case). The first line of the file shows the time and date of the backup; the remaining lines list the files being backed up. You can use this information when

How often to make backups

you need to restore a particular file from a floppy disk. Be sure to make backups on a regular basis. If you use your computer 10 to 20 hours each week, you should probably make a full backup of your hard disk at least once each month. If you use your computer less often, make a full backup at least every other month.

For more information about the **backup** command and the switches you can use with it, see the *Microsoft Operating System/2 User's Reference.*

Restoring Backup Files

Using backup files

To use the backup files, you must restore them to a disk. For example, if you wanted to restore all of the files you backed up in the previous example to a hard disk on drive D, you could type the following:

```
restore a: d: /s
```

This command reads all the files on your backup disks in drive A and writes these files to drive D. The **/s** switch tells MS OS/2 to restore all subdirectories it finds.

As another example, suppose you had backed up a directory called CARTOONS to floppy disks and then removed that directory from your hard disk on drive C. Now, suppose you have re-created the CARTOONS directory with the **mkdir** command and you want to restore the directory's files. To do so, you would do the following:

1. Place the first floppy disk containing the backup files in drive A.
2. Type the following command:

   ```
   restore a:\cartoons c:
   ```

3. Press the ENTER key.

Copying Files and Disks

How backups and copies differ

Another method of keeping your data safe is to copy files to another disk with the **copy** command. Copies sometimes take more space than backups, but copies are usable "as is." That is, unlike those you create with **backup**, you do not have to restore files that you create with the **copy** command in order to use them.

As an example of how to use the **copy** command, if you wanted to copy a file called SITCOM.SHO from a disk in drive C to a disk in drive B, you would type this command line:

```
copy c:sitcom.sho b:
```

When you make a duplicate copy, it is possible to give the duplicate a new name. For example, if you wanted to copy a file called QUIZ.SHO from drive B to drive A, and you wanted to name the duplicate GAME.SHO, you would type this:

```
copy b:quiz.sho a:game.sho
```

Some Practical Uses for Copies

You will find many practical uses for duplicating files with the **copy**, **xcopy**, and **diskcopy** commands. The next few sections will discuss three of the more common uses:

- Supplementing backups
- Making duplicates on floppy disks to share between two people or to transfer between two computers
- Duplicating an original floppy disk for safekeeping

Using Copy to Supplement Backups

You should make copies of your data files more frequently than you back them up. For example, suppose you are going to work on a project with a deadline in two weeks. Over the next two weeks, you will be collecting data in new files and modifying existing files repeatedly. All the files that you are using have an extension of .FIL and are located on drive C in a directory called PROJECT. Before you begin working on your project, make a copy of all the files in the PROJECT directory on a floppy disk in drive A with this command line:

```
copy c:\project\*.* a:
```

On the first day, you modify only one file, called INTRO.FIL. At the end of the first work session, you should copy the file you've modified to the floppy disk containing the copy of your PROJECT directory. To do so, type the following:

```
copy c:\project\intro.fil a:
```

Remember that the disk on drive A already has a copy of the unchanged INTRO.FIL file on it. However, the **copy** command writes over an existing file if that file has the same name as the file being copied. Therefore, the changed INTRO.FIL file replaces the original INTRO.FIL file on the disk in drive A. Now the contents of the disk in drive A are identical to the contents of the PROJECT directory on drive C. If something should happen to the data on the disk in drive C, you could use this current copy to replace that information.

Note Because the **copy** command will write over an existing file if the existing file and the copied file have the same name, you must be certain you are copying a new version of the file over an old version, and not vice versa. As an extra precaution, you may want to rename existing files with the extension .OLD before copying new versions to the same disk.

Using Copy to Share Files

One convenient use for the **copy** command is to copy each subdirectory on your computer to an individual floppy disk. This gives you a second usable copy of all your files and keeps them in a familiar organization.

You must type a separate **copy** command for each subdirectory you want to copy. Unlike the **backup** command, **copy** will not copy files from your hard disk to a series of floppy disks. Once a disk is filled, MS OS/2 displays an error message and the **copy** command ends. To copy other files, you would need to type another command specifying the remaining file or group of files, and you would need another floppy disk.

As an example, to copy all the files in a subdirectory called ALVIN, on drive C to a floppy disk in drive A, type the following:

```
copy c:\alvin\*.* a:
```

The asterisk (*) wildcard means "all." In this example, *.* tells MS OS/2 to copy all filenames with all extensions to drive A. Because you did not specify any

filenames or extensions for the target files (which are on drive A in this case), MS OS/2 uses the original names.

Using the Xcopy command

Another command, **xcopy**, allows you to copy all of the files in a directory and all of its subdirectories to another disk.

For example, suppose you have a directory on drive C called WRITE with subdirectories called MEMOS and LETTERS. If you wanted to copy the directories to another disk, you would first check to see that the contents of WRITE, including the contents of its subdirectories, will fit on the target disk. If the directory and all its subdirectories can fit on the disk, you would type the following:

```
xcopy c:\write\*.* a: /s
```

The **/s** switch tells MS OS/2 to copy all the subdirectories found in the WRITE directory.

Making Duplicates of Floppy Disks

If you have programs or data files that you maintain on floppy disks, you should be sure to make duplicates frequently.

Using the Diskcopy command

If you want to make an exact duplicate of a floppy disk, you can use the **diskcopy** command to copy files track-by-track from one disk to another disk of the same type. For example, to copy every file from a disk in drive A to a formatted disk in drive B, you would type the following:

```
diskcopy a: b:
```

Comparing Copies of a File

Using the Comp command

You can confirm that your working and copied files are identical by using the **comp** command. For example, to compare two files named PLANNED.SHO and PROGRAMD.SHO, you would type this command line:

```
comp planned.sho programd.sho
```

If the two files are identical, the **comp** command displays this message:

```
Files compare OK
```

An error message appears if the files are not identical.

You can also compare groups of files. For example, if you wanted to know whether the contents of two directories called NEWSEASN and FALL88 were identical, you would type the following command line to compare them file by file:

```
comp \newseasn \fall88
```

Comp lists the name of each filename as each pair of files is compared. After comparing each pair, **comp** tells whether the two files are identical. **Comp** stops comparing files if it reaches its limit of ten mismatches.

As another example, suppose you want to compare the contents of a directory on drive C called EDUCATOR with a floppy disk whose contents should be identical to the EDUCATOR directory. Rather than visually comparing each file's creation date and size, you could do the following:

1. Place the floppy disk containing the copy of the directory in drive A.
2. Type the following command line:

```
comp c:\educator a:
```

The **comp** command examines each pair of files to determine whether or not they're the same.

Checking for Disk Space

Using the Chkdsk command

It is important to check the resources available on your computer. This is particularly true when you want to do the following:

- Run several programs on your computer at the same time
- Use a program that requires a lot of memory to run
- Run a program using very large data files

You should find out how much memory and disk space is required to run each application or other process you have on your computer. Applications that include graphics, for example, tend to use a large amount of computer resources. If you run out of memory or disk space while running a process, you can lose the data that you

were working with. If you are running several processes when you run out of memory or disk space, the problem compounds.

The **chkdsk** command allows you to check your computer's resources. If you want to check the amount of disk space available on drive C, type the following command:

```
chkdsk c:
```

MS OS/2 responds with a message similar to this one:

```
31768576 bytes total disk space
 45056 bytes in 2 hidden files
 34816 bytes in 15 directories
6895616 bytes in 462 user files
24793088 bytes available on disk
```

The last line shows you the amount of disk space available to you.

Chkdsk also checks your disk for any errors. If it finds any errors, such as a bad sector, **chkdsk** displays an error message. If you find an error on one of your disks, you can use the **chkdsk /f** switch to fix the error. For more information about this option, see the *Microsoft Operating System/2 User's Reference.*

Summary

The main points of this chapter included the following:

- You should make regular backups of your files. If you use your computer more than 10 hours per week, you should probably make backups at least once a month.
- The **copy** command can create a duplicate of your data files on floppy disks for safekeeping.
- Make a habit of using the **chkdsk** command to learn how much space is available on your hard disk. If you run out of space, you could lose data.

This manual has just touched the surface of what you can do with the MS OS/2 operating system. To learn more about MS OS/2, see the *Microsoft Operating System/2 User's Reference.*

Index