

VERY EASY DOS

A simplified training manual
for novice users of IBM or
IBM-compatible computers with
a hard disk and PC or MS DOS
Version 2, 3, 4, or 5

If DOS seems confusing, frustrating, or intimidating, then this is the book for you.

I wrote this step-by-step tutorial to help you save time, have fun, avoid frustration, and hopefully end up with a warm, fuzzy feeling about DOS. This is indeed possible because *Very Easy DOS* will not expose you to any unnecessary detail.

This 2nd version of *Very Easy DOS* is 1/3 longer than the first version. The extra pages are the result of longer explanations, additional personal opinions, and an Index. The essential DOS topics comprise less than half of this manual. At the start of each Section, I'll let you know which parts are essential and which parts you can skip. I've written the lessons so that, without wrecking anything, you can do them on your own computer at home, or on someone else's computer, or on the computer at work.

In any case, I'd like to have your opinion of it.
My email address is JOHNZORICH@YAHOO.COM

Many thanks to those who helped me with this second version. Most of all, I thank my wife.
--

Section #1

Getting Acquainted

This manual describes a very easy way to use DOS. To make it even easier, only the essentials parts of DOS will be discussed; the rest of DOS will be completely ignored. The essentials will be taught in a simplified style, instead of the overkill of detail that other DOS books use. With *Very Easy DOS*, you'll feel fully trained in hours instead of months.

My assumptions about you are that you have an IBM or IBM-compatible computer with PC or MS DOS and a hard disk. If your computer does not have a hard disk, then this is the wrong manual for you; DOS is very easy only if you have a hard disk. Your DOS version number must be at least 2.0. Starting with version 4.0, you have the choice of using DOS either the way it's described in this manual, or by using a menu "shell". This manual does not discuss such a "shell" because if you learn this manual, you won't need a DOS shell.

If you are working on a computer network and/or are running sophisticated, specialized software packages, you'll want to learn a few more DOS commands than the couple dozen that this manual discusses. However, if you're only using one computer, with the usual **application program** software (such as a wordprocessor, spreadsheet, database, or game), then all you really need is this manual.

I assume you know the basics of a computer, namely how to turn yours on, how to insert floppy disks, and how to find your way around a keyboard. I assume you know the difference between the terms "software" and "hardware" (at least to be able to point to each if you had them in front of you). If you know all that, you are ready to learn DOS.

What the heck is DOS???

DOS is just another computer software program. It's the program that runs the computer when you aren't running an application program. However, in order for an application program to work, it needs DOS. Your computer and your application program both need DOS.

DOS is an acronym for the computer's "**Disk Operating System**" program. It's the software program that (1) sets up and maintains the information storage and retrieval system on the floppy and hard disks, and (2) mediates all other hardware-software interactions. Your application program, not DOS, does most everything else.

Microsoft Corporation wrote the version of DOS that IBM chose for its first line of personal computers. That version is referred to by the acronyms PCDOS (the version needed by IBM computers) or MSDOS (the version needed by IBM-compatible "clones"). Major software stores sell PCDOS as well as a generic version of MSDOS. IBM-compatible computers made by certain manufacturers need a customized version of MSDOS to work correctly; usually, such versions are provided free when the computer is purchased.

There are "disk operating systems" other than PC and MS DOS; for example, UNIX and OS/2. A non-Microsoft clone of MSDOS is also available; it's called DRDOS. However, the term "DOS" by itself has come to refer only to the sequence of programs provided by Microsoft for use on IBM and IBM-compatible "personal computers".

In addition to the essential disk-and-hardware-operating- instruction software, DOS comes with other miscellaneous software programs. Most of these other programs are just useful "tools" that either help you make better use of certain types of hardware or help you save time on difficult tasks.

The version of DOS that came with my computer had 68 such software "tools" (otherwise known as "files" or "utilities"). During the past 3 years, I've found use for only 15 of the 68. Only 3 of those 15 are really essential. Two of those 3 are no longer important after DOS is placed onto my hard disk and my blank floppy disks are initialized. For daily work, I absolutely need only 1 DOS file out of all 68 that came with my version of DOS.

DOS calls that one file `COMMAND.COM`. This manual will discuss how to use some of the other files, but the bulk of what you read here will be about `COMMAND.COM`.

DOS Versions

Microsoft Corporation has updated (upgraded) and expanded DOS many times since providing IBM with the first version. Later versions have all the power of earlier versions, plus more. A later version will have a larger version number than an earlier one. For example, version 2.0 was introduced in 1983, whereas version 5.0 came out in 1991.

"DOS 1.X" refers to version 1.0, 1.1, and all other versions starting with "1". Similarly, 2.X refers to all version 2's, 3.X to all version 3's, and so on. All software manufacturers, not just Microsoft, use this same numbering pattern for all their software, not just for DOS.

Don't be fooled into thinking that the more recent a version of DOS you have the better. Recent versions of DOS may actually interfere with running application programs. To illustrate that, let's examine my own situation.

First, let me explain a term I'll be using in my explanation. The term is **RAM**, which stands for **Random Access Memory**. RAM is the space on the computer's temporary memory chips that must always be large enough to store the following 3 things:

- 1) the essentials of DOS
- 2) the application program(s) you're using at the moment
- 3) the memo and/or letter and/or spreadsheet (and/or whatever) that you're working on at the moment

On older computers, DOS limits your available RAM to 640K maximum. The **K** stands for approximately 1000 bytes of storage space. Each **byte** of memory is enough space to store one single computer instruction or keyboard character. Newer computers have the potential to add extra RAM of different types, but DOS still limits the "conventional" type of RAM to 640K. RAM added above 640K can be used only by certain programs; the above-640K RAM is useless to many programs but essential to others.

RAM is located on memory chips installed inside your computer. Some computers have only 512K worth of RAM chips installed; some older computers have only 256K RAM; and some of the earliest IBM PC's have only 64K. The

owners of such computers may be able to easily upgrade to a full 640K of "conventional" RAM; ask how at a computer store.

Now, back to my explanation:

I have a clone of the earliest version of an IBM PC that had a hard disk (it came on the market in 1983). Such computers need DOS 2.0 or higher to be run correctly. Version 2.0 uses about 40K of RAM just for itself. My computer came with version 3.21, which requires about 60K of RAM for itself. Therefore, because I have version 3.21 instead of 2.0, I have about 20K less RAM available for applications. Not having that 20K has prevented me from using some large programs.

What do I receive in compensation for the 20K of RAM that I lose because I have version 3.21 instead of version 2.0? I gain the ability to run my computer on a network system and to handle several different types of floppy disk drives. My computer has a single, old-style floppy drive; and I am not connected to a computer network. Version 3.21 is not much help to me, here, at home, where I've set up my "computer room" (as my wife calls it) next to the washer and dryer.

DOS version 3.0 was created to run the AT versions of IBM computers (the 80286 machines) and the higher capacity floppy drives that came with them. Version 3.2 allowed the use of 3½ inch floppy drives. For computers with hard disks larger than 32MB, DOS 3.3 provided significant advantages over earlier versions. (**MB** stands for **MegaByte**; one MB is equal to 1000K or approximately one million bytes). These are not the only changes that occurred with each new addition.

DOS 4.0 (and above) allows you to use an entirely different command concept, referred to as a DOS "shell" or "menu system". A menu is designed to allow you to give commands by choosing from options already typed out for you on the screen. Version 4.0, even without the shell turned on, gobbles up more RAM than earlier versions.

I won't be discussing the DOS "shell" in this manual. If you know of and want a "menu" type of interaction with DOS, you don't have to upgrade your DOS version. There are very good DOS menu programs by companies other than Microsoft. Such programs can be turned on and off as desired, so you don't lose

any RAM that you may need for other programs. In addition, most new wordprocessor, spreadsheet, and database application programs have a simple DOS menu as one of their command options. A wonderful DOS menu program came with my mouse (a hand held instrument that moves the cursor around on the computer screen); and another menu system is part of the program I'm using to type this manual!

If you have one of the newer computers that permits you to have more than 640K of RAM, and if you have a large enough hard disk, you may enjoy using an elaborate DOS shell referred to as "Windows". This shell (as well as the ones mentioned above) allows you to give commands to DOS by using a mouse. You use the mouse to interact with a GUI (graphical user interface) where you manipulate words or icons (little drawings) instead of typing commands. Such a "visual environment" was popularized by the Macintosh line of Apple computers. To take full advantage of a Windows shell, you'll need a high-powered computer and application software written specifically for Windows use.

DOS 5 can operate the new 2.88MB floppy disk. Another notable feature of 5.0 is that it uses slightly less RAM than does 4.0. And if your computer has extra RAM added beyond 640K, DOS 5.0 can be set up to need for itself only about 20K of "conventional" RAM. That's half of what even DOS 2.0 needs. What a great improvement!

DOS 6 hasn't been released yet, but a spokesperson for Microsoft recently announced that Version 6 would combine features of DOS, Windows, and OS/2. DOS 6 is to be available in 2 flavors: one for older, less-powerful computers, and one for newer, high-powered ones.

DOS Instruction Manuals

I bought my first computer not knowing a thing about DOS or computers. To my dismay, I ended up with a huge pile of books to read, even before I'd unpacked the computer!

The USER'S GUIDE provided by my computer's manufacturer was only 164 pages long. But the DOS manual itself was 352 pages, and the GW-BASIC one another 347 pages. Then there was the 191-page USER'S MANUAL for the printer and the 756-page book one of my well-intentioned friends had suggested I

buy to help explain the 352-page DOS manual. Finally, there was the 1275-page "DOS Power Tools" book that another friend said would help explain the 756-page DOS book that I had bought to explain the 352-page manual. If I had wanted, I could have also purchased a 2 volume, 6-inch-thick set called "The Complete DOS Encyclopedia". These books are so thick, you end up using them for small children to sit on at the dinner table.

After several months of study, I discovered this: to use a computer well, I do not need to be a speed-demon POWER USER with a library of DOS books. The reason is this: after I've turned on an application program, most of the DOS I know is useless. While I am using an application program, that program itself secretly takes care of any DOS commands it needs. I'm not involved at all. Isn't that great!

All that a computer does is to run application programs. All that DOS does is to allow you to run those programs. What follows in this manual is all you need to know to run DOS effectively and efficiently in the real world inhabited by us tortoises. Leave the 6-inch-thick manuals for the speed-demon rabbits.

Ignorance is not Stupidity

Let me make one last point before we start working. Strictly speaking, you don't need to know anything about DOS to use a computer, as long as someone else has set up the computer for you.

My mother, for example, is an executive secretary at the home office of a large banking corporation. When she wants to type letters and reports, she turns on her computer, and the wordprocessor program automatically starts for her.

Another example is a friend of mine who is the R&D librarian for a biotechnology company. He is constantly accessing research databases by computer. He, too, knew nothing about DOS (before he read this manual). He used to just turn on the power, wait for all the usual DOS messages to stop, and then type in a single word to access whichever database he needed.

Both my mother and my friend are using the computer to its fullest advantage, for the applications that are of interest to them. In this sense, you already know as

much DOS as you need to. However, if you know a little more, the little more that I'll show you in this manual, you won't need "someone else" to set up the computer for you or to make changes every time you obtain a new application program. You'll be able to do it yourself.

Important First Instruction

WHILE YOU'RE WORKING THE PRACTICE LESSONS IN THIS MANUAL, DON'T TYPE ANYTHING INTO THE COMPUTER UNLESS I SPECIFICALLY TELL YOU TO DO SO. WHAT YOU SHOULD TYPE WILL BE SEPARATED FROM THE MAIN TEXT BY A BOX, AS IN THE EXAMPLE ON THE NEXT PAGE:

THIS IS WHAT YOU WOULD TYPE THERE MAY BE MORE THAN ONE LINE YOU SHOULD TYPE EACH LINE IN THE SAME ORDER AS I DO
--

IF IT'S NOT IN A BOX LIKE THE ONE ABOVE, THEN DON'T TYPE IT. IF IT IS IN A BOX LIKE THE ONE ABOVE, YOU SHOULD TYPE THE ENTIRE LINE OR LINES THAT ARE LISTED INSIDE THE BOX. IT WON'T MATTER IF YOU USE CAPITAL OR SMALL LETTERS OR A COMBINATION OF THE TWO. I WILL USE CAPITAL LETTERS MOST OF THE TIME, BUT YOU DON'T HAVE TO.

TWO EXCEPTIONS TO THE ABOVE RULE:

I HAVE PUT THE TERM **enterkey** AT THE END OF EACH LINE. IT MEANS TO PUSH THE *enter* OR *return* KEY, NOT TO TYPE OUT THE WORD.

WHEN I LIST THE TERM **F6key**, IT MEANS TO PRESS THE *F6 key*, NOT TO TYPE OUT THE WORD.

THE FOLLOWING IS ANOTHER EXAMPLE, SO PLEASE DON'T TYPE IT. IF IT WERE NOT AN EXAMPLE, YOU'D TYPE OUT COMPLETELY "COMMAND NUMBER ONE" AND THEN PRESS THE "enter" KEY ON YOUR KEYBOARD. NEXT YOU'D TYPE OUT "COMMAND NUMBER TWO" AND THEN PRESS THE "enter" KEY AGAIN. FINALLY, YOU'D

PRESS THE "F6" KEY ON YOUR KEYBOARD, FOLLOWED BY PRESSING THE "enter" KEY.

COMMAND NUMBER ONE	enterkey
COMMAND NUMBER TWO	enterkey
F6key	enterkey

IF YOU CAN DO THAT, YOU'RE READY TO BEGIN.

Section #2

Hard Disk Organizing

by DOS

IN THIS SECTION YOU'LL LEARN HOW TO ORGANIZE YOUR HARD DISK BY PARTITIONING AND FORMATTING IT (I'LL EXPLAIN WHAT THAT MEANS ON THE NEXT FEW PAGES). WE WILL DISCUSS ONLY ONE METHOD FOR DOING THIS. THERE ARE OTHER METHODS, BUT THE ONE DISCUSSED HERE IS BY FAR THE MOST COMMON.

WARNING !!

ORGANIZING THE HARD DISK AS DESCRIBED IN THIS SECTION WILL ERASE ALL DATA AND PROGRAMS ALREADY STORED ON IT. IF YOU HAVE ANY DOUBTS ABOUT WHAT'S ON THE HARD DISK, ASK SOMEONE WHO KNOWS, NOW!

IF YOUR COMPUTER IS ALREADY RUNNING APPLICATION PROGRAMS, THERE IS NO PRACTICAL REASON YOU NEED TO DO ANY OF THIS SECTION. EVEN IF ALL YOU CAN DO IS RUN A GAME OR TYPE A MEMO, IT MEANS THAT SOMEONE HAS ALREADY DONE THIS SECTION FOR YOU. IF THAT IS THE CASE, SKIP THIS SECTION (YOU MAY WANT TO READ IT, FOR EDUCATIONAL PURPOSES ONLY) AND START FOLLOWING INSTRUCTIONS AGAIN AT THE BEGINNING OF SECTION #3.

WHEN YOU TRY TO DO WHAT I INSTRUCT BELOW, YOUR COMPUTER MAY STOP, BEEP, AND TELL YOU IT CAN'T DO WHAT YOU'RE ASKING. WHAT THAT MEANS IS THAT YOUR COMPUTER'S MANUFACTURER HAS PROVIDED A SPECIAL SET OF COMMANDS (INSTEAD OF THE USUAL ONES) TO ACCOMPLISH PARTITIONING AND FORMATTING THE HARD DISK. NO BIG DEAL. JUST ASK A COMPUTER-LITERATE FRIEND TO READ THE APPROPRIATE SECTION IN THE COMPUTER-OWNER'S MANUAL THAT CAME WITH YOUR COMPUTER, AND DO WHAT IT SAYS THERE INSTEAD OF WHAT I SAY IN THIS SECTION. AFTERWARDS, WHEN YOU'RE FINISHED PARTITIONING AND FORMATTING YOUR HARD DISK, CONTINUE IN

THIS MANUAL STARTING WITH SECTION #3.

DOS 4.0 AND ABOVE ARE INSTALLED ONTO THE HARD DISK BY A SELF-GUIDED, AUTOMATIC INSTALLATION PROCEDURE. TO START THE PROCEDURE, CHECK THE "GETTING STARTED" MANUAL THAT CAME WITH YOUR VERSION. DURING THAT PROCEDURE, DOS ASKS YOU IF YOU WANT TO PARTITION THE HARD DISK. IF YOU ANSWER "YES", DOS AUTOMATICALLY STARTS WHAT IS DESCRIBED IN THIS SECTION.

A 30 megabyte hard disk, such as I have, can store information equivalent to over 10,000 typed sheets of paper! On a new computer, all those sheets are blank (so to speak). Before the computer starts to store information on these sheets, it first needs to organize them; this is like dividing a stack of blank pages into separate volumes and chapters.

Organizing the hard disk is a 2-step process. There are many options; for the average user of a single computer, there is really only one useful option. This Section describes it.

Find your floppy-disk copy of the DOS software. If your DOS version has more than one disk, the disk you want here will be labeled "Disk #1" (or it might be called "Program Disk"). If DOS did not come free with your computer, go to the store that sold you the computer and purchase a version suitable for your machine and your intended use (keeping in mind the discussion of DOS versions that you read in Section #1 above).

Some computers have not one but two floppy-disk drives. The computer considers one of them to be the "A: drive" and the other to be the "B: drive". (If you have 2 floppy drives, refer to your owner's manual as to which is which.) Similarly, some computers have more than one hard disk; the first is referred to as the "C: drive", and the other as the "D: drive". DOS 3.3 or higher can be used to trick the computer into thinking that different parts of a single hard disk are separate drives; that trick won't be discussed in this manual.

For ease of discussion, I'll always assume that the floppy drive is the A: drive, and that the hard disk is the C: drive. Using the other drives (if you have them) is analogous to using the A: and C: drives. Any of the commands I discuss that

require you to mention a drive letter (A: or C:) can be used with the other drives (B:, D:, E:, ...) by substituting the appropriate drive letter. If this is all just too confusing, don't worry; later, you'll understand what I'm "drive"-ing at.

Now let's start to work.

Put DOS Disk #1 into the floppy drive (drive A:) and close the door (if it has a door). Turn off the power to your computer, take a deep breath, and then turn it back on.

DOS will soon provide you with a date and time. If what DOS shows you is correct, just press the ENTER key to continue. If it is not correct, you can either press ENTER (which means you don't care) or you can type in the correct date or time (use the same style that is displayed on the screen). In both cases, press the ENTER key after typing. Having the computer know the correct day is useful if you have a wordprocessor that will automatically put "today's date" on your memos, letters, *etc.* Your computer may have internal hardware such that, once you have given it the correct date and time, you may never have to tell it again.

When DOS is finished setting up the computer, it will display the following symbol: A>. That symbol (or C>) is one of many possible forms of the DOS **command line prompt**, or **prompt** for short. Its presence as the last thing on the screen means that the DOS program is in full control of the computer screen, as opposed to when DOS gives up control to application programs. A prompt doesn't signify anything other than that the DOS software program is waiting for you to issue it a command. You issue it a command by typing the command and then pressing the ENTER key.

The blank space next to the prompt is where you type commands. That blank space is called the **command line**. You're allowed to type **DOS commands** on the command line, one per line. (In DOS 5, a special command allows you to type more than one command per line.) After you press the ENTER key, that one command is executed by DOS. In addition to issuing DOS commands on the command line, you can also issue application-program startup commands (which turn on your wordprocessor, spreadsheet, game, *etc.*). Such startup commands must obey some of the same rules as DOS commands.

It is important to realize that from the moment you start (most) application

programs, the DOS command line and the DOS command-line-prompt are no longer present. "Commands" that you give from that point on are not DOS commands but commands that are unique to the application program that is in control of the screen (whatever application program you've turned on). DOS is still working, but secretly, behind the scenes; any DOS command that is needed is given by the application program, not by you.

FDISK

At this point, type the following DOS command ("enterkey" means to press the ENTER key, not to type out the word):

```
FDISK enterkey
```

You will now see on the screen a list of options, with a number in front of each option. You need to examine the options to find which one says "Create MS-DOS Partition" or "Install MS-DOS Partition" or something like that. Type whatever number is in front of that line, verify that it has shown up in the bracket following the line that reads something like "Enter choice" or "Enter your choice", and then press the ENTER key.

A message might come on the screen saying that an MS-DOS partition already exists. What the message means is that someone has already done this step for you, possibly the factory, possibly the computer store. If that message shows up, press the ESC key and then proceed directly to the FORMAT topic below. If that message did not show up, continue on here. If you are unsure or uncomfortable, shut the computer off and get advice from a computer-literate friend.

You'll now be asked on-screen whether you want to use the entire fixed disk (the hard disk) for MS/PCDOS. If you had an additional disk operating system (and you don't), you would want to use less than the entire disk for DOS. So just type:

```
Y enterkey
```

The computer will now do its thing, organizing large general areas on the hard disk. What happens next depends on your DOS version. DOS 3.3 as well as the automatic installation of DOS 4 and 5 are explained below (skip the next paragraph if you have version 3.3 or above).

Versions earlier than 3.3 will come onscreen with a message telling you to put the DOS disk back into drive A: (the floppy drive) and to press any key. You probably still have the DOS disk in the floppy drive (with the door closed, if there is one), so just press any key. The computer will now re-start itself, as it did on page 10.

Whenever it says to press "any key", I always press either the SPACE-BAR or the ENTER key. There are a few keys that won't work in these situations, and I can't always remember which ones they are. Those two keys always work.

DOS 3.3X and 4.X will return you to the main FDISK menu. When that happens, press the number in front of the line that says something like "set active partition". By doing so, you re-confirm that you want MS or PC DOS to be in charge of the computer (instead of some other disk operating system). When the computer again stops, press the ESCAPE key. DOS 4 will then continue with automatic installation.

DOS 5 will "set active partition" automatically, and then proceed with the rest of the automatic installation.

FORMAT

Now that you have used the command FDISK to organize the large general areas on your hard disk, you have to divide these large areas into small areas. Those of you not using the automatic installation of DOS 4 and 5, type the following:

```
FORMAT C: /S /V  enterkey
```

What you just typed translates as "divide (FORMAT) my hard disk (C:) into thousands of small areas and then place the DOS System files (/S) onto the hard disk so I won't have to keep using this DOS floppy disk to start the computer, and let me give a volume name (/V) to my hard disk". In Section #3, I'll explain what a "file" is. "Volume" is what some programmer somewhere thought was a good name for the contents of a disk.

The "system files" are the 3 pieces of information that constitute the essential

DOS commands needed by you and the computer. Two of these 3 files will be "invisible" to you. They're called invisible because you won't be able to use the DIR command to find them among all the other files (you'll learn about that command in Section #3).

A warning will show up now on the screen; it says that all the data on the non-removable disk (this means your hard disk) will be lost. If you are using someone else's computer, stop now and ask whether erasing it is OK. If you have a brand new computer, you likely don't have anything on the hard disk. However, some computers are sold with several application programs already loaded onto the hard disk. If you don't know for sure what's on your disk, and feel uncomfortable about it, just turn the computer off. Then, phone the computer store that sold you it, and ask them. To continue with formatting, type:

Y enterkey

The computer will do its thing for quite a while (possibly many minutes!). When it's finished, a message will come onscreen saying something like "system transferred". Soon after that, you will see a message similar to "Volume Label-11 characters, ENTER for none". At that point, type in a name for your hard disk, using up to 11 characters (no spaces). For example, you could type MY-HARDDISK, and then press the ENTER key.

Hard-disk volume labels are one way you may be able to prevent yourself from accidentally re-FORMATing your hard disk (and thereby erasing everything on it). Depending on your DOS version, you won't be able to format your hard drive again unless you type in its exact name when DOS asks for it during the FORMAT process.

You're done! You can now forget everything you just learned. What you have done is to set up the computer so that its various parts can use DOS to communicate with each other. You'll never have to do it again, unless you buy a new computer or replace the hard disk.

By the way, those of you with brand-new computers no longer have to have the floppy-disk copy of DOS to start the computer. All the information you need is now on the hard disk. I will be asking all readers to use the DOS floppy at times during the upcoming Sections in this manual, but I do so only for purposes of instruction.

Section #3 Hard Disk Organizing by You

IN THIS SECTION WE WILL DISCUSS DIRECTORIES AND FILES, AND HOW THEY ARE USED. YOU WILL ALSO LEARN HOW TO USE THE FOLLOWING DOS COMMANDS: DIR, COPY, and MD.

THOSE OF YOU WITH DOS 3.3X OR EARLIER WILL NEED A FLOPPY DISK COPY OF DOS DISK #1 (IT MAY BE CALLED "PROGRAM DISK") TO DO THIS SECTION.

IF YOU DON'T HAVE SUCH A DISK, OR ARE WORKING WITH DOS 4.0 OR HIGHER, ASK A COMPUTER-LITERATE FRIEND TO FORMAT A FLOPPY WITH THE DOS "SYSTEM FILES," SO THAT IT CAN BE USED TO START THE COMPUTER. THEN ASK YOUR FRIEND TO FILL UP THAT FLOPPY WITH AS MANY DOS FILES AS WILL FIT, SPECIFICALLY THE FILES THAT HAVE COM, SYS, OR EXE EXTENSIONS; BE SURE TO INCLUDE SOME OF EACH EXTENSION. YOUR FRIEND WILL UNDERSTAND MY REQUEST.

ALL THE DOS FILES ON SUCH A FLOPPY THAT SOMEONE MAKES FOR YOU SHOULD COME ONLY FROM THE HARD DISK OF THE COMPUTER YOU'RE ABOUT USE FOR THE LESSONS BELOW. I WANT THOSE OF YOU WITH VERSION 3.3 OR EARLIER WHO DON'T HAVE AN ORIGINAL DOS DISK, AND THOSE OF YOU WITH DOS 4.0 OR LATER, TO UNDERSTAND, WHEN I REFER TO DOS DISK #1, THAT I MEAN INSTEAD THIS FLOPPY WHICH A FRIEND CREATED FOR YOU.

DIRECTORIES and FILES

So far, we have set up your hard disk so that the computer itself can find information easily. This section explains how to set up your hard disk so that you, the user, can find information easily.

On a hard or floppy disk, a particular piece of information (such as a program, game, or memo) exists in a particular physical location or set of locations. This is similar to how separate songs exist in separate locations on a phonograph record or CD. Each of these separate locations on a computer disk is called a **file**. Some files are large, whereas others are small, just as songs on a record are of different lengths.

The detailed information on how to carry out computer commands are all "inside" files, like words in a song. Files that have information on how to "do" something, such as execute a command, are called "**executable files**". Files that contain data only are called "non-executable" or "data" files. The only thing that is important for you to know about executable files is that the names for them usually end in "**com**" or "**exe**" or "**bat**". Almost all other files are data files. Any application software program you use that "does" something has at least one executable file; usually there are also at least a few data files.

I'd like you to think of these computer files as equivalent to the manila-colored file folders used in offices to hold sheets of paper together. Usually, such folders each have a unique name tag. Similarly, all computer files must have a unique name so that DOS will be able to tell them apart. DOS requires unique filenames because there can be thousands of files on a single hard disk.

To be able to "do" something, an executable file not only has to have the right contents, but it also has to have one of the name endings that DOS requires on executable files. An executable file can't "do" anything if its name doesn't have a correct ending. However, a data file can't "do" anything even if it has one of those endings. An executable file "does" something after you type its name (plus or minus the ending) onto the DOS command line and then press the ENTER key.

To help you organize files, DOS allows you to create a large number of what it calls a "**directory**". Each directory represents an "area" on the hard disk that acts like a file cabinet or file box--a surrealistic file box!! Inside each file box can be several other file boxes, or one to many file folders, or a combination of the two. Inside those boxes can be other boxes and file folders. And inside those boxes can be other boxes and file folders, *ad infinitum* (or so it may seem).

Everything inside a directory (that is, a file box) is either a file or another directory. A directory inside a directory is sometimes referred to as a **subdirectory**. The larger directory is called the **parent directory** of the

subdirectories that it contains. Notice that a directory can be both a parent directory and a subdirectory. I know this is confusing. The diagram on the next page may help (take a look at it now).

It might be helpful to think of the largest directory on a disk as the elderly patriarch of a huge extended family. He has his own children, who have their own children, who have their own children, who someday will have their own children. The official DOS name for this patriarch, this largest box on a disk, the box that contains the contents of the entire disk, is the **root directory** (I'm not making this up). His children are his subdirectories, but they are parent directories to their children (their subdirectories), and so on.

Again, I apologize for how confusing this is. This is the worst part of DOS. If you can understand the concept of directories inside directories, you have mastered the most difficult part of using DOS. We will be discussing this further.

Unlike a phone directory, DOS doesn't store your files in alphabetical order. To see what a mess it can be, put your DOS disk #1 (or the DOS copy a friend made for you) into the floppy drive and close the door (if it has a door). Then turn the power to the computer off and back on again. Press the ENTER key in response to any requests. Wait until all the activity stops and the A> symbol shows up last onscreen. Then type what's in the box below (remember, "enterkey" means to press the ENTER key; it does not mean to type the word).

DIR enterkey

DIR is a DOS command that orders the computer to list out the contents of whatever directory the computer's attention is currently on. The computer can have its attention on only one directory at a time; DOS and you are said to be "in" that directory. In this case, the computer's attention is on the root directory of what's in drive A: (the floppy-disk drive).

What you now see on the screen is a list of the files on the DOS floppy disk. Each line contains the name of a single file, how big it is, and the date and time it was created or last modified. If your version of DOS is like most, the list is so long that it can't fit on one screen, and so the first part of the list scrolls off the top out of sight. My hard disk can store the equivalent of over 80 floppy disks. What a mess it would be if I organized it the same way as this DOS disk.

DOS helps you cope with such messes by providing variations on the DIR command. The variations affect either the way the list of files is displayed or which of the files are listed. Type each DIR variation shown in the box on the next page, pausing between variations to study and compare the results. Change them slightly to see the flexibility of DIR.

I like to think that the /W and /P in the box below stand for "without the extras" and "pause between pages". It doesn't really matter what you think they stand for, as long as you remember what they do. The meaning of the asterisks will be discussed later. But for now, just type the following:

DIR		enterkey
DIR/W		enterkey
DIR/P		enterkey
DIR	*.	enterkey
DIR	*.COM	enterkey
DIR	F*.*	enterkey
DIR	*.S*	enterkey

To help create some order in a large set of files, DOS allows you to make your very own directories (boxes) in which to keep files (and other directories). DOS automatically makes the root directory for you; you have to make all other directories yourself. You even get to pick a name for each!

For example, I put all of my version 3.21 DOS files into a directory called MSDOS321, my graphics program files into a directory called DRAW, my spreadsheet program files into SPREDSHT, my database program files into DATABASE, and my wordprocessor, dictionary, and thesaurus program files all into my WORDS directory. Because I have such a directory organization, when I ask the computer (by typing DIR) for a listing of what's in the root directory of my hard disk, instead of seeing a list that scrolls on forever, I see the following:

```

COMMAND  COM
AUTOEXEC BAT
CONFIG   SYS
MSDOS321  <dir>
DRAW     <dir>
SPREDSHT <dir>
DATABASE <dir>
WORDS   <dir>

```

The "<dir>" after each name is what DOS uses to remind you which name represents a directory and which name represents a file. Any name not followed by <dir> is the name of a file.

When you give the DIR command inside any directory other than the root directory, the subsequent listing will include a single period and a set of 2 periods each followed by <dir>, like so:

```

.      <dir>
..     <dir>

```

Those really don't mean anything in a directory listing, so just pretend they don't exist. We'll discuss ".." later.

Along with all the files that I put into my MSDOS321 directory, I've made a subdirectory to place all my GWBASIC files. I've also made 3 subdirectories inside the WORDS directory; I call these subdirectories WORDPROC, DICTNARY, and THSAURUS. Inside the WORDPROC directory, I've made a subdirectory called VERYEASY, in which I keep my computer copy of this manual. My hard-disk organization is pictured graphically below:

Pictorial Representation of My Hard Disk

(Each box represents a single directory)
(Please don't type any of this!)

```

" Root Directory "
†
† " MSDOS321 "
† † (all DOS files are in here) † Command.com †
† † " GWBASIC " † Autoexec.bat †
† † † † † Config.sys †
† † ^ †
† ^
† " DRAW " † " SPREDSHT " † " DATABASE " †
† † † † † † † †
† ^ ^ ^
† " WORDS "
† † † †
† † " WORDPROC " † †
† † † † † † †
† † † " VERYEASY " † † †
† † † † (this directory contains the † † † †
† † † † the computer version of this † † † †
† † † † "Very Easy DOS" manual) † † † †
† † † ^ † † †
† † ^ † †
† † " DICTNARY " † †
† † † † † †
† † ^ † †
† † " THSAURUS " † †
† † † † † †
† † ^ † †
† ^ †
" ^

```

There is nothing special about the way I've organized my files; it just makes sense to me. You should organize your files into whatever directories make sense to you. You don't have to have directories and subdirectories if you don't want to; they're just a big help in organizing your files.

HOW TO MAKE DIRECTORIES

Let's make a hard-disk subdirectory and put everything that's on your DOS floppy disk #1 into it. To do this we have to learn 2 more DOS commands: COPY and MD.

COPY

Let's use the COPY command to make sure the COMMAND.COM file is in the root directory. (If it is already there, you won't be wrecking anything by doing this lesson.) Each time your computer is turned on, it needs to find that file in the root directory. If the computer can't find it there, then you won't be able to use the computer. It is possible to keep COMMAND.COM some place other than the root directory, but it's not worth the effort to learn how to do so.

The COPY command is one of the commands contained inside the COMMAND.COM file. That sounds a bit confusing, doesn't it? You will better understand how this is possible, after you finish Section #4.

Make sure your computer is on and that you have the DOS floppy disk #1 inside the floppy-disk drive, and the door to it closed. Then press the ENTER key.

The symbol A> (which means the computer's attention is on the floppy drive) should appear last on the screen. If not, re-start DOS by pressing **CTRL-ALT-DEL**; that is, hold down the CTRL and ALT keys and then press the DEL key while still holding CTRL and ALT down. That is a short cut so you don't have to turn the computer's power off-and-on to re-start DOS. The jargon term for starting or re-starting DOS is to **boot** or **re-boot**. Seems silly, doesn't it?

When the symbol A> is last on the screen, type:

```
COPY COMMAND.COM C:\ enterkey
```

What that means in plain English is "please make a copy of the COMMAND.COM file, which is located in the directory that the computer's attention is currently on, and put that copy into the directory called C:\." This "C:"

refers to the hard disk, and "\" refers to the root directory; therefore, "C:\" refers to the root directory on the hard disk.

After the computer stops copying, type:

```
C: enterkey
```

That means "switch the computer's attention to the hard disk". When the C> shows up (which means the computer's attention is now on the hard disk), type:

```
DIR enterkey
```

Among the things you find on the screen will be the C:\ symbol (which lets you know that the computer's attention is now on the C:\ directory) and the COMMAND.COM file that you just copied off the floppy disk. Congratulations!

MD

Before **M**aking a **D**irectory, choose a directory name. The name you choose can be 1 to 8 characters long. The characters can be any combination of numbers, letters, or symbols (no spaces) (I don't like these dumb rules either). The symbols can be any of those shown on number keys 1 thru 7 at the top of your keyboard, plus the dash and apostrophe and some others that aren't worth remembering.

If you write many letters, you might want to keep all of them in a directory called LETTERS. If many of them deal with money matters, you may want to create a subdirectory called \$LETTERS. If you write a lot to your mother, a directory called LETTERS.MOM could be created.

That last name demonstrates one final point about directory names: if you want, you can add up to three more letters, numbers, or symbols (no spaces), as long as you separate these 3 from the first 1-8 by a period. This second part, the part after the period, is sometimes referred to as an "**extension**". If a name does not have an extension, then you do not need to put a period after the first part!! Earlier in this Section, in the discussion of "executable files", you read about "com, exe, and bat" filename extensions. Other examples of valid directory names are:

```
DOS
FEDTAXES.'89
IBM.XT
```

Everything you just read about directory names is also true of **filenames**; that is, they can have up to 8 type-written characters, followed (optionally) by an "extension" of 1-3 characters that are separated from the first characters by a period. An important point is that DOS will not let you use the identical name twice inside the same parent directory (that is, such that they both would show up in the listing after a DIR command). If you try to put 2 files with identical names into the same directory, DOS erases one of them without warning!!!

To be different, directory and filenames have to differ by only one character before, or after, the period. Thus FEDTAXES.'89 is considered different from FEDTAXES.'88, and IBM.XT is different from IBMPC.XT. You can have 2 or more directories and/or files with the same name somewhere on your disk; they just can't be in the same parent directory. That is, DOS won't allow you to put them where they'd show up after you give a DIR command. If you do try to make 2 subdirectories (or even a file an subdirectory) with the same name in the same parent directory, DOS types out some kind of error message onscreen instead. After all, don't human parents want their children to have different names? (My apologies to the parents of Larry, Daryl, and Daryl, on the Bob Newhart TV show.)

Many an application program automatically puts its own unique extension (*i.e.*, the second part of a name, the part after the period) onto names of files that you create using that program. This is sometimes necessary so that such a program can later automatically identify which files it created, as opposed to files created by some other program.

Let's now make a directory called VEDOS (an abbreviation for Very Easy DOS) by typing:

```
MD VEDOS enterkey
```

That translates as "please make a directory called VEDOS inside whatever directory the computer's attention is currently on". To see if you've succeeded, type

```
DIR enterkey
```

Among anything else that you see should be:

```
COMMAND COM
VEDOS      <DIR>
```

Notice that DOS doesn't type out the period between the first and second part of names (such as COMMAND.COM), even tho DOS requires that you yourself must include one when typing commands on the command line. (That doesn't seem fair!?)

Now for the fun part. Switch the computer's attention back to the floppy drive by typing:

```
A:  enterkey
```

That translates as "switch the computer's attention to the floppy disk drive". When the A> symbol shows up last on the screen, type:

```
COPY *.* C:\VEDOS  enterkey
```

That translates as "please make a copy of every file in the directory that the computer's attention is currently on and put those copies onto the hard disk inside the VEDOS subdirectory, which is inside the root directory". The asterisk symbol, *, before the period in any DOS command is interpreted by DOS to mean "every filename of any size from 1 to 8 characters long". The asterisk after the period is interpreted to mean "every filename extension of any size from 0 to 3 characters long".

You read up above that files can be named just like directories, that is, with up to 8 characters (no spaces) followed (optionally) by a period and 3 more characters (no spaces). Therefore, you need to type not one but two asterisks separated by a period, when you want to say "all files".

An important point is that the COPY command will only copy files. It won't copy subdirectories or the files in those subdirectories. More about this later.

The way it is taught in this manual, COPY will copy only the files that show up on the listing you get when you type out the DIR command. COPY is much more

versatile than that, but it is difficult (and not very useful) to learn all that versatility!

You can now take your DOS floppy-disk out of the computer and store it in a safe place. If your version of DOS has a second disk, put it into the floppy drive and then type:

```
COPY *.* C:\VEDOS enterkey
```

That is exactly what you typed to copy the first disk. After the computer is finished copying, take this second disk out of the computer and store it with the first one. Then type:

```
C: enterkey
```

The C> prompt will now show up last on the screen, which means that the computer's attention has switched back to the hard disk.

Those of you who have used this manual to install DOS 3.3 or earlier onto your computer should be very proud of yourselves! You now have put all of DOS onto your hard disk.

Section #4

AUTOEXEC.BAT

IN THIS SECTION WE'LL CONSTRUCT A SPECIAL FILE THAT WILL AUTOMATICLY GIVE DOS COMMANDS TO THE COMPUTER EACH AND EVERY TIME YOU TURN THE COMPUTER ON. WE'LL LEARN A NEW TERM: "THE CURRENT DIRECTORY", AND WE'LL LEARN THE FOLLOWING DOS COMMANDS: COPY CON, PROMPT, PATH, and TYPE.

IF YOU ARE WORKING ON SOMEONE ELSE'S COMPUTER OR ON A COMPUTER AT WORK, TYPE WHAT'S IN THE FOLLOWING BOX. DON'T WORRY ABOUT WHAT YOU'RE DOING--YOU'LL UNDERSTAND IT ABOUT 15 PAGES FROM NOW. THE PURPOSE OF WHAT'S IN THE BOX IS TO PREVENT YOU FROM WRECKING ANYTHING THAT SOMEONE ELSE HAS MADE. DON'T BE CONCERNED ABOUT ANY ERROR MESSAGES THAT ARE SHOWN ON THE SCREEN AFTER YOU TYPE THIS; IF YOU TYPE IT EXACTLY AS I SHOW YOU, NOTHING BAD WILL HAPPEN:

C:			enterkey
CD\			enterkey
RENAME	AUTOEXEC.BAT	AUTOEXEC.OFF	enterkey

NOW WE CAN START, AND WE WON'T WRECK ANYTHING.

Current Directory

I've used the phrase "the directory that the computer's attention is currently on" many times in this manual so far. From now on, I'm going to use the phrase "**the current directory**" instead. So, when I say "the current directory is GAMES", I mean that the computer's attention is currently on the directory called GAMES. Or, if I say that the root directory on the floppy disk is the current directory, I mean that the computer's attention is currently on the A:\ directory.

Before you go on, take the time to think about what I just said, or you'll soon get confused. I've chosen this phrase, "current directory", because it's the phrase many other people and authors and software manufacturers use.

Many DOS commands, as described in this manual only, work exclusively on the directory that the computer's attention is currently on. This is a very important and useful point, if you want to use DOS the easy way. Using DOS the Very Easy way, the DIR command gives you a listing only of files and subdirectories that are inside the current directory, COPY makes copies of only the files in the current directory, and MD makes subdirectories for the current directory only. The problem is, there may be times when you don't remember what the "current directory" is. You can eliminate that problem by creating an AUTOEXEC.BAT file.

When you turn on the computer, or re-start DOS with the CTRL-ALT-DEL keys, DOS does many things, including looking in the root directory for a file called AUTOEXEC.BAT. If it does not find that file, it asks you for the date and time, and then stops to await your commands. If DOS does find an AUTOEXEC.BAT, it doesn't ask for the date and time. Instead, it AUTOmatically EXECutes the commands inside that file.

That explains the first part of the name, "autoexec". The second part of the name, "bat", tells DOS that this is an "executable file", a file that "does" something. You read about such files at the start of Section #3. You are about to make your own AUTOEXEC.BAT file, and you'll put a command into it that causes the "current directory" to be displayed instead of the C> or A> prompts.

COPY CON

We'll make a file by telling the computer to COPY whatever we type into the CONsole (a console is another word for a keyboard) and to place that copy into our new file. Type what's in the box below (I'll soon explain what it means). (Remember, "F6key" means to press the F6 key, not to type out the word.)

C:	enterkey
CD\ COPY CON AUTOEXEC.BAT	enterkey
PROMPT THE CURRENT DIRECTORY IS \$P \$_	enterkey
F6key	enterkey

Now, re-start DOS with CTRL-ALT-DEL. When DOS is finished doing its thing, instead of the usual C> prompt showing up last, you'll see the following:

```
THE CURRENT DIRECTORY IS C:\
```

-

(The small line under the T in THE above represents the cursor.)

The first 2 lines in the box above are to intended to make absolutely sure you are "in" the hard disk root directory. The line starting with "COPY CON" translates as "please make a file called AUTOEXEC.BAT by COPYing whatever I type next into the CONsole, and place that file into the current directory".

The line starting with "PROMPT" tells DOS that the PROMPT (which is normally C> or A>) should from now on be the phrase "the current directory is" followed by the **PATH** (\$P) that the computer has taken to get to the current directory. By the word PATH is meant the sequence of directories (think of them as file boxes) that the computer had to open, starting with the root directory, to get inside the current directory. PATH is a difficult concept, but you will soon understand it.

The last part of the PROMPT line is \$_ (a dollar sign followed by an underline). That tells the cursor to move down to the start of the line just below the prompt, to await your orders, instead of its stopping directly after the prompt. I have done this to make the "command line" (the line you type commands on) look less cluttered.

The F6 key on the last line is a short-cut for typing a ^Z (Control-Z). ^Z tells DOS that you've reached the end of what you want to type into the new file. After you then press the ENTER key, DOS takes over and saves your new file.

PATH

(Look back on page 20 in this manual. There you'll find the "Pictorial Representation of My Hard Disk". Keep a book mark there for easy reference during the following discussion.)

On my computer, the PATH to the DATABASE directory is C:\DATABASE, because to get there the computer first had to open the hard disk root directory, C:\, and then the DATABASE directory.

The PATH to the GWBASIC directory is C:\MSDOS321\GWBASIC, because the sequence of how the boxes (directories) have to be opened is:

- 1) C:\ (the hard disk root directory)
- 2) MSDOS321 directory
- 3) GWBASIC directory.

Right now, I am working on this manual. I am currently in the VERYEASY directory. The PATH the computer took to get me there is C:\WORDS\WORDPROC\VERYEASY. I have an AUTOEXEC.BAT file in my root directory that has a line in it about prompts, as yours now does; if I were to let DOS have control of the computer right now (instead of my wordprocessor being in control), the prompt would read:

```
THE CURRENT DIRECTORY IS C:\WORDS\WORDPROC\VERYEASY
-
```

DOS types out the PATH using backslashes (\) to separate the different directories. This is important, because later I'll show you how to move from one directory to another, and you'll need to type the PATH to the desired directory the same way that DOS does. Please note that the first backslash in the PATH symbolizes the root directory, but all the others are just separating marks to help DOS know where names begin and end. I wish the designers of DOS hadn't used the same symbol to mean two different things in the same sentence!!!

The Importance of PATH

As taught in this manual only, commands typed onto the DOS command line cannot be executed unless the file containing the detailed information on how to carry out that command is inside the current directory. If the executable file (DOS's, your's, or an application program's) whose command information you want to use is not inside the current directory when you give the command (*i.e.*, type the command onto the DOS command line and then press ENTER), that

command won't be executed. Instead, an onscreen message tells you that the command is bad or the file missing. That message may just mean DOS can't find the right file.

There are major exceptions to what I said in the previous paragraph, but the exceptions would require lengthy explanations that aren't worth your time right now. Months from now, ask a friend to explain the full versatility of the command line as well as "RAM resident/TSR" application programs.

DOS always looks inside the current directory for executable files to carry out your commands. It looks additional places only if you tell it where to look. The easiest way to do so is to add a PATH statement to your AUTOEXEC.BAT file. A PATH statement tells DOS additional places to look for executable files that it can't find in the current directory.

Let's add a PATH statement to your AUTOEXEC.BAT by re-typing the entire file (this will erase the original version, but that's OK). The box below assumes that your DOS files are in a subdirectory called VEDOS. If they are not there, and you know the PATH to your DOS files, you should substitute it for \VEDOS on the PATH line below. If you are confused, stop here and ask a computer-literate friend to ferret out the PATH to your DOS files. For those of you ready to start, type:

COPY CON	AUTOEXEC.BAT	enterkey
PROMPT	THE CURRENT DIRECTORY IS \$P \$_	enterkey
PATH	C:\;C:\VEDOS	enterkey
F6		key

TYPE

To review what you've put into your file, type the following:

TYPE AUTOEXEC.BAT enterkey

You just used a new command, TYPE, which says "please show me what's inside whatever file I type right after the word TYPE". What you see onscreen in this instance is:

```
PROMPT THE CURRENT DIRECTORY IS $P $_  
PATH C:\;C:\VEDOS
```

You can't look inside a directory with this TYPE command; to do that, use the DIR command. If you use the TYPE command to look inside an executable file, or even some data files, you'll see what looks like gibberish, accompanied by a lot of beeping sounds. Most people refer to this gibberish as "happy faces". Try it out on COMMAND.COM someday.

Notice that there is a semi-colon (;) after the first backslash on your AUTOEXEC.BAT's PATH line, not a colon (:). In contrast, there are supposed to be colons after each hard disk symbol (C). If you find that you didn't type the PROMPT or PATH statement correctly, just start all over again with COPY CON etc. When the file is completed (and verified using the TYPE command), press CTRL-ALT-DEL to re-start DOS.

Now your AUTOEXEC.BAT file is complete. The "current directory" is always displayed as the prompt whenever the DOS program is in charge of the screen (and that's anytime an application program, such as a wordprocessor or a game, is not in control of the screen). And the computer knows where to look for executable files; it will look for them first in whatever is the current directory, and then in C:\, and then in C:\VEDOS.

Software you buy may tell you to add its directory to the PATH. If so, just re-type the entire AUTOEXEC.BAT file, add a semicolon (not a colon) to the end of the existing PATH line, and then type (no spaces) the complete PATH to the new subdirectory into which you put the new software's files. If you buy another such program, just add another semicolon and add on again. For example, if you buy a DOS utility program and put its files into a subdirectory called UTILITY inside your ROOT directory, you should re-type your present AUTOEXEC.BAT file so that the PATH line reads like this:

```
PATH C:\;C:\VEDOS;C:\UTILITY
```

Don't think you have to put the name of every directory on your hard disk into the PATH statement; the only ones that need to be mentioned are the ones that contain executable files that you would like to use even when you're in a different directory.

Some software programs themselves automatically modify the PATH statement when you first install the program onto the hard disk. They add the location of their executable files onto the front or end of the existing PATH line in your AUTOEXEC.BAT file. If you don't have a PATH line, they may create one!

Sometimes, instead of adding a PATH to the PATH statement, such "automatic" installations will add a new type of line to your AUTOEXEC.BAT. This new line will start with the word "SET" and end with the PATH to the new software's files. This is an alternate way of telling DOS where to look for executable files; it is not important that you learn how to do it yourself.

It is OK to have a PATH statement in your AUTOEXEC.BAT file that is longer than can fit across one line on your computer screen. Just keep typing and adding semicolons and paths, and let DOS worry about the fact that the PATH stretches over 2 lines. Don't worry that part of a word seems to be on two different lines; DOS is smart enough to know what you mean. The PATH statement can be a maximum length of 120 characters; if you try to type one longer, the computer beeps at you.

Fun Time

For the fun of it, you can change your prompt temporarily at any time, without having to write a new file. Just type PROMPT on the command line, followed by a space and then anything up to 120 characters long (or until your computer beeps at you and refuses to accept any more characters). For example, type:

```
PROMPT MASTER, WHAT IS THY WISH? enterkey
```

As the result of that command, the prompt becomes:

```
MASTER, WHAT IS THY WISH?
```

You can convert to the C> symbol by typing the following:

```
PROMPT enterkey
```

When you feel ready to continue, use CTRL-ALT-DEL to re-start the computer. That will get DOS to again execute your AUTOEXEC.BAT file. The result will be that the PATH is again displayed as the prompt. No matter what directory you

are in from now on, the prompt will always display the entire PATH to that directory. You'll need it to be that way to do the remaining practice sessions in this manual.

Internal Commands

The list below comprises some of the important commands contained inside the COMMAND.COM file. When any one of these commands (or any of the others that are "internal" to COMMAND.COM) are entered onto the DOS command line, they are immediately executed by DOS. This occurs whether or not your PATH statement mentions C:\ (the directory that contains COMMAND.COM).

CD	COPY	COPY CON	DATE
DELETE	DIR	ERASE	MD
PATH	PROMPT	RD	RENAME
SET	TIME	TYPE	

You could say that COMMAND.COM's internal commands have been loaded into RAM and are "resident" there, waiting for you to give them orders from any directory whatsoever. Some application programs, after they're first turned on, work in a similar fashion; such programs are described as "RAM resident" (or, obscurely, as "TSR" programs).

P.S.

If you are doing these exercises on someone else's computer or on a computer at work, let's now undo what you did in this Section, and put the computer back the way you found it. (If you are on your own computer, do not do this step, because the computer is set up the way you'll want it from now on.)

The explanation of what I'm asking you to type will have to wait till later. Don't be concerned about any error messages that show up onscreen as a result; if you type exactly what's in the box below, nothing bad will happen. Type:

ERASE	C:\AUTOEXEC.BAT	enterkey
RENAME	AUTOEXEC.OFF AUTOEXEC.BAT	enterkey

Then press CTRL-ALT-DEL to restart DOS. Now the computer is back the way it was before we started this Section.

Section #5

Learning to Move Around

IN THIS SECTION YOU WILL LEARN THE DOS COMMAND "CD", WHICH ALLOWS YOU TO MOVE FROM ANY DIRECTORY TO ANY DIRECTORY.

BEFORE YOU START THIS SECTION, THE PROMPT MUST BE INDICATING A CURRENT DIRECTORY ON THE HARD DISK. IF THE PROMPT IS JUST C>, READ THE PREVIOUS SECTION ON "AUTOEXEC.BAT" TO LEARN HOW TO HAVE THE CURRENT DIRECTORY DISPLAYED AS THE PROMPT. IF YOU DON'T WANT TO DO THAT NOW, OR ARE USING SOMEONE ELSE'S COMPUTER, JUST TYPE THE FOLLOWING IN ON THE COMMAND LINE. THIS WILL MAKE SURE YOU'RE ON THE HARD DISK AND TEMPORARILY CHANGE THE PROMPT TO INDICATE THE CURRENT DIRECTORY:

C:						enterkey
PROMPT	THE CURRENT DIRECTORY IS	\$P	\$_			enterkey

CD

You're going to want to be able to move from one directory to another, that is, to change what the current directory is. If you have all your files in the root directory, and don't have any other directories, then you don't need to learn this part of the manual. However, if you have even one other directory containing files, then you need to know how to get inside that directory.

Sometimes I like to think of the letters CD as being short for "Come Down" since it tells the computer what PATH to come down to reach another directory. Other times I think of CD as being short for "Current Directory" since I'm asking that whatever I type next will become the current directory. Most people think CD stands for "Change Directory" since they are asking for the current directory to be changed. You think of it any way that helps you remember how CD works.

To practice with CD, let's first create some dummy directories. Type what's in the box on the following page exactly as I've written it. Don't worry about anything that happens on the screen while you're typing, and I'll explain what you did right afterwards:

CD\ MD LARGE.BOX CD LARGE.BOX MD MEDIUM.BOX CD MEDIUM.BOX MD SMALL.BOX CD\ MD RED.BOX CD RED.BOX MD WHITE.BOX CD WHITE.BOX MD BLUE.BOX	enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey
---	--

The explanation of what you just typed is:

CD\
This makes sure you are in the root directory. CD\
means "make the root directory the current directory". The backslash, \, by itself is a short cut symbol for the root directory of whatever disk the computer is currently paying attention to. If the "current directory" is a directory on the hard disk, then the backslash by itself represents the hard disk root directory, namely C:\. Similarly, if the current directory is a directory on the floppy disk, then the backslash by itself represents the floppy-disk root directory; that is, it's a short-cut way of typing A:\.

MD LARGE.BOX
The MD command (which you learned about in Section #3) is used here to create a new subdirectory called LARGE.BOX inside the current directory (which is the root directory).

CD LARGE.BOX
The CD command is used to make that new subdirectory the current directory.

MD MEDIUM.BOX
This makes a subdirectory called MEDIUM.BOX inside the current directory, which is LARGE.BOX.

CD MEDIUM.BOX

This makes MEDIUM.BOX the current directory.

MD SMALL.BOX

This makes a subdirectory called SMALL.BOX inside the current directory, which is MEDIUM.BOX.

The remaining lines in the file should be obvious to you, based on the description of the first 6 lines. Right now, try to describe each remaining line to yourself the way I just described the first 6 lines. Afterwards, take a look at the diagram below. It represents the changes you've made to your hard disk organization by doing the exercises in this manual, so far. Keep a bookmark at this page, and refer to this diagram often during the rest of this Section and the following Section. Look at it right now and then go back over the previous page; you'll discover that the page will make more sense to you than it did the first time you read it.

- #1 To use Rule #1, all you have to do is type the two letters CD, followed by a space, and then the name of the desired directory. However, when using this rule, you can change only to subdirectories of whatever is the current directory. In other words, you can change only to a directory that shows up on the list you would get if you were to give the DIR command just before you give the CD command. (You don't need to give the DIR command first; I said "if".) This is the rule you were using when you were making directories and changing to them, 2 pages ago.
- #2 If you want to go to a directory that is not a subdirectory of the current directory, you have to type CD, (followed by a space if you'd like, but it's not necessary), followed by the entire PATH to the directory you want to go to. (If you have forgotten what a PATH is, you should re-read the discussion of PATH in Section #4.) Type the PATH without the disk designation; that is, don't type A: or C: at the start of the PATH.
- #3 If you want to go to the parent directory of the directory you're currently in, you can use Rule #2 or you can instead type just *CD..* (that is, CD followed by 2 periods). There is no logic to this command, other than that one of the designers of DOS thought up the idea to use 2 periods to symbolize the parent directory of whatever is the current directory.

You cannot change to a directory on another disk drive using the CD command. If you are on the hard disk, and you want to go to a directory on the floppy disk in the A: drive, you'd first have to type *A: enterkey*. Afterwards you could use the CD command to move around. Similarly, if you are in a directory on the floppy disk, and you want a directory on the hard disk to become the current directory, you first must type *C: enterkey*, and then use CD to get to it.

The last thing we did 3 pages ago was to create the BLUE.BOX subdirectory inside WHITE.BOX; the current directory is still WHITE.BOX. If you want to make BLUE.BOX the current directory, so you could use Rule #1 by typing:

CD BLUE.BOX enterkey

Or you could use Rule #2, even tho it's not necessary, by typing the entire PATH:

CD\RED.BOX\WHITE.BOX\BLUE.BOX enterkey

Use one of those 2 methods, now, to make BLUE.BOX the current directory; type the line the same way I did. The prompt should tell when you've succeeded; it will read:

THE CURRENT DIRECTORY IS C:\RED.BOX\WHITE.BOX\BLUE.BOX

Let's make the root directory the current directory. Type:

```
CD\ enterkey
```

What you just did was to use Rule #2; you told the computer "I want the current directory to be the last directory in the PATH I type right after the letters CD". Well, since all you typed was the backslash, and because you are on the hard disk and because the first backslash in a PATH always refers to the root directory, the current directory became C:\ (which stands for the hard disk root directory).

Now let's make the current directory SMALL.BOX. To do that, we could use Rule #1 repeatedly by typing:

```
CD LARGE.BOX    enterkey
CD MEDIUM.BOX  enterkey
CD SMALL.BOX    enterkey
```

Or we could use Rule #2 just once by typing:

```
CD\LARGE.BOX\MEDIUM.BOX\SMALL.BOX enterkey
```

You can use whichever you like. Pick one method, and type it out now. The result will be that your prompt will look like:

```
THE CURRENT DIRECTORY IS C:\LARGE.BOX\MEDIUM.BOX\SMALL.BOX
```

Take another look at the "Pictorial Representation of Your Hard Disk" on page 36. You've used CD to move from the innermost directory inside RED.BOX to the innermost one inside LARGE.BOX. You did that by first moving up to the root directory and then down to SMALL.BOX.

How would you go back to BLUE.BOX without stopping at the root directory? The easiest way to do that would be to type the following (do so now):

```
CD\RED.BOX\WHITE.BOX\BLUE.BOX enterkey
```

The prompt will now look like this:

```
THE CURRENT DIRECTORY IS C:\RED.BOX\WHITE.BOX\BLUE.BOX
```

You have just demonstrated that you don't have to stop anywhere along the way to your next current directory; just type in the entire PATH, press ENTER, and you're there already! I had you do more than that at first, up above, to show you the flexibility of CD.

Now that you're in BLUE.BOX, you could go to WHITE.BOX by using Rule #2 and typing out the entire PATH to it, like so:

```
CD\RED.BOX\WHITE.BOX  enterkey
```

Or you could use Rule #3 and type just

```
CD..  enterkey
```

In the future, use whichever you like; but right now, try out "CD.. enterkey" three times in a row. You'll end up back at the root directory.

It doesn't matter which Rule # you use to get to your new current directory. Use whichever Rule # you feel most comfortable with. If you don't yet feel comfortable with CD, try re-doing this Section. CD is easy, once you understand it; and it is essential that you understand it well.

If you are concerned about all the directories that we just littered the hard disk with, don't worry. In the next Section, you learn how to get rid of them.

Section #6

Changing Your Mind

IN THIS SECTION YOU WILL LEARN THE FOLLOWING DOS COMMANDS: RD, ERASE, and RENAME.

There will come a day when you wish you'd organized or named things differently. This section describes how you can make changes.

First, let's put some files into those dummy directories that you created with the MD command several pages back. We'll practice with the COPY command while we do it. Move to the directory that contains your DOS files (that is, make that the current directory). Many of you can do that by typing the following: *CD\VEDOS enterkey*. If your DOS files are some place else, type the letters CD followed by the PATH to the directory containing them. You must have your DOS files in the current directory to do the rest of this Section.

After you have switched to the directory containing your DOS files, type:

```
COPY F*.* C:\LARGE.BOX\MEDIUM.BOX enterkey
```

What you just did was to ask for all the files that start with the letter F in the current directory to be copied to the MEDIUM.BOX directory.

Notice how we used asterisks to substitute for saying "every name of any length" just as we did in Section #3, when you first were introduced to the COPY command. Also notice that the entire PATH to the destination directory has to be specified. If you don't correctly specify the PATH, you'll get a nasty message onscreen which says that DOS can't find the destination directory, or that it won't copy a file onto itself. Now copy the COM files to RED.BOX by typing

```
COPY *.COM C:\RED.BOX enterkey
```

RD

Now let's get rid of SMALL.BOX directory. There are a few ways this could be done; the way I'm going to teach you is the safest. By safest I mean that you have the least chance of getting rid of the wrong thing. Type:

```
RD C:\LARGE.BOX\MEDIUM.BOX\SMALL.BOX enterkey
```

That translates as "Remove the Directory at the end of the PATH that follows the letters RD". Specify the entire PATH, starting with the name of the disk drive (in this case we're on the hard disk, but you could just as well be removing a directory on a floppy disk, in which case you would have wanted to start the PATH with A:\). Only the very last directory in the PATH, as you've written it, is removed; the other directories you mention are untouched.

Let's try the same thing with the MEDIUM.BOX directory. Type:

```
RD C:\LARGE.BOX\MEDIUM.BOX enterkey
```

You now get a message that translates as "DOS can't get rid of this directory, because you misspelled the filename, because you're asking to get rid of the current directory, or because that directory still contains files and/or subdirectories". Well, we did indeed put files into MEDIUM.BOX. DOS won't let you get rid of a directory that still contains files and/or subdirectories (and it won't let you get rid of the current directory while you're still "in" it).

ERASE

There weren't any files in SMALL.BOX, so getting rid of it was easy. However, to get rid of MEDIUM.BOX we must first empty it of all its files. There are a few ways this could be done; the way I'm going to teach you is the safest. By safest I mean that you have the least chance of getting rid of the wrong thing. Type the following:

```
ERASE C:\LARGE.BOX\MEDIUM.BOX\*.* enterkey
```

A message will then come onscreen asking if you are sure you know what you're doing. Type the letter Y and then press the ENTER key, to reply "yes".

What you just typed translates as "erase all the files in the last directory specified in the PATH listed right after the word ERASE". (As you've seen before, *.* means "all files".) Specify the complete PATH, starting with the name of the disk drive (right now, you're on the hard disk, but you could just as well be erasing files in a directory on a floppy disk, in which case you would have wanted to start the PATH with A:\).

Instead of using the word ERASE, some people like to use the word DELETE (or DEL). They all mean the same thing to DOS.

Now that you have erased all the files in MEDIUM.BOX, and because you previously removed its subdirectory, and because MEDIUM.BOX is not the current directory, you are capable of getting rid of MEDIUM.BOX. Type:

```
RD C:\LARGE.BOX\MEDIUM.BOX enterkey
```

Now let's continue our discussion of ERASE.

If all you want to do is erase a single file, substitute its name for *.* on the command line. For example, if you were to type:

```
ERASE C:\RED.BOX\FORMAT.COM enterkey
```

you would erase the FORMAT.COM file in the RED.BOX directory (assuming you have that file there).

ERASE is a dangerous command. If you are sloppy in what you type, you end up erasing a great deal of work. For example, if you were to type (but **DO NOT TYPE THIS !!!**) ERASE *.* , you would be asking to erase everything in the current directory. That may be what you want to do; or maybe you're tired and confused and have forgotten what directory you're in, and so you just asked to erase all your company's tax files, or all of your daughter's term paper. AAAARRGH!!

I strongly urge you to follow my "complete PATH" way of removing files. Even tho there are shortcuts, the time saved over the next 20 months won't amount to 20 minutes. But if you make a mistake with a short cut, it may take you 20 hours

to replace all the information that you accidentally erased. It's your choice. I'm recommending that you be careful and patient. Use the long way; in the long run, you'll be glad you did.

Let's now get rid of the remaining practice directories. Type:

RD	C:\LARGE.BOX	enterkey
RD	C:\RED.BOX\WHITE.BOX\BLUE.BOX	enterkey
RD	C:\RED.BOX\WHITE.BOX	enterkey
ERASE	C:\RED.BOX*.*	enterkey
RD	C:\RED.BOX	enterkey

RENAME

You may want to change the name of some of the files that you create yourself. To do so is easy with the RENAME command. (But don't change any program-file names in any software you buy, or you might make the program inoperable!!)

Let's change the name of the AUTOEXEC.BAT file:

CD\	enterkey		
RENAME	AUTOEXEC.BAT	VAMPIRE.BAT	enterkey

Use the DIR command to see what you've done.

The RENAME command translates as "in the current directory, please get rid of the first filename I just typed and replace it with the second filename". A very important point is that the contents of the file don't change, only the name.

If you do try to use a name that is already in use for another file or subdirectory in the current directory, DOS stops and gives you an error message about duplicate names.

You can't use RENAME to change the names of directories. DOS doesn't have a command to alter directory names. If you want to rename a directory, you could make a new directory with the desired name, copy all the files from the original directory to the new one, and then eliminate the old directory.

Let's end by resurrecting your AUTOEXEC.BAT by typing:

```
RENAM  VAMPIRE.BAT  AUTOEXEC.BAT  enterkey
```


Section #7

Batch Files

IN THIS SECTION, YOU WILL LEARN HOW TO WRITE SPECIAL FILES THAT WILL SAVE YOU TIME AND REDUCE FRUSTRATION. THIS SECTION IS NOT ESSENTIAL FOR YOU TO LEARN, BUT BATCH FILES CAN BE A BIG HELP.

Batch files (also known as Bat files--and I'm not trying to be a Joker) are executable files that you yourself make. They can carry out several commands with only one command by you. The only rule is that the names of such files have to end in .BAT. You've made one Bat file already, the AUTOEXEC.BAT file.

The way Bat files work is that you type just the first part of the file name (the part before the period), press ENTER, and then DOS executes all the commands in that file, one right after the other. They are carried out as a "batch".

Let's create a file to see how this works. Type everything that's in the next box; most of it should be familiar to you. You've used CD, COPY, MD, DIR, and ERASE before. If you feel uncomfortable at this point with these commands, look them up in the INDEX at the back of this manual, and read the appropriate pages.

The only new commands in the following box are TIME and DATE, which allow you to find out the time and date that the computer thinks it is, and to modify them if they're incorrect. If they are incorrect, enter the correct data, or press the ENTER key to say you don't care. When you're ready, type:

CD\ COPY CON EASY.BAT MD EASY COPY *.* C:\EASY CD EASY DIR ERASE C:\EASY*.* CD\ RD C:\EASY DATE TIME ERASE C:\EASY.BAT F6key	enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey enterkey
---	--

Now type the word EASY and then press the ENTER key.

Impressive, isn't it? You typed a single word, and 10 commands were carried out. If they weren't, all it means is that you typed something incorrectly. However, you can't use the TYPE command to look for mistakes, because the last line in the file erased the file itself!!

Delayed Reactions

Take some time now to notice the effect on Bat files of commands that take input from you before they are complete, commands such as ERASE *.* , DATE, and TIME. A Bat file carries out commands one line at a time; the next line is not carried out until the previous line is fully carried out and control has been returned to DOS. This is very useful, as I will explain in the next set of paragraphs.

Suppose you have a wordprocessor program plus a completely separate thesaurus program. Separate thesaurus programs are usually started on the DOS command line by commands that have some additional part at the end, for example "/i". The /i (or whatever) lets you "install" the program so that it is ready to use ("resident" in RAM, as described at the very end of Section #4), but DOS remains in full control of the screen. The thesaurus can be accessed later by pressing the appropriate "hot key" (which is different for each program). When you no longer need the thesaurus, there may be a way to remove it from RAM, such as re-typing the startup command, substituting /U for /i (the /U means to "un-install").

I bet that was confusing to many of you. If so, read the next few paragraphs, and then re-read the one above.

Let's say that your hard disk is set up like mine (see the diagram on page 20) and that you always like to turn on the thesaurus whenever you turn on the wordprocessor, and that you like to turn the thesaurus off when you've finished using the wordprocessor. Let's pretend the startup commands are SYNONYM for the thesaurus and FULLPAGE for the wordprocessor.

A single Bat file could start both programs for you and later turn off the thesaurus as soon as you turn off the wordprocessor. You could call the file TYPING.BAT, and you might create it like the following (I am numbering the lines for you, only to make explanations easier; if you type such a file, you would not type in the line numbers):

Line#		
1	CD\	enterkey
2	COPY CON TYPING.BAT	enterkey
3	CD\WORDS\THSAURUS	enterkey
4	SYNONYM/i	enterkey
5	CD\WORDS\WORDPROC	enterkey
6	FULLPAGE	enterkey
7	CD\WORDS\THSAURUS	enterkey
8	SYNONYM/U	enterkey
9	CD\	enterkey
10	F6key	enterkey

Then, when you want to type a letter or report, all you do is type the one word TYPING on the DOS command line, press the ENTER key, and the following sequence of events takes place:

1. Lines 3 & 4 cause the current directory to be switched to THSAURUS, and the thesaurus to be "installed", but DOS remains in control.
2. Lines 5 & 6 cause the current directory to be switched to WORDPROC, and the wordprocessor to be turned on. DOS gives control of the screen to your wordprocessor, and the Bat file stops temporarily.

3. As soon as you turn off (exit) your wordprocessor program, even if it's hours later, the Bat file immediately starts up where it left off, namely:
4. Lines 7 & 8 cause the current directory to be switched back to THSAURUS and the thesaurus to be "un-installed". Line 9 switches you back to the root directory. Because there are no more command lines in the Bat File, it gives full control back to you; an empty command line awaits your orders.

Isn't that fantastic?! Bat files are wonderful tools. Please re-read this Section. I'm sure that you will be much clearer on the concepts after the second reading. If you are just totally confused and frustrated after doing that, just skip this entire Section. There is no reason at all that you need to be an expert at Bat files; they are just a convenient way to give multiple commands. If you'd rather give your commands one at a time, that is just fine.

If you do choose to make lots of Bat files, I suggest you create a subdirectory in the root directory, and call it BATFILES. Copy all your Bat files to it, and put all future ones into that subdirectory. Do not move AUTOEXEC.BAT, because DOS looks for that one only in the root directory.

However, think back to the discussion of PATH in Section #4. There, you were told that a command you type on the command line won't do anything unless DOS can find the executable file containing that command. DOS on its own looks for such files only in the current directory. Bat files are executable files; to get such a file to "do" something when it's not in the current directory, you must tell the DOS where to find it.

The simplest way to do that is by adding a directory PATH to the PATH statement in your AUTOEXEC.BAT file (you learned about that in Section #4). If you put all your Bat files into a BATFILES subdirectory in the root directory, make sure you re-type your AUTOEXEC.BAT file and add the following to the end of what you already have on your PATH line (notice that the first symbol is a semicolon, not a colon):

```
;C:\BATFILES
```

That tells DOS to also look for commands inside the BATFILES directory, just in case it can't find the command inside the current directory or any of the other directories mentioned in your AUTOEXEC.BAT's "PATH" line.

A Personal Comment:

There are many other commands that were specially created by Microsoft (the creator of DOS) for Bat files, but knowing them will not make your application programs work any better. I've seen thick books for sale, and even one video, that claim to fully explain batch files. Bat files should be fun, not work. Resist the temptation to become an expert. Instead, spend your time becoming an expert with your wordprocessor, database, spreadsheet, or favorite game; that will be much more rewarding.

Section #8 Floppy Disks

IN THIS SECTION, YOU'LL LEARN DOS COMMANDS THAT HELP YOU USE FLOPPY DISKS. THE COMMANDS ARE: FORMAT, LABEL, COPY, and DISKCOPY.

Even tho you do have a hard disk, you're going to want to use floppy disks for storage of certain files. Such files might include duplicate copies of files that you've spent a long time creating and that you would hate to lose if the hard disk ever breaks. You might want to keep certain financial records only on floppies, away from other users of your computer. You might want to create something at home and then take it on floppies to the computer at work or school.

FLOPPY DISKS

You should be familiar with the type of floppy disk that works on your computer, at least to the extent that you could buy some at a store. As of 1991, there have been at least eight, different, common types of floppies (I find that amazing). They are listed below in approximately the chronological order in which they were introduced to PC's:

- 5- $\frac{1}{4}$ inch drive, single sided, 160K capacity
- 5- $\frac{1}{4}$ inch drive, single sided, 180K capacity
- 5- $\frac{1}{4}$ inch drive, double sided, 320K capacity
- 5- $\frac{1}{4}$ inch drive, double sided, 360K capacity
- 5- $\frac{1}{4}$ inch drive, double sided, 1200K capacity
- 3- $\frac{1}{2}$ inch drive, double sided, 720K capacity
- 3- $\frac{1}{2}$ inch drive, double sided, 1440K capacity
- 3- $\frac{1}{2}$ inch drive, double sided, 2880K capacity

Some of the currently-used terms are confusing: "high density" refers to disks with larger storage capacity, whereas "double density" refers to lower capacity disks.

If you are having trouble using floppies and don't know what to do, just bring

your computer's owner's manual and your DOS manual to a computer store and yell "help". The problem is that you not only have to have the right type of disk, but the right disk drive and the right version of DOS. If any of the three is incompatible with the other two, you get nasty messages from DOS when you try to use floppies.

Problems multiply fast when you try to use the same floppy disk on 2 different computers. The first thing to try in situations where a 1st computer's disks won't work on a second computer is to try the reverse; that is, try disks from the 2nd computer on the 1st one. Of course, you can never use a 3½ disk in a 5¼ drive, or vice versa, because they won't fit. Newer drives can read all older disk types, but the converse is not true. Swapping disks in this manner will help you determine if you have tried to use a newer type of disk in an older type of drive.

Another way to look at this disk-sharing problem is that the "format" DOS has set up on one disk is unreadable by the other computer's version of DOS. Newer versions of DOS can read all older format types, but the converse is not true. As mentioned in the preceding paragraph, you may solve your problem by using the older computer/drive/DOS as the source of the disks you want to use on both machines. If things still don't work for you, it may be that one of the drives needs adjustment. One article I read said the problem is usually head alignment on the newer drive, but another article said it's usually a hardware-design bug that you won't be able to do anything about. Bah humbug!

It is possible for a more advanced computer+drive+DOS to format older varieties of floppy disks, but subsequent use of such disks in an older-variety drive may prove unreliable. You can also format newer varieties of floppies so that they will perform as if they are older types. Read the FORMAT section of your own DOS manual to learn how to do it; I won't be discussing it here.

FORMAT

Before you can use new, blank floppies, DOS requires that you FORMAT them, similar to what you did for the hard disk back in Section #2. There, you read that FORMATING is the dividing of the disk space into small areas that DOS can use for storage and retrieval of files. The formatting of floppies accomplishes the same task.

During this step, DOS allows you to name the floppy disk. The name appears later, whenever you issue a DIR command for any directory on the floppy. DOS calls your name a "volume". For example, one of my floppies is named "Backup#1"; after I give the DIR command to that disk, the directory listing starts with a line that says "Volume in drive A is Backup#1". Having a volume name for a floppy can be useful if the label or tape you normally use to identify the floppy ever falls off.

If you have the correct kind of floppies, you can successfully format them. If you have the wrong floppies, when you try to format as described below, DOS gives you an error message indicating that either

- 1) the disk is bad or damaged
- 2) it's incorrect for the floppy drive you have
- 3) it's incorrect for the version of DOS you have.

The command to FORMAT a floppy disk is a dangerous one; if you aren't careful, you end up formatting your hard disk. You may recall from Section #2 that the FORMAT command erases everything on a disk !!!

The author of one of the large books I bought on DOS for "power users" wrote how he destroyed several of his chapters. Late one weary night, he was careless about how he formatted some floppies. (He was out of blank ones and he wanted to make a back-up copy of the chapters he had just typed onto the hard disk.) Because he was tired and careless, he erased (formatted) his hard disk, and therefore his chapters!

I'm about to show you how to avoid ever having such a story of your own to tell. You'll be making a Bat file to format your floppies. I'll have you call the file FLOPPY.BAT, but you could call it anything that ends in .BAT. However, the name you pick should not duplicate a DOS command name or the first part of the name of any existing executable file. Type:

CD\ COPY CON FLOPPY.BAT FORMAT A: /V F6key	enterkey enterkey enterkey enterkey
---	--

What you have just done is to create a fool-proof way to issue the FORMAT command so that you will never type FORMAT C: or just FORMAT (which would also format the hard disk). Now, when you want to format a floppy disk, all

you have to do is to type "FLOPPY enterkey" and away you go. The /V symbols on the FORMAT line tell DOS to allow you to give a volume name (up to 11 characters long) to this disk, a name that will show up when you give the DIR command to A:\. If you don't want to give names to your disks, leave the /V out of your FLOPPY.BAT file. Follow the onscreen instructions whenever you format a floppy; they're not difficult.

After you've formatted a floppy disk, you can make its root directory the current directory by typing "A: enterkey". You can then use MD to create subdirectories, and use CD to move around in them, just as you do on the hard disk. You can't, however, use CD to change drives; if the current directory is C:\VEDOS, you can't switch to A:\ by typing "CD A:\ enterkey". You have to type "A: enterkey" instead. Later, to return to C:\VEDOS, you must first type "C: enterkey".

LABEL

If you have at least DOS 3.0, you can change the volume name you've given to a particular floppy disk, or give a disk a name that never had one. Just put that disk into the floppy drive, and type

```
LABEL A:newname enterkey
```

You have to substitute your own new filename where I put "newname". That's all there is to it.

COPY

Copying from the hard disk to a floppy is easy. First, go to the directory that contains the file you want to copy (that is, make that the current directory). Second, type:

```
COPY filename A:\ enterkey
```

Where I put "filename", you should put the filename of any file in the current directory. If you don't want to place the new copy of the file into the floppy-disk's root directory, then right after the "A:\" add the rest of the PATH to the subdirectory that you want to copy to be placed into. If you want to copy from the floppy to the hard disk, first go to the appropriate floppy disk directory. Then, type that same COPY command, except substitute the letter C for the letter A.

You may want to copy from a floppy to a floppy, but have only one floppy drive. If so, insert the source floppy (the floppy containing the file to be copied), go to the directory containing that file, and then type:

```
COPY filename B:\ enterkey
```

DOS then copies the file to RAM, and asks you to insert a floppy into "Drive B". Drive B is the name DOS gives to a second floppy drive in computers with 2 floppy drives. But you have only one floppy drive! What you do now is to take out the source floppy and insert the target floppy (the floppy you want to the copy to be placed onto). When you then press ENTER, the computer copies the file (which it has stored in RAM) onto your target floppy. The computer now thinks you're on Drive B. Afterwards, to get it to realize it's really on Drive A, press ENTER; if that doesn't work, type "A: enterkey".

DISKCOPY

There will come a time when you'll want to make a floppy-disk copy of some other, entire floppy disk. There are at least 3 ways to do this. The simplest way is to use the DISKCOPY command. This command makes an exact reproduction of a floppy disk (the "source" disk). Have available a blank floppy (this will be the "target" disk); this disk does not have to be formatted; if it isn't, DOS will automatically format it for you. Then type:

```
DISKCOPY A: A: enterkey.
```

That translates as "please make an exact floppy-disk copy of whatever is in the floppy drive". Follow all the directions that appear onscreen as a result. It's fast and it's easy. However, be aware that DISKCOPY will destroy any old files on the target disk!!

IN CONCLUSION

You now know everything you need to know about using floppies. There is more to learn, but it is definitely not necessary. You will function quite well, and some people will consider you an expert, knowing just what you know now.

Section #9 CONFIG.SYS

IN THIS SECTION, YOU WILL WRITE A NEW FILE, CALLED "CONFIG.SYS", THAT WILL INSTRUCT DOS TO RESERVE PARTS OF THE COMPUTER'S RANDOM-ACCESS MEMORY (RAM) FOR IMPORTANT TASKS.

IF YOU ARE WORKING ON SOMEONE ELSE'S COMPUTER OR ON A COMPUTER AT WORK, TYPE WHAT'S IN THE FOLLOWING BOX. THE PURPOSE OF IT IS TO PREVENT YOU FROM WRECKING ANYTHING THAT SOMEONE ELSE HAS MADE. DON'T BE CONCERNED ABOUT ANY ERROR MESSAGES THAT ARE SHOWN ON THE SCREEN AFTER YOU TYPE THIS; IF YOU TYPE IT EXACTLY AS SHOWN BELOW, NOTHING BAD WILL HAPPEN:

C:	enterkey
CD\	enterkey
RENAME CONFIG.SYS CONFIG.OFF	enterkey

The CONFIG.SYS file is similar to the AUTOEXEC.BAT file in that it is a file you yourself make with the COPY CON command, and that it should be kept in the root directory of the hard disk. As with the AUTOEXEC.BAT file, DOS looks for this file right after CTRL-ALT-DEL is pressed or the computer is first turned on.

The purpose of AUTOEXEC.BAT is to automatically give commands instead of having to type them in on the command line each time you start the computer. The purpose of CONFIG.SYS is to give commands that are impossible to give on the command line; that is, the only way to give these commands is by having them in this file before you turn on your computer.

The commands you're about to give in this file deal with reserving parts of your available RAM for special purposes. As explained in Section #1, RAM is space on the computer's memory chips that must be available to temporarily store, among other things, all the application programs that you are actually using at the moment. CONFIG.SYS takes some of this memory out of the amount available

for application programs and gives it to DOS, so that DOS can execute special functions.

Depending on what programs you usually run, a CONFIG.SYS file may be absolutely necessary. Most large wordprocessors, spreadsheets, and databases sold today need at least some RAM allocated to DOS by these commands. If your programs are working just fine, then you could just skip this entire Section on CONFIG.SYS. That is, skip it until you buy new software that has installation instructions that tell you to add specific commands to your CONFIG.SYS file.

In the CONFIG.SYS file that you'll be making here soon, you're going to tell DOS how much RAM to allocate for simultaneous file handling (the FILES command), and how much RAM to allocate for fast temporary storage (the BUFFERS command). Type the following:

CD\ COPY CON CONFIG.SYS FILES = 20 BUFFERS = 20 F6key	enterkey enterkey enterkey enterkey enterkey
---	--

Explanations of those 2 commands (FILES and BUFFERS) would be too involved to include in an introductory manual such as this. In one DOS book I have, the explanation of the FILES command alone goes on for 5 pages. Trust me on this one; create the above file, in the hard-disk root directory, and don't think about it again unless you're asked.

In the owner's manual of some of the software you buy, it may ask you to increase the number of FILES and/or BUFFERS from what you just did. If so, rewrite CONFIG.SYS, and do whatever they ask. If different programs give different requests, use the largest number any program requests, and leave it that way for all programs. The numbers I picked here are ones that a couple different authors say are good starting points these days. Instead of having you involved, some software programs rewrite your CONFIG.SYS and AUTOEXEC.BAT files for you automatically during their installation steps.

There are several other commands for CONFIG.SYS files, but none that you really need to be concerned about. Some commands that might seem important are issued by default; such commands do not need to be mentioned in CONFIG.SYS unless you disagree with the default settings. For example, the

default "language" setting is for the USA.

DOS 4 & 5 have other important CONFIG.SYS commands, but these are added during automatic installation. Other commands may or may not be needed, depending on what extra hardware you have; the installation instructions for such hardware will tell you how to modify your CONFIG.SYS appropriately.

P.S.

If you are doing these exercises on someone else's computer or on a computer at work, let's now undo what you did in this Section, and put the computer back the way you found it. (If you are on your own computer, don't do this step, because the computer is set up the way you'll want it from now on.) Don't be concerned about any error messages that show up onscreen as a result; if you type exactly what's in the box, nothing bad will happen. Type:

CD\		enterkey
ERASE	C:\CONFIG.SYS	enterkey
RENAME	CONFIG.OFF CONFIG.SYS	enterkey

Then press CTRL-ALT-DEL to restart DOS. Now the computer is back the way it was before we started this Section.

If you ever see a message onscreen that says something about being "out of environment space", you have a problem that you can fix with your CONFIG.SYS file. The problem is that the space in RAM that DOS has set aside to store such information as PROMPT, PATH, and SET statements is so full that DOS won't let you add (for example) more directories to your PATH.

If you have DOS 3.0 or later, you can add more space. If you have DOS older than 3.0, you can't. Those of you with DOS older than 3.0, reduce your space requirements by reducing the size of your prompt (to just PROMPT \$P) and removing the least important directory from your PATH. The rest of you, with at least DOS 3.0, add this line to your CONFIG.SYS file:

```
SHELL C:\COMMAND.COM /E:32 /P
```

and then restart the computer. Only if you again see the "out of environment space" message, re-type that file and line to:

```
SHELL C:\COMMAND.COM /E:512 /P
```

Depending on the version of DOS you have, one of those two commands will at least double the size of the "environment" space.

Section #10

Backing Up

IN THIS SECTION, YOU'LL LEARN WHY AND HOW YOU SHOULD MAKE FLOPPY-DISK COPIES OF IMPORTANT HARD-DISK FILES

You'll eventually have much time invested in the files you've created on the hard disk. Such files include not only your Bat files, but your spreadsheets, databases, and so on. If the hard disk ever dies (and eventually it will), you will lose all these files forever, unless you have a copy elsewhere of the most recent version of each of them.

One way to make sure you have those back-up copies is to get into the habit of making a floppy-disk copy of any important file. You could make such a copy, using the COPY command, immediately after you finish working on the file, each and every time you work on it. If this is the method you choose to do backup, then you can skip the rest of this Section. This Section discusses ways to automate the backing-up of many files at once.

A second way to "backup" is to copy the entire hard disk to floppies (many floppies!) once a week or so, using the BACKUP command that comes with DOS. A third way is to use the XCOPY command, which is available only in DOS version 3.20 and higher; this command acts like a hybrid of BACKUP and COPY, incorporating the best of both.

Whichever you choose (COPY, BACKUP, or XCOPY), I suggest that you do not backup the entire hard disk. Backing-up absolutely everything will take a "long" time and require many disks; the more work backing up is, the less likely you're going to do it regularly. Therefore, the more likely you are to lose something important when your hard disk finally dies.

You already have floppy-disk copies of all your application programs (the original copies; if you want backup copies of them, use DISKCOPY). I suggest that you backup only irreplaceable data files, and that you use a Batch file to automate the process.

BACKUP (the DOS command)

The DOS file BACKUP.EXE contains the BACKUP command. It creates a much more organized backup than does COPY; and the copies it creates are more tightly packed so that you don't have to use as many floppy disks as with COPY. However, these BACKUP-command-created floppy disks are unreadable directly. To ever use them again, you first have to execute the RESTORE.EXE command (also included in your DOS files) to place the files back onto a hard disk in a readable form.

I don't like the BACKUP command, and neither do any of the authors of any of the books that I've read. Nor is it recommended by recent articles in leading computer magazines. The problem is that the RESTORE command can easily screw up or be used incorrectly, possibly resulting in the loss or alteration of the files you thought you'd saved. If you are determined to use this command, go to a software store and buy a "Utility Program" version of it (ask the sales-person for a few recommendations). These commercial amplifications of BACKUP are faster, easier to use, and less likely to screw up.

If you have a large number of files to backup, or if you have large files (so large that a single file doesn't fit on a single floppy disk), then you can't easily use COPY or XCOPY for backing up. You'll likely have to use BACKUP (altho I strongly suggest a commercial version, not DOS's).

COPY

Let's say that the PATHs to your irreplaceable files are

- C:\WORDS\WORDPROC\ROMANCE.DOC (a novel you're writing)
- C:\DATABASE\FRIENDS.DB (addresses of friends)
- C:\SPREDSHT\BUDGET.SS (your finance records)
- C:\DRAW\TURTLE.GR (an important graphic)
- C:\MSBASIC\ANOVA.BAS (a program you wrote in BASIC)
- C:\WORDS\WORDPROC\PRINTER.LQ (a customized font file)

To use COPY for backup purposes, you might want to backup each file as soon as you finish working on it. Or you might want to write the following batch file:


```

CD\                                     enterkey
COPY CON  SHUTDOWN.BAT                 enterkey
COPY C:\WORDS\WORDPROC\ROMANCE.DOC  A:\ enterkey
COPY C:\DATABASE\FRIENDS.DB  A:\     enterkey
COPY C:\SPREDSHT\BUDGET.SS  A:\     enterkey
COPY C:\DRAW\TURTLE.GR  A:\         enterkey
COPY C:\MSBASIC\ANOVA.BAS  A:\     enterkey
COPY C:\WORDS\WORDPROC\PRINTER.LQ  A:\ enterkey
F6key                                  enterkey

```

I've called that file SHUTDOWN.BAT, but you could call it anything you want to that ends in ".BAT" and which does not start with the name of a DOS command. For example, don't call it BACKUP.BAT; because if you do, when you type "BACKUP enterkey", DOS will run its own program (BACKUP.EXE) instead of your Bat file!

Each line in the file translates as "please COPY the file at the end of the following complete PATH, and place that copy into the root directory of whatever disk is in the floppy drive, and if there is an old copy already on the floppy, replace the old copy with the new copy". Whether on the hard or floppy disk, DOS always replaces the copy in the target directory with the copy from the source directory whenever and wherever you use the COPY command; that is, if there is a copy in the target directory.

To use SHUTDOWN.BAT to up-date your floppy-disk copies of those files, all that you need to do is to insert your "backup" floppy disk, and then type "SHUTDOWN enterkey".

One problem with using a COPY-command version of a backup Bat file is that every time you use it, it copies every listed file, even the ones that have not been modified since the last time you backed up. Because of this, if you have a long list of irreplaceable files, the backup process could take a long time, long enough for you to forego backing up frequently.

Another problem is that all the files you want to backup may not fit on a single floppy disk. To solve this problem, you have to get creative. For example, you could write several different shutdown bat files, such as SHUTDWN1.BAT, SHUTDWN2.BAT, etc., corresponding to backup floppy disks that you label #1, #2, etc. When you want to update the files on disk #1, put it into the computer and

type "SHUTDWN1 enterkey". When you want to update disk #2, put it in and type "SHUTDWN2 enterkey". Or come up with your own system. Make it simple, or you won't feel like using it.

XCOPY

(Only DOS versions 3.20 and higher have XCOPY. If your version of DOS is earlier than 3.20, you can skip this topic.)

The XCOPY version of SHUTDOWN.BAT is identical to the COPY version up above except that "XCOPY" is substituted for the word COPY, and that the symbols "/M" are added to the end of each typed line. The floppy-disk copies made by such an XCOPY shutdown file are identical to those made by the COPY version of it; that is, the backup files are readable and usable without having to first use any type of "restore" command. The following is the XCOPY version of SHUTDOWN.BAT:

```
CD\                                     enterkey
COPY CON SHUTDOWN.BAT                  enterkey
XCOPY C:\WORDS\WORDPROC\ROMANCE.DOC A:\ /M enterkey
XCOPY C:\DATABASE\FRIENDS.DB A:\ /M    enterkey
XCOPY C:\SPREDSHT\BUDGET.SS A:\ /M     enterkey
XCOPY C:\DRAW\TURTLE.GR A:\ /M        enterkey
XCOPY C:\MSBASIC\ANOVA.BAS A:\ /M     enterkey
XCOPY C:\WORDS\WORDPROC\PRINTER.LQ A:\ /M enterkey
F6key                                   enterkey
```

The "/M" symbols at the end of each line translate as "backup this file only if the hard-disk copy of it has been Modified since the last time it was backed up". Therefore, an XCOPY shutdown Bat file is not only a fast way to backup your important files, but a smart way: they're backed-up only if you've been working on them lately.

I use an XCOPY shutdown Bat file for backing up. All I do is put one of my "back-up" floppy disks (this can be any formatted floppy I feel like using) into the drive, close the door, type SHUTDOWN, press ENTER, wait about 30 seconds for the activity to stop, put the disk away, and then turn off the computer. Very easy, and very important.

If your backup files won't all fit on one floppy, XCOPY makes it quite easy for you to use only a single SHUTDOWN.BAT file. In such a case, you give the SHUTDOWN command and then watch the screen for a message that says the disk is full or out of space. If you get such a message, you then put in another floppy and re-type the same SHUTDOWN command; DOS remembers the files that have just been backed up and so backs up onto the second floppy only the files that haven't yet been backed up. If this second floppy fills up, you put in a third floppy, and type the command again. It doesn't matter exactly which floppy you put in when.

Nowadays, because I use several backup disks, I end up with more than one backup version of each file. This happens because the first time I backup a particular file, I may use backup-disk #1, but the next time I may use disk #5. Eventually, I run out of space on all the disks. To regain space, I erase all but the newest version of each file. I do end up with a mess on these floppies. However, as I said up above, the only reason to have "backup" floppy-disk copies of what's on the hard disk is in case the hard disk dies. This may not happen for years, and so spending much effort to be neat while you prepare for it seems like a waste of time.

If my hard disk does die, I'll get it replaced. Afterwards, I'll re-load DOS and my application programs. Then I'll go thru my backup floppies and COPY back onto the hard disk the most recent version of each file. Presto, I'm back in business.

Too Many Files !!!

At some point in the future, you may have so many files that writing a shutdown Bat file, and keeping it current, will be an overwhelming amount of work. To handle this type of situation, start adding a special character to the beginning of any important filename.

For example, I add the dollar sign, \$, to the beginning of all files that I create myself that I want backed-up regularly. In addition to the names I already had in my SHUTDOWN.BAT file, I added the following:

```
XCOPY C:\WORDS\WORDPROC\$.*. * A:\ /M enterkey
```

(or COPY C:\WORDS\WORDPROC\\$.*. * A:\ enterkey, if you have a DOS version earlier than 3.20).

Remember that "*.*" means "any name of any length". By putting \$ in front, it means "any name of any length that starts with \$".

Actually, I have several such lines in my SHUTDOWN.BAT file, one for each subdirectory in which I create files. You can use such a trick for any directory that has files you create and name yourself. All other files that you want backed up will have to be named specifically, similar to the way we did it in the first part of this Section. Or, if you want, backup every file in the desired directory, by leaving off the dollar sign; that is, type *.* instead of \$*.*.

If you start to have so many directories that writing a line in a SHUTDOWN file for each subdirectory is getting to be a nuisance, you can use another XCOPY option: you can put the symbols /S after the /M. The /S means "backup not only the directory I just listed, but all its subdirectories, and all their subdirectories, and all their subdirectories, and so on".

For example a librarian friend of mine has subdirectories inside subdirectories inside subdirectories inside subdirectories inside his wordprocessor directory, and he frequently makes new files in even more new subdirectories. To backup all the files in all those subdirectories, he could put the following one line in his SHUTDOWN file:

```
XCOPY C:\WORDS\WORDPROC\*.* A:\ /M /S enterkey
```

The /S part of the command orders DOS to check every subdirectory level in WORDPROC for any file that is not already backed up. The /S command also causes DOS to create subdirectories on the floppy disk, subdirectories that match the organization and names of those on the hard disk. The backup copies of the files are put into subdirectories with the same name as they had on the hard disk.

One Final Thought:

You should also get in the habit of backing up a file while you are working on it. While you are in the act of creating or editing a file, the copy you're working on is stored in RAM (random access memory), which is a temporary storage area. While you're working on something important, if you turn off the power, or the power goes out due to a rainstorm, RAM is erased permanently. Not good, especially if you haven't saved what you've spent the last 2 hours creating.

Most application programs have a way to "save to disk" with just a couple of keystrokes or mouse movements. "Save to disk" means to take a copy of whatever you're working on in RAM and put it onto the disk. (Usually the program gives you a choice of saving onto the hard or floppy disk, but I'm going to pretend here that the only choice is the hard disk.)

If the power goes out after a file is saved to disk, the only information lost is what you created since the last time you saved to disk. That is true because storage on the hard disk is not affected by power outages.

Another way you can lose your work is this: if you load into RAM more than one program (not counting DOS), you might "crash" the computer (actually the program crashes). What happens is that the computer screen freezes up, and you can't get it to do anything. This happens either immediately or later when you try to perform a particular task. A crash can happen for a variety of reasons, but neither the software nor the computer is ever hurt by it.

After a crash, press CTRL-ALT-DEL to re-start the computer. Sometimes even that will not work. You may have to turn the power switch off and then back on again. In any case, you just lost everything that you had stored in RAM!!

Software is available that, at intervals you set, automatically "saves to disk" whatever it is you're working on. However, this is just another program that uses more RAM. It, itself, might crash your program under the right circumstances.

I was trying out just such a RAM-saving program recently; either it or I malfunctioned, the result of which was that it erased all of my computer version of this manual!! When I then used a utility program that is designed to un-erase erased files, even the pieces of the file were unrecoverable! No problem: I simply

took out my backup floppy-disk of this manual and copied it onto the hard disk. Presto, I was back in business.

Instead of buying specialized software, maybe you could set an electronic watch to alarm every 15 minutes or so, as a reminder to save to disk. Do get into the habit of saving often; you'll be very glad you did.

Section #11

Editing Files

IN THIS SECTION, YOU'LL LEARN HOW TO USE DOS TO EDIT FILES YOU'VE CREATED WITH THE "COPY CON" COMMAND.

You may have a long SHUTDOWN.BAT, as I do. If you have to re-type the entire file to make additions or deletions, you'll be tempted not to change it when you should. For example, you may not add a newly created, irreplaceable file to the backup list.

To combat such reluctance, let's now learn the essentials of EDLIN, which is DOS's own version of a wordprocessor. I think of EDLIN as being short for "EDiting one LINe at a time".

Starting with DOS 5, a program called "EDIT" is also available. EDIT is identical to EDLIN in function, but is easier to use. EDIT is so simple to learn that I won't bother explaining it. Read about it in your DOS 5 owner's manual. If you have DOS 5 or above, skip this Section and learn EDIT instead.

Most people hate EDLIN. They find it difficult and unnecessary. I find it easy and indispensable. It's easy because I use just the essentials; I don't bother with any of the flashy or complicated parts. It's indispensable because, altho my regular wordprocessor can also edit COPY-CON-created files, EDLIN is much more convenient when I'm troubleshooting Bat files. Using EDLIN, I don't have to keep turning on and off my wordprocessor before I can try out the new version of the bat file.

WARNING: USE EDLIN ONLY FOR EDITING FILES YOU'VE CREATED YOURSELF USING THE "COPY CON" COMMAND. USING THEM TO SEE WHAT'S INSIDE A DOS PROGRAM FILE OR AN APPLICATION PROGRAM FILE MAY RUIN IT, EVEN IF YOU DON'T CHANGE ANYTHING!

If you have only short files, you can use the COPY CON command to re-type the entire file when you want to change it. If that's the case, you can skip this Section and read it at some future date when you start to have long files. Knowing EDLIN is not an essential skill; don't feel you have to learn this Section, ever.

The EDLIN commands are contained in the DOS file named EDLIN.EXE. This program has only 5 commands that are really needed. All 5 are single characters, not whole words. They are **E**, **D**, **L**, and **I**, which are the first 4 letters of the EDLIN name; the letter N is not used, but typing a Number is the 5th EDLIN command you should know. Let's read a quick explanation of each command, and one other symbol, and then we'll spend lots of time using the commands:

- E** This stands for "please End this editing session and save the new version of the file".
- D** While a file is being edited by EDLIN, each line is automatically assigned a number. The letter D stands for "please Delete the line whose number precedes the letter D".
- L** This stands for "please List out this file, line by line".
- I** This means "please Insert one or more new lines just above the line whose number precedes the letter I".
- #** Typing the Number of a line (1, 2, 3...) means "please let me edit this line only".
- *** EDLIN uses an asterisk, *, as a prompt/cursor; for our purposes, just ignore it altogether.

Those commands won't seem so confusing after you make a file and then edit it. You can use capital or small letters for any of the commands; I will use a mixture so that you don't confuse my capital i (I) with a number 1. Type:

CD\ COPY CON PRACTICE.#1 AAAAA BBBBB CCCCC DDDDD EEEEE F6key	enterkey enterkey enterkey enterkey enterkey enterkey enterkey
---	--

Now let's give the command that allows you to edit that file. The command is "EDLIN" followed by the name of the file that you want to edit. The way I teach EDLIN, that file must be in the current directory. Your file is, so just type:


```
EDLIN PRACTICE.#1 enterkey
```

To see what the file looks like, type:

```
L enterkey
```

What you now see is a line-by-line list of the file you just created, with numbers next to each line. The screen will look something like the following:

```
1: AAAAA
2: BBBBB
3: CCCCC
4: DDDDD
5: EEEEE
```

If this file had more than 23 lines, the L command would cause the start of the file to scroll off the top of the screen, out of sight. If you ever do have a long file that you want to list, instead of typing just L, type the first and last line number you'd like to see, and then the L (use a comma between the two numbers, but no spaces). If you ask to see more than 23 lines, it will again scroll offscreen.

Let's list lines 2 thru 4 only by typing:

```
2,4L enterkey
```

Let's say you meant to type F's instead of C's on line 3. To change line 3, just type:

```
3 enterkey
```

What you now see is:

```
3:*CCCCC
3:*
```

Line 3 has been typed out for you and you have a blank line 3 below it. You can now type anything you want, and then press the ENTER key. Whatever you type becomes the new line 3, and the old line 3 is erased. Now type:

FFFFF	enterkey
L	enterkey

Your file now looks like this:

```
1: AAAAA
2: BBBBB
3: FFFFF
4: DDDDD
5: EEEEE
```

You have just edited your file! But now let's say you want to add back the C's on a line between the F's and D's. To insert a line of C's above line 4, type:

4i	enterkey
CCCCC	enterkey
F6key	enterkey

Notice that with the "i" command only, you have to press the F6 key (which types out a ^Z for you), to let DOS know that you are finished. If you don't, EDLIN will keep inserting lines above the old line 4 forever. Don't worry that you put the ^Z on a line numbered 5; EDLIN ignores any line containing only a ^Z. Now type:

L	enterkey
---	----------

What you will see is:

```
1: AAAAA
2: BBBBB
3: FFFFF
4: CCCCC
5: DDDDD
6: EEEEE
```

You have inserted a line of C's above the old line 4 and have converted the old lines 4 and 5 into lines 5 and 6. (That sentence had to be confusing! Take a moment now to re-read and think about it.) Now, add some lines onto the very end of the file, by typing:

7i	enterkey
XXXXX	enterkey
YYYYY	enterkey
ZZZZZ	enterkey
F6key	enterkey

What you just did was to insert 3 lines above line 7. Well, there were only 6 lines total (no line 7), so DOS just added them to the end of the file. To see, type:

L	enterkey
---	----------

The file now looks like this:

```
1: AAAAA
2: BBBBB
3: FFFFF
4. CCCCC
5: DDDDD
6: EEEEE
7. XXXXX
8. YYYYY
9. ZZZZZ
```

Now let's say you decide that you don't want to have a line of F's (line 3) in your file; type:

3d	enterkey
L	enterkey

What you now see is:

```
1: AAAAA
2: BBBBB
3: CCCCC
4. DDDDD
5: EEEEE
6. XXXXX
7. YYYYY
8. ZZZZZ
```

You just deleted old line 3 altogether. All the lines below the old line 3 have

moved up one number, and you now have only 8 lines in your file instead of 9.

Let's pretend that this file is really your AUTOEXEC.BAT, and that line 2 is your PATH line. Let's say that you just bought a "utility" software program. The instruction manual for the software recommends putting all its files into a subdirectory called UTILITY inside the directory containing your DOS files. Then it says to "add UTILITY to the PATH statement in your AUTOEXEC.BAT file". (The purpose of PATH was explained in Section #4.) To do that now, type:

```
2 enterkey
```

which produces onscreen:

```
2.*BBBBB  
2.*
```

Now type:

```
BBBBB;C:\VEDOS\UTILITY enterkey
```

To end your editing, type:

```
E enterkey
```

That command ends the editing session and saves the new version of the file under the original name.

To make sure everything is just as you expect, type:

```
TYPE PRACTICE.#1 enterkey
```

If you have done the exercise above exactly as I've written it, you will now see:

```
AAAAA  
BBBBB;C:\VEDOS\UTILITY  
CCCCC  
DDDDD  
EEEEE
```

XXXXX
YYYYY
ZZZZZ

What an ace editor you are!

BAK files

Every time you edit a file with EDLIN, DOS takes the old version of the file and saves it. DOS doesn't ask permission to do this, and you can't stop it. The old version gets saved with a new name; the new name for the old version has the same first part (the part before the period) as it had originally, but the part after the period (the "extension") is changed to BAK. I guess this is done to prevent you from editing a file so poorly that you wish you had the original back.

To get everything back the way it was before you started an editing session, use the ERASE command to get rid of the file that now has the original name (this is the new version); and then use the RENAME command to change the BAK file (this is the old version) to the original name.

However, if you don't want the BAK file, just ERASE it. Use the DIR command now to see that you not only have a file called PRACTICE.#1, but also one called PRACTICE.BAK

BEFORE WE GO ON TO THE NEXT SECTION, LET'S GET RID OF BOTH "PRACTICE" FILES BY TYPING:

```
ERASE C:\PRACTICE.* enterkey
```

(If what's in the box above doesn't make sense to you, review the ERASE topic in Section #6.)

Section #12

That's All Folks!

You now know everything you need to know to be an independent user of an IBM-compatible computer that runs on DOS 2.0 thru 3.3X (and 4.0 and above, if you don't use the "DOS shell" option). There definitely is a great deal more you could learn. You could spend all your time becoming an "expert" at DOS. However, it wouldn't help you use your application programs any better.

You'll have many temptations over the coming months to spend time and money on books, software, videotapes, and even audio-cassettes that all claim to make using DOS easier. Have faith, and resist temptation. Use the time and money on new application software instead; that will be a thousand times more useful and rewarding.

You may want to learn more about particular commands, especially DIR, COPY, and FORMAT, as well as "memory management" commands that were introduced with DOS 5. If so, the DOS manual that came with your DOS disks may be all you need. Explanations there will be complex compared to mine, but DOS commands are indeed complex. I have simplified them here to make your introduction to DOS as free of frustration as possible.

I have a relative who knows as much as you do now, and she's considered the "computer expert" at her office. I have a co-worker who, after learning most of this manual, became sought-after by others in his office as a DOS trouble-shooter. You really do know quite a lot!

Stop here if you want, or go on to "Tortoise Tips" (in the next Section). Tortoise tips are a collection of my thoughts and biases on non-DOS matters; they are not essential to your learning DOS.

A final point: if you have been doing these lessons on someone else's computer, or on a computer at work, you may want to put the hard disk back the way you found it. To do so, get rid of the C:\VEDOS directory. It contains the duplicate set of DOS files that we copied for practice. On the other hand, if the original DOS files were cluttering up the Root Directory, you should consider keeping VEDOS

and erasing the Root Directory DOS files (except COMMAND.COM, AUTOEXEC.BAT, and CONFIG.SYS--those are needed in the Root Directory).

Section #13 Tortoise Tips

HERE ARE MY PERSONAL OPINIONS ON SOME NON-DOS TOPICS:

PRINTERS

You will eventually want to own a printer. Half the pleasure of owning a computer is being able to make a "hard copy" (*i.e.*, a paper printout) of your computer creations.

The printer you buy is critically important. Your application software has to be able to "talk" to the printer. Not all software "supports" (that is the jargon term for being able to "talk to") all printers. You may have written an entire book with your wordprocessor program, but your printer may not be able to print out a single syllable.

I personally know of a large company that decided to change the wordprocessor program used on every one of their desktop computers. After a separate copy of the new program was bought for and installed on each computer, it was discovered that the old-fashioned printers they had thruout the company could not work with this new program. All the printers had to be replaced--a very large, unbudgeted item!

Many application programs have a "configuration" step during installation that allows you to easily customize the program to support a particular printer. The program may list hundreds of printers, and you are supposed to choose which printer you want the program to support. If yours is not listed, it may be that your printer wasn't being sold until after that program was created.

In such situations, it often works to choose a different printer made by the same company as your printer. You'd want to choose one that seems as similar to yours as possible. Many companies have built "clones" of IBM or Epson printers; yours might be such a printer, in which case selecting one of the IBM or EPSON choices might work. You can always just choose one from the list, try it out for a

while (that is, print something), and then choose another to try. Hopefully, you can find a "configuration" that works well with your printer.

You can also ask for help at the store where you purchased your computer. There you'll likely find several different printers. Store management should be willing to let you bring in a particular program you're concerned about, set it up on the in-store model of the computer you purchased, attach the printer you're thinking of buying, and try it out.

COMPUTER PAPER

You have a decision to make: to buy cheap paper or to buy expensive paper. The expensive paper is usually described as "clean edge" because when you tear off the perforations, the resulting page has such clean edges that it looks like a sheet of regular typing paper. A case of 2500 sheets costs about \$30 (1991 dollars). Having expensive paper in your printer at all times is only a convenience. It means that when you want to print something on good paper to give to someone else (your teacher, customer, or boss), you don't have to change paper.

PRINTER INK

There are several ways the various types of printers put ink on paper. These ways include inked ribbons, inkjet cartridges, and laser cartridges. These ink parts last a long time. However, when they finally break or otherwise become unacceptable, late in the day, the night before a report is due, you'll wish you had a spare. Buy a spare, store it where you'll remember, and sleep more soundly as a result.

SURGE SUPPRESSORS

According to the experts, buying a quality surge suppressor is like buying an insurance policy. It'll protect against voltage fluctuations and miscellaneous interferences that will make your programs and hardware act goofy. Having one is so important that one computer magazine had a feature article in 1990 on the pro's and con's of different models. Buy one at a computer store, not the hardware store; you can get a top of the line, 4-outlet size for about \$40 (1991 dollars).

SOFTWARE

You get what you pay for. Never was a saying more true than that one, at least in regard to application programs.

However, you can indeed get very good ShareWare programs for about \$3 a disk thru computer-magazine advertisements, or free thru electronic bulletin boards. If you like such programs after trying them out, you're supposed to send the author of the software a small fee. You do risk acquiring computer viruses this way, but very expensive software bought at computer stores has also been shown to be infected.

You can also get free copies of programs from friends, who got them from their friends, who got them from their friends, who made a copy of the original. "Free" here means "pirated, bogus, ripped off, bootlegged..." Not only are such copies illegal, but you run a very good chance of picking up a virus.

I used to work in a company in which every copy of the wordprocessor and the spreadsheet programs was bootlegged. None of the secretaries thruout the company and none of the accountants in the business office had original disks or an owner's manual. Many computers were missing some of the original program files. One computer had 3 separate versions of the spreadsheet program. Another computer had 2 different versions of DOS! Month after month, large blocks of employee time were wasted trying to solve some glitch or other in these programs. I found that almost all such glitches were the result of not having the original program disks and manual.

The makers of software spend tens of thousands of hours putting together an integrated package of tools that won't disappoint you. The more time they spend doing it right, the more it's worth. They deserve to be paid. (No, I do not work for a computer or software company; I'm a bio-tech chemist. I've never even met someone who writes software.)

Your company could be served with a lawsuit if a disgruntled ex-employee writes to the software manufacturer about the bootleg programs on your company's hard disks. Some software manufacturers offer large rewards for such information. I've read of more than one case where such a manufacturer then arranges for the police

to visit your company, armed with a search warrant. The police confiscate the evidence. The evidence is on your hard disks. Therefore, they take away your computers. Won't that be fun. (Thank God you have all your important files backed-up onto floppies!!)

Be legal. Buy the software. Get the manuals. Be eligible for technical support (which is the right to call them up and ask questions). Be eligible for upgrades (which is the right to turn in your old version and get a newer version for a fraction of list price). I cannot stress enough that, by being legal, you'll save time and money in the long run.

You do not need to buy the most expensive programs available. In a recent issue of a leading computer magazine, the program given the highest recommendation for home-offices and small businesses (to do routine wordprocessing and number crunching) had a \$60 list price. This single program "integrated" (that is, combined into one easy-to-use package) not only a wordprocessor, spreadsheet, and database, but also a few other handy tools.

I personally know of another company that purchased separate copies of a high-powered \$500 wordprocessing program for each of its more than 40 desk-top computers. Recently, each program was upgraded to the newest version, at a cost of \$200 per computer. This software is sophisticated enough to be used to write a huge book, yet everyone in this company types only short reports and memos with it. The secretaries who might be able to fully utilize it have their own dedicated wordprocessors instead. The spreadsheet that is loaded onto every computer also costs \$500 per copy, but everyone uses it almost exclusively to create one-page tables, data sheets, and schedules. Is this a wise use of dollars? Wouldn't it have been smarter to buy a \$60 program for such work?

You may be able to cheaply acquire unsold copies of older versions of expensive, high-powered software. This may seem like a good idea; but expensive software comes with fat instruction manuals, which may frustrate you with sheer complexity. Also, high-powered software requires a large amount of memory (RAM), which your computer may not have.

You should assess your needs and then buy the best that suits them. If your needs are simple, buy a simple, inexpensive program. If your needs are complex, buy (don't steal, pirate, copy, or bootleg) what you need.

Many software stores are set up to demonstrate what they sell, that is, to let you sit down and try it out on their computer before you buy it. If they won't, go some place else that will. Why spend hard-earned money on something sight unseen? You really cannot judge a program by its cover. Be sure to evaluate such software by using a computer similar to your own; otherwise, you may jump to a wrong conclusion.

COMPUTERS

As of 1991, there are only 2 major types of stand-alone, desk-top, personal computers: the ones made by or compatible with IBM, and the ones made by or compatible with APPLE. I am not mentioning other brands because I want to keep this discussion short.

The various IBM models that have been manufactured up to 1991 are referred to generically as 8088's, 80286's (or just 286's), 80386's (or just 386's), and 80486's (or just 486's). An 8088 is often referred to as an "XT" and a 286 as an "AT". The XT and AT letters refer to names that IBM gave to the original versions. The numbers refer to the type of chip used as the "brain" of the computer. The higher the number of the chip, the faster and more versatile the computer has the potential to be, if the rest of the hardware (and software) can take advantage of it. Already the manufacturer of these chips, INTEL, has begun discussing when it will have the 586, 686, and 786 chips ready!!!

Some cheap clones (the jargon term for a computer that is compatible with software that works on IBM's computers) are made with higher-numbered chips; the rest of the hardware may or may not be capable of taking full advantage of that chip's speed and power. Most clones are fine, but let the buyer beware!

The software available to IBM clones is countless. Almost all of it runs using DOS. Many of the newer programs have the option of using a mouse instead of the keyboard for many tasks. Starting with DOS 4.0, you can use a mouse to give frequently-used DOS commands, by using the "DOS Shell".

APPLE computers are in 2 main categories: the APPLE II's and the

MACINTOSH's. Both have lot's of software available. The II's are found predominantly in grade schools and children's bedrooms, because much of the software written for it is geared toward the younger crowd. The MAC's are very powerful machines used predominantly in businesses and upscale home-offices. Many people find them easier to use than IBM's, specifically because all the "DOS" commands and most of the application program commands can be given using a mouse to point at words or pictures (icons) instead of typing. On the other hand, some people find them actually more difficult to use than IBM's; that is because many people prefer to use a keyboard rather than a mouse. In many situations with MAC's, you are absolutely required to use the mouse.

MAC's have lots of very good business and scientific software available. Generally, the graphics and graphing software written for the MAC has been superior to that written for the IBM, altho the gap may have closed by now.

Be careful. Be patient. Investigate before you buy. Do you really need a powerhouse computer? You ought to buy a computer that can run the software you want at the speed you need. Take the time to know what you want before you buy; this may involve seemingly endless questioning of your computer-literate friends, as well as visiting many computer stores.

Analogy Time

Let me draw an analogy for you that explains my view of computers and software.

Computers are like cars, and software programs are like gasoline. The real purpose of a car is as a transportation device; a car helps you get where you want to go. There are other transportation devices, but none so personal and so powerful. But without gasoline, a car has no useful purpose.

You can drive an expensive car and use expensive gas in it. You can drive a cheap car and use cheap gas. You can put cheap gas into an expensive car, and expensive gas into a cheap car. Any of these will get you where you want to go.

A personal computer is also a powerful device; it helps you accomplish certain tasks much easier or faster. Without application software, however, a computer has no useful purpose. Whether you "drive" an expensive, state-of-the-art

computer or a basic clone, whether you use a cheap typing program or version 5.1 of the hottest wordprocessor, you still end up with the same words in your letter (for example).

It's easy to be persuaded by Madison Avenue that a fast, high-tech, fully-loaded, luxury sedan is somehow better than a cheap, used, subcompact truck. And it's easy to be convinced that 94 octane fuel somehow works better than 92, even when the owner's manual tells you that all you need is 87 octane.

Similarly, it's easy to be persuaded that an expensive, state-of-the-art, fully-loaded computer is better for you than an inexpensive one that was state-of-the-art 3 short years ago. And it's easy to be convinced that the application program you've been using for the past 3 years (and are completely satisfied with) should be upgraded or replaced by the latest, "dramatically improved" offering.

What I am trying to say is this: do you really need a Rolls Royce if you're just going to the store for a loaf of bread? If you're only going to write a memo, or balance a checkbook, do you really need a 80486 computer with 32,000K of RAM, a 600MB hard disk, a modem/FAX card, and a color laser printer?

With help from friends and sales people, decide what it is you want to accomplish, find programs to do it, and buy a computer that can handle those programs well.

Most IBM-compatible software published prior to 1990 need a minimum of an 8088 computer and DOS v2.0 to run correctly. Starting about 1990, a great many new programs started requiring a minimum 80286 and v3.0 of DOS. There are also many programs that require at least an 80386 and DOS 3.1. Nonetheless, if you have a more powerful computer or a more recent version of DOS than is needed by your software, all that happens is that more speed and/or more tricks are available to you. You don't end up with a better memo, or a more correctly calculated spreadsheet.

Don't forget to have fun

THE END

(for you, it's just the beginning)