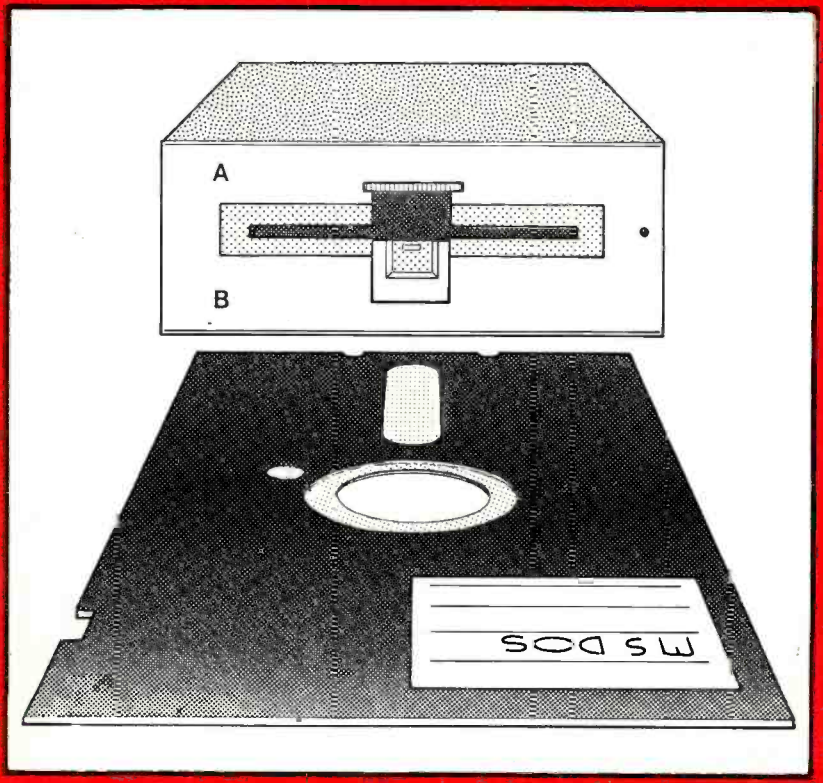
A Concise Introduction to MS-DOS

REVISED EDITION
NOW COVERS VERSIONS
3.3 AND 4.0

N. KANTARIS



A Concise Introduction to MS-DOS

ALSO AVAILABLE

(by the same author)

BP243 BBC-BASIC86 on the Amstrad PCs and IBM Compatibles Book 1 - Language

BP244 BBC-BASIC86 on the Amstrad PCs and IBM Compatibles Book 2 - Graphics and Disk Files

BP250 Programming in FORTRAN 77

BP258 Learning to Program in C

BP259 A Concise Introduction to UNIX

BP260 A Concise Introduction to OS/2

A Concise Introduction to MS-DOS

by
Noel Kantaris

BERNARD BABANI (publishing) LTD.
THE GRAMPIANS
SHEPHERDS BUSH ROAD
LONDON W6 7NF
ENGLAND

PLEASE NOTE

Although every care has been taken with the production of this book to ensure that any projects, designs, modifications andåor programs etc. contained herewith, operate in a correct and safe manner and also that any components specified are normally available in Great Britain, the Publishers and Author do not accept responsibility in any way for the failure, including fault in design, of any project, design, modification or program to work correctly or to cause damage to any other equipment that it may be connected to or used in conjunction with, or in respect of any other damage or injury that may be so caused, nor do the Publishers accept responsibility in any way for the failure to obtain specified components.

Notice is also given that if equipment that is still under warranty is modified in any way or used or connected with home-built equipment then that warranty may be void.

© 1987 and © 1989 BERNARD BABANI (publishing) LTD

First Published – August 1987
Revised Edition – January 1989
Reprinted – September 1989
Reprinted – March 1990
Reprinted – September 1990

British Library Cataloguing in Publication Data:
Kantaris, Noel
A concise introduction to MS-DOS
1. MS-DOS (Computer operating system)
I. Title
005.4'3 QA76.6

ISBN 0 85934 177 1

Typeset direct from disc by Commercial Colour Press, London E7.
Printed and Bound in Great Britain by Cox & Wyman Ltd, Reading

ABOUT THE BOOK

To help the beginner, this concise guide to PC/MS-DOS, has been written with an underlying structure based on "what you need to know first, appears first". However, the book is also circular, which means that you don't have to start at the beginning and go to the end. The more experienced user can start from any section.

This book does not seek to replace the documentation you receive with the PC/MS-DOS operating system, but only to supplement and explain it. The book covers all the latest PC/MS-DOS versions, including the whole range of versions 3.x and version 4.0, as implemented by IBM for their microcomputers, and by Microsoft on behalf of manufacturers of other 'compatible' personal computers, including the AMSTRAD PCs. The book covers both floppy disc-based systems and hard disc-based systems as applicable to the PC, XT, AT, PS/2 and compatible microcomputers.

A separate section of the book deals with the enhancements to be found in the new PC/MS-DOS version 4.0, one aspect of which is substantially different to earlier versions of the operating system. This is due to the addition of a DOS shell, a menu-driven graphical interface, which make this version easier to use by newcomers to the world of DOS. The menus employed in the DOS shell are consistent with those used in Microsoft Windows which, according to the designers of this version, will make it easier for those who are familiar with it to migrate to the OS/2's Presentation Manager - a more advanced operating system which seeks to replace PC/MS-DOS. However, as OS/2 can only run on PCs based around the 80286 and 80386 processors, the millions of users with machines based on the 8086 and 8088 processors will be confined to the PC/MS-DOS operating system. Thus, the DOS shell of version 4.0 will provide the closest link, be it only an artificial one, between DOS and OS/2.

ABOUT THE AUTHOR

Graduated in Electrical Engineering at Bristol University and after spending three years in the Electronics Industry in London, took up a Tutorship in Physics at the University of Queensland. Research interests in Ionospheric Physics, lead to the degrees of M.E. in Electronics and Ph.D. in Physics. On return to the UK, he took up a Post-Doctoral Research Fellowship in Radio Physics at the University of Leicester, and in 1973 a Senior Lectureship in Engineering at The Camborne School of Mines, Cornwall, where since 1978 he has also assumed the responsibility of Head of Computing.

TRADEMARKS

MS-DOC and *Microsoft Windows* are registered trademarks of Microsoft Corporation

IBM, PC, XT, AT, PS/2, PC-DOS and *OS/2* are registered trademarks of International Business Machines corporation

8086, 8088, 80286 and *80386* are registered trademarks of Intel Corporation

Concurrent DOS and *C-DOS* are registered trademarks of Digital Research Corporation

ACKNOWLEDGEMENTS

I would like to thank colleagues at the Camborne School of Mines for the helpful tips and suggestions which assisted me in the writing of this book.

CONTENTS

INTRODUCTION 1
- Structure of MS-DOS 3
- Booting up the System 5

FILES & THE DISC DIRECTORY 7

MANAGING DISC FILES 11
- DATE & TIME Commands 11
- FORMAT Command 11
- SYS Command 13
- Compatibility between disc drives 14
- COPY & DISKCOPY Commands 14
- DISKCOMP & COMP Commands 15
- DELETE Command 16
- RENAME Command 16
- CHKDSK Command 17
- XCOPY Command 17
- COMMAND.COM Processor 18

THE DIRECTORY TREE 19
- Managing Directories 22

THE EDLIN LINE EDITOR 25
- Insert Command on a New File 26
- List Command 26
- Edit Command 27
- The Insert Command on an Existing File 27
- Delete Command 28
- Move and Copy Commands 28
- Search Command 28
- Search and Replace Command 29
- Exiting edlin 30

SYSTEM CONFIGURATION 31
- The CONFIG.SYS File 31
- The AUTOEXEC.BAT File 35
- Simple Batch Files 36
- Additional Batch-file Commands 39

MANAGING YOUR SYSTEM 41
Changing the Access Date of a file 41
The SET Command 41
The TYPE Command 41
The VER and VOL Coms 42
The MORE and SORT filter Commands 43
The PRINT and PRTSC Commands 43
The BACKUP and RESTORE Commands 44

DOS VERSION 4.0 47
Installation 48
DOS Shell 48

COMMAND SUMMARY 51

GLOSSARY OF TERMS 63

INDEX 65

INTRODUCTION

Most 16-bit microcomputers use Microsoft's Disc Operating System (MS-DOS) as the prime means of interaction between user and computer. Owners of IBM PCs know this operating system as PC-DOS or DOS, which is IBM's implementation of MS-DOS. The name MS-DOS will be used throughout this book to distinguish this operating system from another popular one, namely Digital Research's Concurrent DOS (or C-DOS).

Since its inception in 1981, PC/MS-DOS has been the standard operating system for personal computers and by now is being used by more than 12 million people. As the number of users increased over the years, so too has the complexity of applications run on their PCs. To meet these ever increasing demands, PC/MS-DOS has also increased its functionality several times in the form of *new* versions, as shown in the table below.

Version	*Date*	*Main changes in functionality*
1.0	1981	Original Disc Operating System
1.25	1982	Support for double-sided discs
2.0	1983	Support for sub-directories
2.25	1983	Support for extended character set
3.0	1984	Support for 1.2MB floppy disc and larger capacity hard disc
3.1	1984	Support for PC networks
3.2	1986	Support for 3½" floppy disc
3.3	1987	Support for PS/2 range of computers
4.0	1988	Support for extended memory (EMS), hard disc partitions beyond 32MB and a graphical interface DOS shell

One aspect of version 4.0 of the PC/MS-DOS operating system makes it different from earlier versions. This is the result of the addition of a DOS shell — a menu-driven graphical interface — which makes this version easier to use for the newcomer to the DOS environment. The menus employed in the DOS shell are consistent with those used in Microsoft Windows which will make it easier for those who become familiar with it to use the OS/2's Presentation Manager at some later stage. Because of this substantial difference between earlier versions of PC/MS-DOS and version 4.0, the additional functions and advantages gained by the latter's use are discussed in a separate section of this book.

It is assumed here that the reader is familiar with handling floppy discs, floppy and hard disc drives and that the installation manual which comes with every microcomputer has been read and complied with. What this book tries to do is supplement the PC/MS-DOS operating manual by explaining the various commands with ample working examples which is something that these manuals seldom seem to do. They are excellent as reference purposes for those who already know, but learning from them is almost impossible.

The MS-DOS operating system consists of a collection of small, specialised programs that make up the working environment which allows you to create and save programs, copy or delete data files from disc or perform other input and output (I/O) operations, such as finding a program or a file on a particular disc or printing the contents of that file on the printer. In general, MS-DOS is the mirco's administrator and understanding the way it works is very important. Running a computer without understanding its operating system is similar to trying to run a library without any knowledge of librarianship. Very soon chaos will be the order of the day.

At first sight, the various commands within MS-DOS might appear difficult to understand and remember, particularly if the first encounter with them is made through the operating manual which was designed, after all, to act as a reference manual and was never intended to be used to learn from it. However, there is a certain simplicity in the way MS-DOS likes to receive instructions and once this simplicity is understood, using its various commands becomes natural.

This book seeks to bring to the forefront and exploit this inherent simplicity in the MS-DOS operating system by presenting, with examples, the principles of what you need to know, when you need to know them. At the same time, the book has been written in such a way as to also act as a reference guide, long after you have mastered most MS-DOS commands. To this end, a summary of the commands supported by the MS-DOS operating system is given in the last-but-one section of this book. The commands are explained with relevant examples and, as such, the section can serve as a quick reference guide.

The Structure of MS-DOS:

To understand how to use MS-DOS you must understand its underlying structure. The various MS-DOS administrative functions are contained in three, separate, main files (later on, we will explain what files mean and their naming convention). These are:

MSDOS.SYS
IO.SYS
COMMAND.COM

or IBMDOS.COM, IBMBIO.COM and COMMAND.COM, in the case of the IBM PC.

The first file is the core of the operating system, while the second one, also called the Basic Input Output System (BIOS), allows the core to communicate with the hardware. It is the BIOS that is adapted by manufacturers of different hardware so that the operating system can appear to function in the same way, even though there might be differences in hardware design. The last file, COMMAND.COM, is the Command Processor which analyzes what is typed at the keyboard, and if correct, finds and starts execution of the appropriate command.

MS-DOS has over twenty built-in commands, normally referred to as 'internal commands', instantly available to the user as they reside in memory. In addition to these internal commands, there are over forty 'external' commands which are to be found on the System and Utility discs supplied by the manufacturer of the operating system. The machine program which makes up each of these external commands is saved in a file under an appropriate name with a .COM or .EXE extension to the filename (more about this later). Collectively, these internal and external commands make up the computer's Disc Operating System. These commands will be examined in detail in the following sections of this book.

Booting up the System:
To start up the computer, usually referred to as booting up the system, the MS-DOS disc which contains the System files (known as the System disc), must be in the A: drive for a floppy disc-based system, or the files must have been transferred on the C: drive for a hard disc-based system. It is assumed here that you have followed the manufacturer's instructions on how to format the hard disc and transfer the System and all other files from the System and Utility discs onto it. If this is not the case, then do so before going on any further.

In short, when you receive your new computer, you need to run a utility called FDISK (which stands for Fixed Disc) which allows preparation of the hard disc to run DOS or any other operating system. The utility allows you to partition the hard disc in such a way as to allow more than one operating system to reside on it. Normally, you'll be using the hard disc to run MS-DOS only, on one partition. Be very careful with this utility, because proceeding any further with it will format your hard disc automatically, without any warning, with consequent loss of all data that might reside on it. Nonetheless, on formatting the hard disc, the utility transfers onto it the two hidden system files and copies the COMMAND.COM file onto it, but does not copy the rest of the MS-DOS operating system files. To achieve this, and assuming that the MS-DOS system disc is still in the A: drive, type

```
A>COPY A:*.* C:
```

repeating the command with the second (utility) disc, if there is one, inserted in the A: drive. However, before you start using the FDISK utility, make sure that your computer dealer has not actually done all this for you, in which case FDISK is best left alone. Also, before using this utility, make absolutely sure that you have read and complied with your manufacturer's instructions on the subject.

If, after successfully booting up the system (indicated by the appearance of a prompt A> or C>, for booting from a floppy or hard disc, respectively), you were to type DIR (followed by 'Return') to find the contents of the disc, neither of the first two System files (MSDOS.SYS and IO.SYS) would appear on the DIRectory as they are hidden so that you can not delete them. Only the third file (COMMAND.COM) would be displayed.

In addition to these three special files, there are a number of other files supplied by MS-DOS which perform various important tasks. These files are collectively known as the MS-DOS utilities and will be examined in detail later. To be able to distinguish between disc drives, MS-DOS refers to them by a letter followed by a colon, e.g. A: or C: for the prime drive of the appropriate system. In a twin disc-based system, there are two drives; A: and B:, with drive A: being the leftmost or uppermost of the two, while on a hard disc-based system there is a floppy disc drive, A: and a hard disc drive, C:. Users on networked systems can access a network hard disc by assigning it as another drive on their micro, namely as E: or Z:

On booting up the micro from a PC/MS-DOS System disc, the following tasks are performed:

(a) A self test on its Random Access Memory (RAM) is performed

(b) A check is made to see if a floppy disc is in drive A:, and if there is, whether it is a System disc. If it is, it boots the system from the A: drive

(c) If no floppy exists in drive A:, an attempt is made to boot the system from drive C:, if there is one, otherwise in the case of the IBM, it goes into Read Only Memory (ROM) based BASIC

(d) Configures the system by executing the CONFIG.SYS file

(e) Reads the BIOS and the MS-DOS operating system

(f) Loads into RAM the COMMAND.COM file so that internal commands can be made available instantly

(g) Executes the commands within the AUTOEXEC.BAT file, if one exists, otherwise it asks for the Date and Time which can be reset at this point. Pressing the Return key, confirms what is displayed.

Should you receive any error message while these tasks are being performed, you could restart the process, after rectifying the error, by pressing simultaneously the three keys marked **Ctrl**, **Alt** and **Del**. This will reboot the system.

MS-DOS has over twenty internal commands built into it which are instantly available as they reside in memory. These are:

Command	*Meaning*	*Command*	*Meaning*
BREAK	Set Ctrl Break on/off	CD	Change directory
CLS	Clear Screen	COPY	Copy file(s)
CTTY	Change I/O device	DATE	Display/set date
DEL	Delete file(s)	DIR	Display directory
ECHO	Set Echo on/off	EXIT	Exit Command level
MD	Make directory	PATH	Search alternative directories
PROMPT	Change prompt		
REN	Rename file(s)	RD	Remove directory
SET	Change parameters	TIME	Display/set time
TYPE	Display a file	VER	Display DOS version
VERIFY	Check disc writing	VOL	Display disc volume

These commands (to be explained later), together with the rest of the operating system, occupy some 40 Kbytes of RAM, as they are loaded into memory on booting up the system.

However, MS-DOS provides over forty additional commands which, to avoid eating up more of the computer's memory, reside on the System disc. These are known as external commands and can only be invoked if that disc is inserted in either drive A: or B: (for a floppy disc-based system). For a hard disc-based system, these additional commands would have been transferred onto drive C: and can, therefore, be accessed directly from that drive.

FILES & THE DISC DIRECTORY

To see what files are held on the System disc, you must access the disc directory by typing DIR at the A> prompt, as follows:

```
A> DIR          (and press the Return key)
```

The A> prompt will be displayed on booting up the system if you are using a floppy disc-based system, otherwise the prompt will be C>. Amongst the many files to be listed will be the following:

Filename	*Extension*	*Size*	*Date*	*Time*
APPEND	COM	1725	18-05-87	12:00
COMMAND	COM	24844	10-07-87	12:00
MORE	COM	2377	18-05-87	12:00
ATTRIB	EXE	8776	18-05-87	12:00
BACKUP	EXE	23230	18-05-87	12:00
CHKDSK	EXE	10272	18-05-87	12:00
DISKCOPY	EXE	4992	18-05-87	
EDLIN	EXE	8018	18-05-87	
FORMAT	EXE	14567	18-05-87	
LABEL	EXE	3646	18-05-87	
PRINT	EXE	11840	18-05-87	
RESTORE	EXE	21632		
SORT	EXE	2794		
SYS	EXE	5232		
ANSI	SYS	1902		
CONFIG	SYS	77		
VDISK	SYS	6221		

Note that a filename consists of up to 8 alphanumeric characters (letters and numbers only) and has a three letter extension, separated from the filename by a period, i.e. COMMAND.COM or CONFIG.SYS, without any spaces in between, unlike the listing appearing on your screen, where the periods have been omitted and the extensions have been tabulated. Some of these files might have different extensions from the ones shown above, i.e. BUCKUP.EXE might appear as BACKUP.COM in your system, as the extensions tend to differ for different versions of PC/MS-DOS. The size of each file (in bytes) is also given on the listing together with the date and time it was created, which again might differ for different versions.

The extensions .COM, .SYS and .EXE are the most common extensions of the files which make up MS-DOS. They contain instructions which are executed directly by the computer. Other extensions commonly used by programs or users are:

```
.BAK    .BAS    .BAT    .DAT    .DOC    .TXT    .TMP
```

which indicate "backup" files, "Basic" programs, "batch" files, "data" files, "document" files, "text" files and "temporary" files, respectively.

Returning to the result of issuing the DIR command; what is more likely to have happened in your case is that the listing of the first half of the files on your disc have scrolled out of view. In all, there are approximately forty utility files on the System disc and you can only see the last twenty or so. To stop the scrolling of a long directory, use the /P option after the DIR command, as follows:

```
A> DIR/P
```

which will page the directory, displaying twenty files at a time. Alternatively, you could see all these files on your screen by using the /W option, as follows:

```
A> DIR/W
```

which lists the files sideways, as shown below:

```
APPEND   COM   COMMAND  COM   MORE     COM   ATTRIB   EXE
BACKUP   EXE   CHKDSK   EXE   DISKCOPY EXE   EDLIN    EXE
FORMAT   EXE   LABEL    EXE   PRINT    EXE   RESTORE  EXE
SORT     EXE   SYS      EXE   ANSI     SYS   CONFIG   SYS
VDISK    SYS
```

Note that in this case the information relating to the size of each file and the date and time of its creation has been omitted from the listing.

The slash (/) options in MS-DOS commands, like the /P and /W in the DIR command above, are also referred to as switches. However, as this name could be confused with physical switches, the name options will be used throughout.

You can limit the information which appears on your screen by being more selective with the use of wildcards. For example, to list all the .EXE files on your disc, type

```
A> DIR *.EXE
```

where the wildcard character "*" stands for "all" files.

Note that spaces are very important to MS-DOS. Had you not included a space after DIR in the above command, MS-DOS would have responded with its favoured error message,

```
Bad Command or file name
```

which does not tell you very much, except that MS-DOS does not understand you!

The wildcard character "*" can also be used as part of the filename. For example,

```
A> DIR BAS*.*
```

will list all the files with all extensions on the logged drive (in this case drive A:), starting with the three characters BAS, irrespective of the ending of the filenames. The full MS-DOS command should also specify which drive you want to access, but can be omitted if the command refers to the currently logged drive. Thus,

`A> DIR BAS*.*` or

`A> DIR A:BAS*.*` will access the specified files on drive A:, while

`A> DIR B:BAS*.*` will access the specified files on drive B:.

Alternatively, you can change the logged drive by simply typing its identification letter at the prompt. For example,

`A> B:` will change the logged drive, indicated by changing the prompt, to

```
B>_
```

which indicates that the currently logged drive is now B:. All further commands which do not specify a different drive, will access drive B:. To revert back to the previously logged drive, type A: at the B> prompt.

A more precise wildcard is the query character "?" which can be substituted for a single character in a filename. For example, assuming that there are several consecutively

numbered files on your disc with filenames TEXT1.DOC to TEXT999.DOC, typing

```
A> DIR TEXT?.DOC
```

will list all files with the extension .DOC, from TEXT1 to TEXT9, but not those within the range TEXT10 to TEXT999. On the other hand, using two consecutive query characters in the filename, such as

```
A> DIR TEXT??.DOC
```

will list all files with the extension .DOC, from TEXT1 to TEXT99, but exclude those within the range TEXT100 to TEXT999.

To list all the files from TEXT1 to TEXT999 you must use the wildcard character "*" in place of the single query, as follows:

```
A> DIR TEXT*.DOC
```

Finally, typing

```
A> DIR *.*
```

will display all files with all extensions which, of course, has the same effect as typing

```
A> DIR
```

Nevertheless, the *.* is worth noting as it is the most useful three-character combination in MS-DOS and will be mainly used in housekeeping commands to be explained later.

Should you ever want to find out whether a particular file exists on a disc, just type its name after the DIR command. If the file exists, MS-DOS will display it, otherwise the message

```
File not found
```

will appear on your screen.

MANAGING DISC FILES

MS-DOS provides several commands which help you to manage your disc files efficiently. Some of these commands are internal and some are external. If the commands under discussion are external commands, it will be pointed out so you can insert the System disc in the logged drive which is the drive indicated by letter on the screen prompt.

The DATE Command:
Typing the command

```
A> DATE
```

at the prompt, evokes the response

```
Current date is dd/mm/yy
Enter new date:
```

at which point you can either type a new date or press Return to indicate that date is not to be changed. The above date format assumes that you have included the command COUNTRY=xxx (or equivalent in the case of PC-DOS 3.3), where xxx is a three digit code representing your country, in your CONFIG.SYS file (to be discussed later), otherwise the date will be shown in mm/dd/yy format.

The TIME Command:
Typing the command

```
A> TIME
```

at the prompt, evokes the response

```
Current time is Hrs:Mins:Secs
Enter new time:
```

at which point you can either type a new time or press Return to indicate that time is not to be changed.

The FORMAT Command:
One of the first things you will need to do, as a new user, is to make a working copy of your System disc, or favoured software package, or just a backup copy of your programs or data. Such packages and/or data are far too valuable in terms of money or time invested in producing them to be

used continually without the safeguard of backup copies. Again, it is assumed that in the case of a hard disc-based system, your hard disc has already been formatted according to your manufacturer's instructions when setting up the system, and that all the MS-DOS external command files have been transferred onto it.

A new floppy disc must be formatted before it can be used by your computer's operating system. A floppy disc that has been formatted in one computer, can only be used in another computer if they are compatible and use the same operating system.

To format a disc, in the case of a twin floppy based system, insert the System disc in the A: drive, as FORMAT is an external command and needs to be loaded into RAM from the System disc. Then insert the new floppy disc in the B: drive and type

```
A> FORMAT B:/S/V
```

In the case of a hard disc-based system, the logged drive will be C: and apart from the prompt being different, the new floppy disc is inserted in the A: drive. In this case, the command is:

```
C> FORMAT A:/S/V
```

Drive A: (or C: in the case of a hard disc) is accessed momentarily, the FORMAT utility file is loaded into RAM and executed. You are then given instructions to insert a floppy disc in drive B: (or A: in the case of a hard disc system), and press Return to begin. Be very careful never to format an already formatted disc (particularly the C: drive), as *all* files that might be on it will be lost.

> The /S option instructs MS-DOS to copy the two hidden system files and the COMMAND.COM file onto the newly formatted disc. This will be required if you intend to use the disc to boot up the system.
>
> The /V option allows you to give a Volume label to your new disc, after formatting is completed.

There are some additional options that can be used with the FORMAT command which, however, are dependent on the type of disc drive being used and size of disc. These are as follows:

Disc type	*Disc size*	*Parameters*
160/180 KB	5.25"	/1, /8, /4
320/360 KB	5.25"	/1, /8, /4
1.2 MB	5.25"	/4, /N, /T
720 kB/1.44 MB	3.5"	/N, /T

where

/1 formats only one side of the disc.

/8 formats 8 sectors per track.

/4 formats 40 tracks with 9 sectors per track for 360 Kbytes using a 1.2 Mbyte high-capacity disc drive. This option must be used if you are using double-density and not high-capacity, double-density discs in a 1.2 Mbyte drive.

/N specifies the number of sectors per track to format, written as /N:9 for nine sectors.

/T specifies the number of tracks, written as /T:40 for forty tracks. To format a 720 Kbytes double-sided disc in a high-capacity 3.5" disc drive (1.44 Mbytes), use options /N:9/T:80.

If options /N or /T are specified, then both parameters must be entered. All other options can be used separately or omitted altogether from the command. Omitting the /S option from the FORMAT command saves disc space.

The SYS Command:
Should you change your mind after you have formatted a disc without the use of the /S option, you can use the external SYS command to transfer the System files from the logged disc drive onto a previously formatted disc, inserted in another drive. The command takes the form:

`A> SYS B:` in the case of a floppy disc-based system, or

`C> SYS A:` in the case of a hard disc-based system.

To successfully transfer the operating system to a disc with this method, the disc must either be newly formatted or else have space on it for the transfer of the operating system by perhaps already having a different version of it on the target disc. Finally, note that the SYS command

transfers only the two hidden files of the operating system which means that you must use the COPY command (see next section) to transfer the COMMAND.COM and CONFIG.SYS files.

Compatibility between 360 Kbyte and 1.2 Mbyte disc drives:
The PC, XT and compatibles have 360 Kbyte double-sided, double-density disc drives. Discs are formatted with 40 tracks per side, 9 sectors per track with 0.5 Kbyte of information per sector, resulting in 360 Kbyte capacity. The AT, XT286 and compatibles have 1.2 Mbytes high-capacity double-sided disc drives. Discs are formatted with 80 tracks per side, 15 sectors per track with 0.5 Kbyte of information per sector resulting in 1.2 Mbytes capacity. However, each track takes the same physical space as that of the 360 Kbyte drive, the difference being that the 1.2 Mbyte drive writes tracks that are half the width of the 360 Kbyte drive. Discs formatted on 1.2 Mbyte disc drives with the /4 option use only one half of the width of each of the 40 tracks. This information can easily be read by a 360 Kbyte drive (as a result of tolerance in signal level), provided the other half of the track is completely clear. Should you now use the 360 Kbyte drive to write to the disc, information is written to the full width of the track which can still be read by the 1.2 Mbyte disc drive (again, as a result of tolerance in signal level).

However, any subsequent writing to such a disc using the 1.2 Mbyte drive, results in changes to only one half of the track width. The result is half a track containing the new information with the corresponding other half of the same track containing the old, half-overwritten information, which makes it impossible for the 360 Kbyte disc drive to make any sense of it.

There are no such compatibility problems arising from the use of 3.5" discs which have been formatted in 720 Kbytes capacity in a high-capacity (1.44 Mbytes) disc drive and subsequently used to read or write to them by either a 720 Kbyte or a 1.44 Mbyte disc drive.

The COPY Command:
To copy all files on the disc in the logged drive to the disc in the B: drive, type

```
A> COPY *.* B:
```

Note the most useful three-character combination in MS-DOS, namely *.* which means "all filenames with all extensions".

However, if you wanted to copy a set of files from drive B: to drive A:, while being logged onto the A: drive, type

```
A> COPY B:*.DOC A:
```

which means: COPY from B: drive all files with extension .DOC to the A: drive.

The /V option can be used at the end of the COPY command to force MS-DOS to verify that the file(s) it has copied can be read. For example,

```
A> COPY FORMAT.COM B:/V
```

will copy the formatting utility file FORMAT.COM from the logged drive to the B: drive and force verification that the file can be read.

The DISKCOPY Command:

Both the formatting and copying can be done in one go by using the DISKCOPY command, as follows:

```
A> DISKCOPY B:
```

which will copy all the files from the logged drive, in this case A:, to the B: drive and format the disk in the B: drive at the same time.

WARNING: It is preferable to use the FORMAT and COPY commands than use the DISKCOPY command when copying all files from one disc to another. The reason is that bad sectors are frozen out when formatting a disc with the FORMAT command and the subsequent use of COPY, avoids these sectors. The DISKCOPY command on the other hand, seeks to make an identical copy (sector by sector) of the original disc which means that it attempts to write on bad sectors, if any, with consequent loss of information.

The DISKCOMP and COMP Commands:

These two external utilities are mostly needed if you use the DISKCOPY command. The first one compares the contents of two discs, while the second one compares the contents of two files. The commands take the following form:

`A> DISKCOMP A: B:`	compares the discs in the A: and the B: drives
`A> COMP B:DISKCOPY.COM`	compares the DISKCOPY.COM file to be found in the A: and the B: drives

The DELETE Command:
Unwanted files on a disc can be deleted, as follows:

`A> DEL EXAMPLE.TMP`	deletes EXAMPLE.TMP on the A: drive
`A> DEL B:EXAMPLE.TMP`	deletes EXAMPLE.TMP on the B: drive
`A> DEL *.*`	deletes all files on the logged drive!

Luckily, the use of the DEL *.* command evokes the response

```
Are you sure? (Y/N)
```

which acts as a safety net. It is a good idea to always check what you are about to DELete from your disc by first using the DIR command. For example, say you intend to DELete all the .TMP files from your disc. First use DIR *.TMP and if what is displayed on screen is what you want to DELete, then type DEL and press the F3 function key. This has the effect of displaying on the screen the last command you typed on the keyboard, minus the characters you typed prior to pressing the F3 key. Thus, DEL replaces DIR and the use of F3 displays the rest of the command. In this way you avoid making any mistakes by re-typing.

The RENAME Command:
The REN command is used to rename files. As an example, let us assume that we want to rename a file on the disc in the logged drive from its current filename OLDFILE.DOC to the new filename NEWFILE.DOC. This can be done as follows:

```
A> REN OLDFILE.DOC NEWFILE.DOC
```

Note the importance of spaces after REN and in between the two file names. The command can be interpreted as:

```
Rename from filename1 to filename2
```

To rename a file on a disc in a disc drive other than the logged drive, the disc drive specification must also be included in the command, as follows:

```
A> REN B:OLDFILE.DOC NEWFILE.DOC
```

Note that, if you intend to rename a file and give it a filename that already exists on disc, you must first delete the unwanted file before renaming, otherwise MS-DOS will refuse to obey your command, informing you that the filename you have chosen already exists on disc.

The CHKDSK Command:
This command checks a disc, reporting whether it is formatted as single or double sided, how many files are stored on the disc, how much space they take and how much space is still available. The command also checks the RAM, reporting on both the total memory available and the number of bytes still free. The command takes the form:

`A> CHKDSK` which checks the disc in the logged drive, or

`A> CHKDSK B:` which checks the disc in the B: drive.

The additional /F option, allows CHKDSK to also do some routine maintenance, namely fixing lost clusters. A cluster is the minimum amount of space (one or more sectors) that can be allocated to a file on disc. Each disc has a file-allocation table (FAT) where a note is kept of which clusters have been allocated to which file.

However, with heavy disc use, the file-allocation table can be corrupted and using CHKDSK will report "lost clusters found". The /F option, converts these into files and gives them the general name FILExxxx.CHK, where xxxx starts with 0000 and increments by 1. These files can then be checked and perhaps deleted if found to be useless.

The XCOPY Command:
The XCOPY command allows us to copy files and directories, including lower level sub-directories, if they exist (see following section for a full explanation of these), to the specified destination drive and directory. The command takes the following form:

```
A> XCOPY source_filespec destination [options]
```

where source_filespec specifies the source file or drive and directory you want to copy and destination can be the drive to which you want this source file to be copied to. Some of the options available (for a full list see the 'Command Summary' section) are as follows:

/D copies source files which were modified on or after a specified date
/P prompts the user with "(Y/N?)" before copying files
/S copies directories and their sub-directories unless they are empty
/V causes verification of each file as it is written

The COMMAND.COM Processor:
This command starts a new command processor that contains all internal commands. This is loaded into memory in two parts: the resident part and the transient part which can be overwritten by some programs in which case the resident part can be used to reload the transient part. The command takes the form:

```
A> COMMAND [options]
```

with the following available options:

/E specifies the environment size in bytes, with a default value of 160 bytes

/P prohibits COMMAND.COM from exiting to a higher level

/C executes a following command

An example of usage of this command, consider the following statement which might appear within another program:

```
COMMAND /C CHKDSK A:
```

which starts a new command processor under the current program, runs the CHKDSK command on the disc in the A: drive, and returns to the first command processor.

The use of this command will become much clearer after we discuss batch files, later on in this book. It will then be possible to appreciate how this command can be used to allow one batch file to run from within another - something that is only possible since the arrival of PC/MS-DOS version 3.3. However, for the sake of completeness, this procedure is explained here, even though it might not be appreciated that much at this point. Thus, assuming that a batch file (which is an executable program each line of which could be activating an MS-DOS command) by the name FIRST.BAT is being executed and the line

```
COMMAND /C SECOND.BAT
```

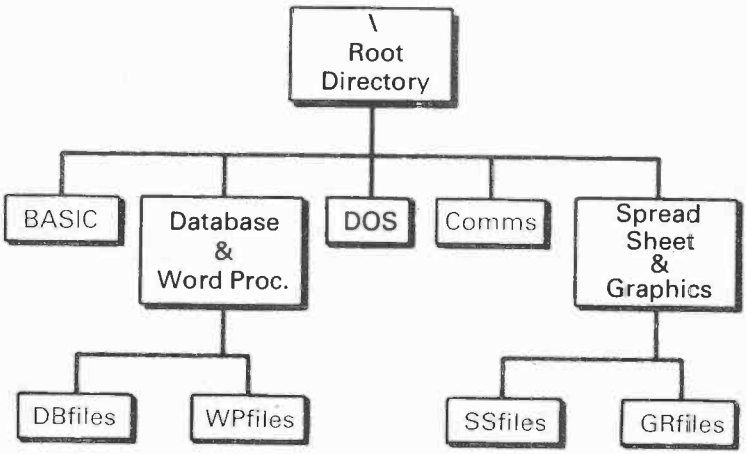
is encountered within it, then a second command processor is loaded and program control is passed to the SECOND.BAT batch file. As soon as execution of this file ends, program control passes back to the next statement within the FIRST.BAT batch file.

THE DIRECTORY TREE

If you are using a system with normal capacity disc drives, then organizing the files you keep on discs is relatively straightforward. The usual method would be to keep similar applications on the same disc, so that one disc might contain files on word processing, another on spread sheets, and another on databases. MS-DOS keeps track of all such files by allocating space on each disc, called a directory, in which such information as the name of each file, its size, the date it was last amended, etc, is kept.

However, as you move up to systems with high capacity disc drives (1.2 Mbytes) and especially to systems with hard discs of 10, 20 or more Mbytes, the amount of information you can store on them increases so much, that unless you organize the way you keep your files on such discs, you could easily spend all of your time trying to find one.

MS-DOS can help you to organize your files on disc by providing a system of directories and sub-directories. The key to MS-DOS' system is the "root" directory, indicated by the back-slash sign (\), which is the main directory under which a number of sub-directories can be created. In turn, each sub-directory can have its own sub-directories, as shown below.



The root directory is shown here with five sub-directories under it, while two of these have their own sub-directories below them. For maximum efficiency, the root directory should contain only the System and start up files, together with information on its sub-directories, a kind of an index drawer to an office filing system.

Files in different sub-directories can have the same name because MS-DOS can be told which is which via a system of PATH names. For example, a file in the SSFILES sub-directory could have the same name, say SALARY87.TMP, as one in the GRFILES sub-directory. Nevertheless, we can instruct MS-DOS to fetch the file in the SSFILES sub-directory by giving its path name which is:

```
\SPREADSH\SSFILES\SALARY87.TMP
```

whereas that of the file in the GRFILES sub-directory is:

```
\SPREADSH\GRFILES\SALARY87.TMP
```

In the example shown previously, the contents of the various sub-directories might be as follows:

	The root directory which contains the two hidden System files MSDOS.SYS and IO.SYS, the Command Processor COMMAND.COM, the CONFIG.SYS file, the AUTOEXEC.BAT file, the names of all its sub-directories (five in our example), and a number of batch files (corresponding to the sub-directories) which allow direct access to the sub-directories from the root directory.
BASIC	A sub-directory containing all the BASIC programs which came on your System disc and the MS-DOS examples disc. These are files which either contain the letters BAS, for example BASICA or GWBASIC, or have a .BAS extension.
DATABASE	A sub-directory containing a database with built-in word processor. Below this, there are two sub-directories; one for the database files (DBfiles), and one for the word processor files (WPfiles). The actual files in these two different sub-directories will most certainly have different extensions; perhaps .DBS for the DBfiles and .DOC for the WPfiles, but the exact extension will be dictated by the actual software package.

DOS — A sub-directory containing all the MS-DOS files comprising the external MS-DOS commands.

COMMS — A sub-directory containing communications programs, propriety backup software, etc.

SPREADSH — A sub-directory containing an integrated spread sheet and graphics package. Below this, there are two sub-directories, one for the spread sheet files (SSfiles), and one for the graphics files (GRfiles). Again, the actual files in these two sub-directories will have different extensions which more than likely will be dictated by the software package.

MS-DOS provides three special commands for the creation and management of sub-directories. these are:

Command	*Meaning*	*Example*
MD	Make sub-directory	C> MD \BASIC
CD	Change directory	C> CD \BASIC
RD	Remove directory	C> RD \BASIC

These will be explained in detail shortly, but before we go any further, it will be extremely useful and prudent to have a prompt which indicates in which directory we are at any given time. We can do this by changing the prompt from always being C>, to indicate the PATH. To achieve this, type

```
C> PROMPT $P$G
```

which after pressing Return, will change the prompt to

```
C:\>_
```

indicating that the current directory is the root directory, shown by the back-slash (\).

This change of the prompt is imperative because without it you could be copying files to the wrong sub-directory without realising it. This command should be included in your AUTOEXEC.BAT file so that it can be executed on booting up the system.

Managing directories:
Before a directory can be used, it must exist. If it does not, you can make it with the MD command.

To make the sub-directory called BASIC, so that you could transfer to it all BASIC programs and files from your MS-DOS System and examples discs, type the following line

```
C:\> MD \BASIC
C:\>_
```

which makes the BASIC sub-directory of the root directory and waits for further commands. Note that the full path was given after the MD command, by specifying first the root directory with the use of the back-slash (\) and then the sub-directory by its name.

To transfer files from a disc in the A: drive, first change directory using the CD command by typing

```
C:\> CD \BASIC
C:\BASIC>_
```

which causes the prompt to change, indicating that MS-DOS has actually changed directory. Without the prompt change, you would have had the typical "where am I?" problem. Note that the moment we create a sub-directory we tend to refer to its parent as directory, even though itself might be a sub-directory to another parent directory.

To copy all BASIC programs and files to this directory, place in turn each MS-DOS disc in the A: drive and type

```
C:\BASIC> COPY A:BAS*.*
C:\BASIC> COPY A:*BAS*.*
C:\BASIC> COPY A:*.BAS
C:\BASIC>_
```

Alternatively, we could have issued these commands from the root directory without first changing directories. As an example, the first line of the three copy commands given above, would have to be typed as

```
C:\> COPY A:BAS*.* C:\BASIC
```

which, however, involves more typing on your part.

Should you be dissatisfied with the name of a directory, you will have to make another directory giving it your preferred name, copy to it all files from the unwanted directory, delete all files from the unwanted directory, and then remove the unwanted directory from its parent directory. This procedure is essential because:

(a) you can not rename directories, and
(b) you can not remove directories unless they are empty.

As an example of the above procedure, let as assume that we have created, as discussed previously, a sub-directory to the root directory, called DATABASE. To have created such a sub-directory, we would have had to return to the root directory from whichever sub-directory we were at the time, by typing

```
CD \
```

at the prompt.

We now proceed to create a sub-directory to the DATABASE directory, called DOCBASE.

To create sub-directory DOCBASE, first change directory from the root directory to that of DATABASE, as follows:

```
C:\> CD \DATABASE
C:\DATABASE>_
```

then make a sub-directory called DOCBASE by typing

```
C:\DATABASE> MD DOCBASE
```

at the prompt. Note that we have omitted the back-slash from in front of the sub-directory name which causes it to be made in the currently logged directory. Had we included the back-slash, the sub-directory DOCBASE would have been created as a sub-directory of the root directory.

Alternatively, we could make DOCBASE without first changing directory by issuing the MD command from the root directory, but giving the full path specification, as follows:

```
C:\> MD \DATABASE\DOCBASE
```

Having made sub-directory DOCBASE, copy into it your files from the A: drive, as discussed previously.

Let us now assume that for some reason the directory name DOCBASE offends you and you would like to change it to WPBASE instead. To do this you will have to type in the following commands, assuming you are at the root directory.

```
C:\> CD \DATABASE
C:\DATABASE> MD WPBASE
C:\DATABASE> CD \DATABASE\WPBASE
C:\DATABASE\WPBASE> COPY \DATABASE\DOCBASE\*.*
C:\DATABASE\WPBASE> CD \DATABASE\DOCBASE
C:\DATABASE\DOCBASE> DEL *.*
Are you sure? (Y/N)Y
C:\DATABASE\DOCBASE> CD \DATABASE
C:\DATABASE> RD DOCBASE
C:\DATABASE>_
```

In order of appearance, these lines of commands do the following:

(a) change directory to DATABASE
(b) make a sub-directory called WPBASE
(c) change directory to WPBASE
(d) copy from sub-directory DOCBASE all files to the logged sub-directory
(e) change directory to DOCBASE
(f) delete all files from logged directory
(g) MS-DOS asks for confirmation
(h) change directory to DATABASE
(i) remove sub-directory DOCBASE.

As you can see, the procedure is cumbersome, so think how you want to structure your hard disc before plunging into it blindly.

Re-structuring directories and sub-directories, moving files from one sub-directory to another, or making backups of groups of files, can be made easy if you use certain propriety software. Such utilities can save you hours of effort and frustration and are well worth the very small initial outlay.

THE EDLIN LINE EDITOR

MS-DOS provides you with a simple line editor, called EDLIN, and you should become familiar with its use. In general, edlin allows the creation and editing of ASCII files. These are text files which when sent to the screen or printer are interpreted as text, unlike the .COM files which are binary.

Edlin can also be used to create the source code of various programming languages, such as Fortran and C. In such cases, remember to give the file the appropriate extension. For the two languages mentioned above, these will be **.for** and **.c,** respectively.

To invoke **edlin**, the MS-DOS System disc or a disc that contains it must be in one drive, and the file you want to create or edit must be specified. Thus, typing the command:

```
C:\> edlin test.txt
```

expects to find both **edlin** and the fictitious file **test.txt** on the disc in the logged drive (in this case C:), while typing

```
C:\> edlin A:test.txt
```

expects to find edlin on the disc in the logged drive and the file test.txt on the disc in the A: drive.

If the file does not exist on the specified disc, then edlin responds with

```
New File
*_
```

and waits for further commands, while if the file already exists, then EDLIN loads the file into RAM and responds with

```
End of input file
*_
```

Note the "*" prompt which is characteristic of **edlin**. Let us now create a text file, called **test.txt**, which we will use to demonstrate the power of **edlin**. To start, type at the MS-DOS prompt

```
C:\> edlin test.txt
```

which should cause **edlin** to respond with

```
New File
*_
```

if that file does not exist on your disc. If it does exist and you do not want to spoil its contents, then type q (for quit) and press the Return key.

The Insert Command on a New File:
To insert lines of text, use the command i (for insert) at the prompt. In the case of a new file, as no lines of text exist in the file, type 1i and then type in the short text given below.

```
*1i
        1:*first line of text
        2:*second line of text
        3:* ^C
*_
```

After typing 1i at the prompt, edlin responds by giving a new line number (in this case 1:) with an asterisk after it to indicate that this is the current line. At this point we type 'first line of text'. On pressing the Return key, **edlin** gives us an additional line number, now 2:*, into which we type 'second line of text'. Again, on pressing Return, we are offered a further line number, and so on. To end the insertion mode, type Ctrl-C. The character ^C is the two-key depression Ctrl-C (hold the key marked **Ctrl** down and press the **C** key).

The List Command:
To see what text is in the file, type l (for list) at the prompt, as follows:

```
*l
        1: first line of text
        2:*second line of text
*_
```

The line numbers are inserted by edlin so that you can refer to the line you want to edit. The * in line 2 indicates that this line was the last to be edited or inserted when edlin was used last. Note that now there is only one current line. Should the file you are listing be very long, listing in this manner causes the current line to appear in the middle of the listing.

To list specific lines, use the l command with line numbers. For example,

```
*5,15l
```

will list lines from 5 to 15 inclusive. Note the syntax of the command which is: "From line number to line number Command". There must be no comma between the second line number and the command letter.

The Edit Mode:
To change the current line, type the new line number and press Return. This puts you in edit mode and will cause the line whose number you typed to be displayed. Pressing Return again, confirms that you are happy with the contents of that line, otherwise you can either press the right cursor key to reveal each letter of that line, or re- type the entire line, making any necessary changes. In our case, we want to change line 2 to

```
second line of text, edited
```

so enter the edit mode and change the line appropriately. This is best done by using the right arrow cursor key to reveal the whole of the existing line and then typing the extra information at the end of it. The Ins and Del keys can also be used to edit the text.

The Insert Command on an Existing File:
To insert lines of text, use the command i (for insert) at the prompt. However, be warned. Using i by its own will insert the new line before the current line (the one with the * after the line number). To insert lines at any other point, give the line number before the command.

In our case, we would like to insert two additional lines after the existing two. To do this, type

```
*3i
        3:*third line of text
        4:*fourth line of text
        5:* ^C
*_
```

Again, insertion mode is terminated in line 5: by pressing **Ctrl-C.** If we now list the contents of the file, we get:

```
*l
        1: first line of text
        2: second line of text, edited
        3: third line of text
        4:*fourth line of text
*_
```

The last line to be inserted becomes the current line.

The Delete Command:
To delete unwanted lines of text, use the d command (for delete) at the prompt. However, if you use the **d** command without any number associated with it, you will delete the current line (the one with the asterisk). Therefore, if you want to delete line 13, say, type

```
*13d
```

or if you want to delete a group of lines, type

```
*13,15d
```

which is translated as "lines 13 to 15 to be deleted".

The Move and Copy Commands:
To move or copy text, use the m or c commands (for move or copy). These commands must be preceded by three numbers, as follows:

```
*13,15,8m
```

which is interpreted as "lines 13 to 15 to be moved to a position before line 8".

Similarly, the c command will copy a block and insert it before the given line. To move or copy a single line, the first two numbers in the command will have to be the same. After moving or copying lines, always use the list command to force renumbering of the file's contents.

The Search Command:
To search for the occurrence of a word or a specified number of characters in a file you have created using **edlin**, use the search command. Just as in the list and delete commands, a line range is first specified, followed by the **s** (for search) command. Thus, typing

```
*1,4s edited
```

evokes the response

```
        2: second line of text, edited
*_
```

which displays the line containing the word 'edited'.

Note that the space between the command s and the word 'edited' becomes part of the search string. Had we been searching for the characters 'con' within the word 'second', we would have had to omit the space between the command s and the string 'con'.

The search command finds only the first occurrence of the specified string. To continue the search for further occurrences of the same string, simply type s again. Thus, typing

```
*1,4sir
        1: first line of text
*s
        3: third line of text
*_
```

causes **edlin** to first find the string 'ir' in the word 'first' of line 1:, then by typing s again, it forces **edlin** to find the same string 'ir' in the word 'third' of line 3:.

The Search and Replace Command:

This command is similar to the search command, except that it requires a replacement string. Thus, typing

```
*1,4r edited ^Z re-edited
```

will cause all occurrences of the word 'edited' to be replaced by the word 're-edited' in all the specified lines of text. Here, of course, it only occurs once in line 2: of the text. The character ^Z is the two-key depression **Ctrl-Z** (hold the key marked **Ctrl** down and press the **Z** key), which acts as a delimiter between the two strings. Again note that the space in front of both words becomes part of both the searching and the replacing strings.

Exiting edlin:
To end the current session and exit edlin at any point, type

```
*e
```

which saves a new file under the chosen filename. However, if the filename already existed on disc prior to using **edlin,** ending **edlin** has the following effect: First the name of the old file on the disc is given the extension **.bak,** then the new file you have created by editing the old one is saved with the original extension. In this way you can make mistakes without disastrous effects since the system makes a backup file of the original. If need be, you could DELete the **.txt** file and then REName the backup file (**.bak**) to its original name and extension.

Note that edlin is disciplined not to allow editing of backup files so, should you want to start using **edlin** to edit the contents of a .bak file, you must first rename it, by giving it a different extension, before proceeding.

If, on the other hand, you realised that too many mistakes were made during editing, you could use the **q** command to quit, as follows:

```
*q
```

instead of using the **e** command as discussed above. Doing this causes **edlin** to ask you whether you want to abort. Typing **y** (for yes), leaves the name and contents of the original file on disc unaltered.

The edlin line editor supports a wealth of extra commands which were not discussed above. The commands presented are more than adequate for writing and editing all but the most difficult programs. If you intend to write complicated programs which might require extensive editing, then it is best to use a full screen editor or your word processor, provided, of course, your word processor can export files in ASCII format.

SYSTEM CONFIGURATION

The CONFIG.SYS file:
This file allows you to configure your computer to your needs, as commands held in it are executed during booting up the system. The easiest way to create or amend this system file is with the use of the line editor **edlin**, as discussed in the previous section.

Do remember that if you are setting up your system for the first time, you will need to create/change the CONFIG.SYS file that exists on your System disc. If your system has already been implemented by, say, your computer staff, do not edit this file or use **edlin** to look at its contents (use instead the **type** command), unless you have to and you know precisely what you are doing, as the file contains entries that MS-DOS uses to define specific operating attributes.

To use **edlin** to change the CONFIG.SYS file, type

```
C:\> edlin CONFIG.SYS
```

which will cause **edlin** to respond (if both the line editor, **edlin** and the CONFIG.SYS file exist on the disc in the logged drive) by

```
End of input file
*_
```

To see what text is in the file, type **l** (for list) at the prompt, which will display the contents of the file as follows:

```
*l
       1: FILES=8
       2:*BUFFERS=2
*_
```

The meaning of each of these statements is explained below. Now edit each line in succession by typing its number and changing it appropriately as given below, then enter the insert mode, by typing 3i, to add the two additional lines. The final contents of the file should be:

```
FILES=15
BUFFERS=25
BREAK=ON
COUNTRY=044
```

To end the insert mode, press **Ctrl-C** on the 5: line and then exit **edlin** by typing **e**. This will save your changed file as CONFIG.SYS and the contents of the old file as CONFIG.BAK. Do refer below for the meaning of each and every statement you have been asked to change or add in your CONFIG.SYS file.

Following is a list of the commands that you can include within the CONFIG.SYS file which MS-DOS supports. However, do remember that any changes made to this file only take effect after re-booting which can be achieved by pressing the three keys marked **Ctrl**, **Alt** and **Del** simultaneously. A brief explanation of the commands is also given.

BREAK — By including the command BREAK=ON in the CONFIG.SYS file, you can use the key combination **Ctrl-C** (hold the key marked Ctrl down and press C) or **Ctrl-Break**, to interrupt MS-DOS I/O functions.

BUFFERS — MS-DOS allocates memory space in RAM, called buffers, to store whole sectors of data being read from disc. The default number of buffers is 2, each of 512 bytes of RAM. If more data are required, MS-DOS first searches the buffers before searching the disc, which speeds up operations. The number of buffers can be changed by using:

```
BUFFERS=n
```

where n can be a number from 1 to 99. However, as each buffer requires an additional 0.5 Kbyte of RAM, the number you should use is dependent on the amount of available memory. Best results are obtained by choosing between 10-30 buffers, the exact number being dependent on the size of the application package being used in relation to the size of your micro's RAM.

COUNTRY — MS-DOS displays dates according to the US format which is month/day/year. To change this to day/month/year, use the command

```
COUNTRY=044
```

where 044 is for U.K. users.

Non U.K. users can substitute their international telephone country code for the 044. The default value is 001, for the USA.

Users of a hard disc with PC-DOS 3.3 should enter this statement as

```
COUNTRY=044,437,C:\DOS\COUNTRY.SYS
```

where 437 is the code page of pre-3.3 versions of DOS and COUNTRY.SYS is to be found in the \DOS sub-directory. In PC-DOS 3.3 the extended IBM character set has been changed to accommodate several versions of it by offering several choices on the characters displayed or printed. If you are a newcomer to DOS then use the multilingual code page 850 in which many of the scientific symbols and box graphic characters have been replaced by international characters. This allows more Europian languages to be supported. However, using any other than code page 437 might cause problems with some application programs as not all would have adapted to the new codes. For more details regarding the new code page numbers and the requirment of using the DEVICE= statement in the CONFIG.SYS file when the new codes are being used, see your DOS reference guide. No additional DEVICE= statements are required with code page 437.

CODEPAGE This command is to be found in PC/MS-DOS versions 3.3 and later. The table that DOS uses to define a character set is called a code page. Thus include the command

```
CODEPAGE=437
```

where 437 is the code page definition of pre-3.3 versions of DOS.

DEVICE MS-DOS includes its own standard device drivers which allow communication with your keyboard, screen and discs. However, these drivers can be extended to allow other devices to be connected by specifying them in the CONFIG.SYS file. Example of these are:

```
DEVICE=ANSI.SYS
```

which loads alternative screen and keyboard drivers for ANSI support — features of which are required by some commercial software.

```
DEVICE=MOUSEAnn.SYS
```

allows the use of specific mouse devices.

```
DEVICE=VDISK.SYS n
```

allows you to specify the size n in Kbytes (default 64) of RAM to be used as an extra very fast virtual disc. With computers which have more than 640 Kbytes of RAM, the option /E can be used after n in the command to allocate the specified memory size from the extra area of RAM.

```
DEVICE=DRIVER.SYS
```

allows you to connect an external disc drive.

```
DEVICE=EGA.SYS
```

provides mouse support for EGA modes.

```
DEVICE=COMn.SYS
```

specifies asynchronous drivers for the serial ports, where for n=01 specifies an IBM PC AT COM device, and n=02 specifies an IBM PS/2 COM device.

FILES

MS-DOS normally allows 8 files to be opened at a time. However, some software such as relational databases, might require to refer to more files at any given time. To accommodate this, MS-DOS allows you to change this default value by using:

```
FILES=n
```

where n can be a number from 8 to the maximum required by your application which usually is 20, although the maximum allowable is 99.

LASTDRIVE This command is used if additional drives are to be connected to your system, or you are sharing a hard disc on a network. The command takes the form:

```
LASTDRIVE=x
```

where x is a letter from A to Z (default E).

SHELL Manufacturers of some micros provide a "front end" or an alternative Command Processor to COMMAND.COM as real-mode command-line processor. To invoke this, the command SHELL must be included within the CONFIG.SYS file. The command takes the form:

```
SHELL=FRONTEND.COM
```

where FRONTEND is the name of the alternative Command Processor. The default value of SHELL is COMMAND.COM.

The AUTOEXEC.BAT file:
This is a special batch file that MS-DOS looks for during the last stages of booting up and if it exists, the commands held in it will be executed. One such command is the KEYBxx which configures keyboards for the appropriate national standard, with xx indicating the country. For the U.K., the command becomes KEYBUK, and you will need to execute it if your keyboard is marked with the double quotes sign on the 2 key and/or the @ sign over the single quotes key and/or the £ sign over the 3 key.

To create the AUTOEXEC.BAT file, use the **edlin** line editor as follows:

```
C:\> edlin AUTOEXEC.BAT
```

which expects to find **edlin** on the disc in the C: drive. If AUTOEXEC.BAT does not exist on this disc, then **edlin** will respond with

```
New File
*_
```

at which point you enter the insert mode, type the command and save the created file, as follows:

```
*i1
        1:*KEYBUK
        2:* ^C
*e
C:\>_
```

Users of a hard disc with PC-DOS 3.3 should enter the KEYBUK statement as

```
1:*KEYBUK 437 C:\DOS\KEYBOARD.SYS
```

where 437 is the code page of pre-3.3 versions of DOS and KEYBOARD.SYS is to be found in the \DOS sub-directory. As mentioned previously under the COUNTRY section, in PC-DOS 3.3 the extended IBM character set has been changed slightly to accommodate several versions of it by offering several choices on the characters displayed or printed. Each such version is referred to by a specific code page number which defines the character set to be used. If you intend to use any other code page than 437, then you should refer to your DOS reference guide.

There are a lot of other commands that can be included in the AUTOEXEC.BAT file, such as setting the PATH, changing directory or executing a program. Such commands will be discussed in detail in the next section. However, in the mean time remember to re-boot the system in order to activate the AUTOEXEC.BAT file after you have created it.

Simple Batch Files:

Naturally, we would like to be able to use the MS-DOS external commands from anywhere within the directory tree without having to specify where the commands are kept (in this instance, we have transferred them into the DOS directory). The same could be said for the programs kept in the COMMS directory. This can be achieved by the use of the PATH command, as follows:

```
PATH C:\;C:\DOS;C:\COMMS
```

which should be included in the AUTOEXEC.BAT file. Note the repeated reference to the C: drive which allows the path to be correctly set even if the user logs onto a drive other than C:.

By now you must have noticed that every time you boot up the system, the commands within your AUTOEXEC.BAT file are echoed (displayed) onto the screen. To clean up the screen of such echoes, change your AUTOEXEC.BAT file to include:

```
ECHO OFF
CLS
PATH C:\;C:\DOS;C:\COMMS
KEYBUK
PROMPT $P$G
ECHO HELLO ... This is your
VER
```

which has the following effect. First echoing is switched off, but only after executing the ECHO OFF command and thus, to clear the screen of the displayed command, we employ the CLS command (for v3.3 users these first two commands could be replaced by @ECHO OFF which eliminates the echoing of the command). Then, the path, keyboard and prompt commands are executed unseen, until echo is re-activated by executing the ECHO command with a trailing message which is displayed on the screen, followed by the version (VER) of your MS-DOS.

To complete the implementation of the hard disc, we need to create a few batch files in the root directory which will help to run the system efficiently. For example, we might require to know the exact name of a DOS or a COMMS command. This can be arranged by creating a batch file for each to display the corresponding directory, whenever the appropriate name is typed. As an example, we will use the **edlin** line editor to create the DOS.BAT file in the root directory, as follows:

```
C:\> edlin DOS.BAT
New File
*1i
          1:*ECHO OFF
          2:*CLS
          3:*CD \DOS
          4:*DIR/P
          5:*CD \
          6:* ^C
*e
C:\>_
```

In line 3:, the directory is changed to that of DOS and line 4: causes the contents of the DOS directory to be displayed using the paging (/P) option. Finally, line 6: returns the system back to the root directory.

Thus, typing DOS, displays the DOS directory, while typing any external MS-DOS command, invokes the appropriate command.

A similar batch file can be built for displaying the COMMS directory, the only difference being in line 3: of the file, so that the correct directory is accessed and displayed. Furthermore, it is most likely that the software packages you will be using in place of our imaginary DATABASE and SPREADSH packages (included as sub-directories to the root directory), require you to type their name in order to activate them. However, most packages also include a second file which is loaded from the first when its name is typed. In such cases you cannot use the PATH command within the AUTOEXEC file to point to the particular package, as MS-DOS will search for the second file in the root directory. To overcome this, you must use the APPEND command, as follows:

```
APPEND C:\;C:\DATABASE
```

and you must include it within the AUTOEXEC.BAT file after the PATH command.

However this command is not implemented in the IBM PC-DOS versions discussed here. Instead, you will have to write a special batch file to do the same job. As an example, let us create the DATABASE.BAT file, again using the edlin line editor, as follows:

```
C:\> edlin DATABASE.BAT
New File
*1i
        1:*ECHO OFF
        2:*CLS
        3:*CD \DATABASE
        4:*DATABASE
        5:*CD \
        6:*C ^
*e
C:\>_
```

which when activated by typing DATABASE, changes the directory to that of DATABASE and loads the file DATABASE. Any other files called from within that file will be found in the correct directory.

Finally, it would be ideal if the language BASIC could be accessed direct from the root directory. However, we can not include the BASIC directory in the PATH command of the AUTOEXEC.BAT file, as we have done with the DOS and COMMS directories, because there are three main versions of the Basic language, two of which are included in the IBM PC-DOS System disc (BASIC and BASICA; A for advanced), and GWBASIC (which is Olivetti's implementation of the language).

Instead, we have to create a rather special batch file in the root directory, again using the EDLIN line editor, as follows:

```
C:\> edlin BAS.BAT
New File
*1i
          1:*ECHO OFF
          2:*CLS
          3:*CD \BASIC
          4:*%1
          5:*CD \
          6:* ^C
*e
C:\>_
```

Note the variable %1 in line 4: which can take the name of any of the three Basic languages mentioned above, provided the appropriate variable name is typed after the batch file name. For example, typing

```
C:\> BAS GWBASIC
```

at the prompt, starts executing the commands within the batch file BAS.BAT, but substituting GWBASIC for the %1 variable. Thus, line 4: causes entry into GWBASIC, provided it exists in the BASIC directory. Similarly, typing

```
C:\> BAS BASICA
```

causes entry into BASICA, again provided it exists in the BASIC directory.

Additional Batch-file Commands:
Apart from the batch-file commands discussed already, there are a number of additional commands which can be useful when writing batch files. These are presented below.

Command	*Action*
FOR	Repeats the specified MS-DOS command for each 'variable' in the specified 'set of items'. The general form of the command is: `FOR %%variable IN (set of items) DO command` where 'command' can include any DOS command or a reference to the %%var. For example,

```
FOR %%X IN (F.OLD F.NEW) DO TYPE %%X
```

will display F.OLD followed by F.NEW

GOTO label — Transfers control to the line which contains the specified label. For example,

```
GOTO end
---
---
:end
```

sends program control to the :end label

IF — Allows conditional command execution. The generaal form of the command is:

```
IF [NOT] condition command
```

where 'condition' can be one of

```
EXIST filespec
string1==string2
ERRORLEVEL=n
```

Each of these can be made into a negative condition with the use of the NOT after the IF command.

REM — Displays comments which follow the REM

SHIFT — Allows batch files to use more than 10 replaceable parameters in batch file processing. An example of this is as follows:

```
:begin
TYPE %1 | MORE
SHIFT
IF EXIST %1 GOTO begin
REM No more files
```

If we call this batch file SHOW.BAT, then we could look at several different files in succession by simply typing

```
SHOW file1 file2 file2
```

as the SHIFT command causes each to be taken in turn.

MANAGING YOUR SYSTEM

MS-DOS provides several commands which help you to manage and control your system's environment. Some of these commands are internal MS-DOS commands and some are external. First we discuss the internal commands.

Changing the Access Date of a file:
If your computer is not fitted with a battery backed clock and you have not been entering the correct time and date on booting up the system, then all your saved files will be showing the default date 1/1/80 in the directory entry. To change this date for a given file, set the current TIME and DATE and type

```
C:\> COPY filespec + filespec
```

where filespec stands for drive, path, filename and extension. Ignore the message "Content of destination file lost before copy" given by MS-DOS when this command has been executed.

The SET Command:
To find out what parameters have been set up, type

```
C:\> SET
```

at the prompt which would evoke the response

```
COMSPEC=C:\COMMAND.COM
PATH=C:\;C:DOS;C:\COMMS
PROMPT=$P$G
```

COMSPEC shows which Command Processor is being used by the system, while PATH and PROMPT display the corresponding commands in your AUTOEXEC.BAT file.

The TYPE Command:
This command allows you to see on screen the contents of text files. The command takes the form:

```
C:\> TYPE filespec
```

This command is useful because it only lets you have a look at the contents of files without changing the environment in any way.

For example, if you ever wanted to find out what is held in either the CONFIG.SYS or AUTOEXEC.BAT files, then use this command rather than the line editor **edlin.**

If the text file you are looking at is longer than one screen full, then use **Ctrl-S** key sequence (while holding down the key marked **Ctrl**, press the **S** key once) to stop the scrolling of the display. Any key will start the display scrolling again.

Using TYPE on other than ASCII files (such as a .COM or .EXE file) could cause your system to "hang" as a result of attempting to display certain sequence of machine code that might be contained in the file. If that happens, use the **Ctrl, Alt, Del** key sequence to re-boot the system.

The TYPE command could be used to direct text files to the printer by typing

```
C:\> TYPE EXAMPLE.TXT >PRN
```

where PRN stands for "printer" which is connected to the parallel printer port.

The VER Command:
To find out which version of MS-DOS/PC-DOS you are currently using, type

```
C:\> VER
```

at the prompt.

The VOL Command:
To find out the volume label of the disc in the logged drive, type

```
C:\> VOL
```

at the prompt. If the disc was not labelled during formatting, then the computer will respond with

```
Volume in drive B has no label
```

otherwise the appropriate label will be displayed.

The MORE filter Command:
This external filter command allows you to view text files a page (23 lines) at a time — you are prompted to press a key to display the next page. As such, it can be combined with other commands to control scrolling of long ASCII files. For example,

```
C:\> TYPE EXAMPLE.TXT |MORE
```

or even used by itself (giving quicker response) as

```
C:\> MORE<EXAMPLE.TXT
```

can help you with viewing long text files if you are not used to or quick enough to use the Ctrl S key sequence to halt scrolling, after issuing the TYPE command.

The SORT filter Command:
One of the ways in which this external command can be used is to sort and display alphabetically the contents of a directory. For example,

```
C:\> DIR |SORT
```

will sort the contents of the logged directory, including the header and footer information, and display the result. For long directories, use this command together with the MORE filter, as follows:

```
C:\> DIR |SORT |MORE
```

to display the sorted directory a page at a time.

A hard copy of the sorted directory of a disc could be obtained by typing

```
C:\> DIR |SORT >PRN
```

which re-directs output through the parallel printer port.

The PRINT Command:
The first time this command is used it has to be loaded into memory as it is an external MS-DOS command. However, from then on it resides in memory and can be used without having to re-load it.

The PRINT command provides background printing, that is, it can print long files while you are doing something else with your computer. In fact, using this command provides you with a print spooler which allows you to make and control a queue of several files for printing. The command takes the form:

```
C:\> PRINT filespec          adds filespec to print queue
C:\> PRINT filespec /C       cancels printing that file
C:\> PRINT /T                terminates all printing
C:\> PRINT                   displays files in queue
```

The PRINT command assumes that you have continuous paper in your printer. There is no facility to pause printing. To print the two text files TEXT1.DOC and TEXT2.DOC, type

```
C:\> PRINT TEXT1.DOC
C:\> PRINT TEXT2.DOC
```

Wildcard characters can also be used in the command, as follows:

```
C:\> PRINT TEXT*.DOC
```

which will spool all the files starting with the characters TEXT and having the extension .DOC to the printer.

The PRTSC Command:
Text which is displayed on the screen can be sent to the printer by pressing the Print Screen **(Shift-PrtSc)** key.

On the other hand, pressing the **Ctrl** and **Print-Screen** keys simultaneously causes re-direction of output to the printer. To cancel the effect, repeat the same key stroke.

The BACKUP and RESTORE Commands:
The external BUCKUP command allows you to archive files from the hard disc. Since your disc contains valuable work, you must make additional copies of all your important files. The BACKUP utility allows you to generate those backup copies on floppy discs. If you have a hard disc, you should use this utility often; daily if necessary. The command takes the form:

BACKUP *source destination options*

where *source* is the drive/path/files to be backed up,
destination is the drive to backup to, and
options are:

/A to add the files to a disc in the destination drive

/D: *date* to backup only files from the specified date onwards

/M to backup only files modified since they were last backed up

/S to also backup sub-directories of the source path.

Thus, to backup, for the first time, all the word processor files whose path is \DATABASE\WPFILES, we type

```
C:\> BACKUP C:\DATABASE\WPFILES\*.* A:
```

while to backup only files modified since they were last backed up, we type

```
C:\> BACKUP C:\DATABASE\WPFILES\*.* A:/M
```

In both cases, the wildcard characters *.* ensures that all files with all their extensions in the WPFILES sub-directory are backed up.

The RESTORE external command allows you to de-archive files. It is the only utility which can restore to the hard disc files previously copied to floppy discs using the BACKUP utility. The command takes the form:

`RESTORE` *source destination options*

where *source* is the drive to restore from,
destination is the drive/path/files to restore, and
options are:

/P to prompt Y/N? before restoring, and
/S to also restore files from sub-directories.

Thus, typing

```
C:\> RESTORE A: C:\DATABASE\WPFILES\*.*/P
```

restores selected files from the floppy disc in the A: drive to the sub-directory WPFILES in the C: drive.

* * *

MS-DOS has many more commands which can be used to control a micro in special ways. However, this is an area which lies outside the scope of this book. What was covered here, together with the summary of the DOS commands given in the last-but-one section of this book, is more than enough to allow effective control of a microcomputer.

* * *

DOS VERSION 4.0

DOS version 4.0 ends the limit of 32MB partition on hard discs by employing a 32-bit File Allocation Table (FAT), instead of the previous 16-bit employed in earlier versions, with the limit of single partitions now being stretched to 4GB. If one of these partitions is less than or equal to 32MB, then it would be possible to boot up the system from that drive with DOS v3.x, otherwise not. The SHARE program is automatically installed if a hard drive with greater than 32MB is set, so programs that will not work with SHARE installed will not run under DOS v4.0 on a large hard disc.

It also supports the expanded memory specification (EMS), as developed by Lotus, Intel and Microsoft, which breaks the 640kB barrier by exploiting the expanded memory boards to store data associated with the BACKUP, FASTOPEN and VDISK commands. These are made possible with the use of two expanded memory drivers — XMA2EMS.SYS and XMAEM.SYS. The first is for use with systems based on the 8086, 8088 and 80286 processors, while the second is for use with the 80386 processor.

Other improvements include extra video mode settings to support displays of greater than 25 lines, as provided by enhanced (EGA) and video (VGA) cards, and graphic screen printing for EGA and VGA graphics adaptors. The GRAPHICS program now allows the 'PrtSc' key to perform a graphics dump from EGA and VGA screens to a variety of IBM printers. The commands required to perform these can be modified for other non-IBM printers. You specify your printer during installation, but make sure that the printer is switched on before booting up the system with DOS v4.0, otherwise you'll be told by the installation program that the printer of your choice is not supported!

Further additions or modifications to DOS commands are described in the section 'Command Summary' with v4.0 commands being marked with a superscripted asterisk (*). Perhaps it is worth mentioning that the resident parts of v4.0 require an additional 15kB of memory above and beyond the requirement of v3.3, with an additional 300kB of memory for loading the DOS shell which however is overwritten when programs are loaded. The use of the DOS shell, which is optional, will be discussed shortly. Finally, error messages have been enhanced so as to at least give an indication as to what might be wrong — a vast improvement on previous versions of the operating system.

Installation:
To install DOS v4.0, you can use the SYS command on discs with a previous installation of DOS, but newcomers or those who want to redefine a drive with more that 32MB will have to format their disc. A new menu-driven program, called SELECT, can guide the user through installation and automatically creates AUTOEXEC.BAT and CONFIG.SYS files.

After the introductory information screens in which it is explained that in order to install DOS v4.0 one requires a blank floppy disc, DOS asks the user to choose between three levels of installation: minimum, balanced and maximum DOS functionality, all of which are dependent on the amount of memory available. Choosing any of these only varies the amount of information that the installation program writes to the AUTOEXEC.BAT and CONFIG.SYS files, with the maximum DOS functionality choice requiring about 7kB more memory. However, since the files thus created are unlikely to fit the needs of users with already implemented systems, the choice of DOS functionality becomes irrelevant. Such users can continue to use their old AUTOEXEC.BAT and CONFIG.SYS files.

SELECT then asks the user about the type of keyboard, printer and the drive and directory onto which to install DOS v4.0, at which point various configuration choices are offered regarding the AUTOEXEC.BAT and CONFIG.SYS files. Once accepted, the program first installs DOS on the blank floppy disc, and then on the hard disc without any further intervention on the part of the user. The installation process creates two new files on the hard disc: AUTOEXEC.400 and CONFIG.400 which contain all the options selected during installation. Renaming these files with their appropriate extensions provides bootable files. Users with already implemented systems must be very careful before renaming these files, as they will loose their existing ones.

DOS Shell:
One aspect of v4.0 of the operating system makes it substantially different to earlier versions. This is the result of the addition of a DOS shell into v4.0, a menu-driven graphics interface, which makes this version easier to use for newcomers to the DOS environment; the experienced user can continue to use the command line. The menus employed in the DOS shell are consistent with those used in Microsoft Windows which will make it easier for those who become familiar with it to migrate to the OS/2's Presentation Manager — a more advanced operating system which seeks to replace PC/MS-DOS.

However, as OS/2 will only run on PCs based around the 80286 and 80386 processors, the twelve million users with machines based on the 8086 and 8088 processors will be confined to the PC/MS-DOS operating system. Thus, the DOS shell of v4.0 will provide the closest link, be it only an artificial one, between DOS and OS/2.

When the program first loads, normally as a result of the last entry in the AUTOEXEC.BAT in the name of DOSSHELL, the user is presented with the Start Program screen (see Fig.1). Along the top and below the date and time bar, there is an action bar with three pull-down menus with the captions Program, Group and Exit. This action bar can be accessed with use of the F10 function key, with F1 providing help. The first two of these options allow the user to add applications to the program list, change their information, or create more than one list. However, to achieve these the user must be capable of writing short code reminiscent to that of batch files.

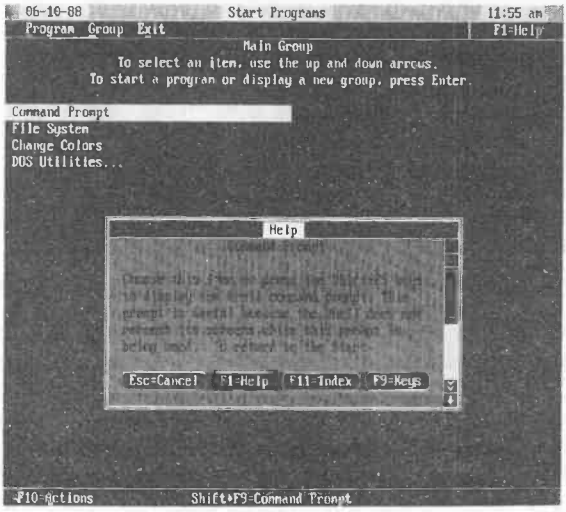


Fig. 1. The Start Program screen with a menu of options.

On the main part of the screen is a menu of options, the first of which is Command Prompt. Choosing this option quits DOSSHELL and returns the user to the command line indicated by the C> prompt. The second menu entry is the File System which is similar to the one supplied with Presentation Manager. It displays the user's sub-directories on the screen, together with their files (see Fig.2). Selecting this option causes a divided screen to be displayed; on the left is a directory tree, while on the right is a list of files within the currently logged directory. Each of the files has an icon alongside it that indicated whether it is an executable file. The option also allows the user to add sub-directories and files, copy, delete or rename them.

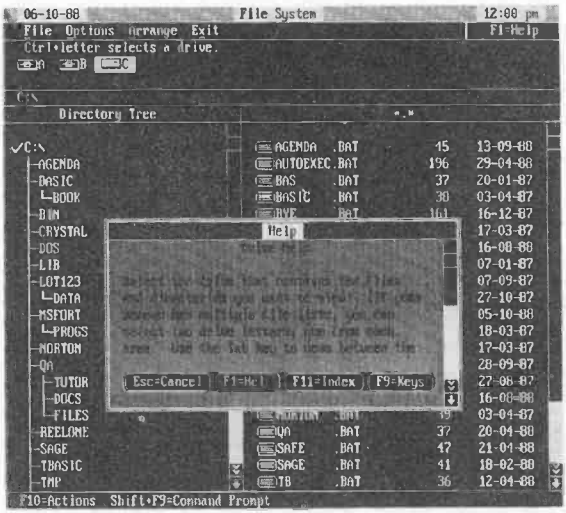


Fig. 2. Choosing the File System from the Start Program.

The third option allows the user to change screen colours, while the fourth allows access to the DOS Utilities such as XCOPY, BACKUP and RESTORE. The three dots following the menu option indicate the existence of a sub-menu. Selection of these options is either by a double click with the mouse, or by using the cursor keys and pressing 'Return'.

No doubt new users will find DOSSHELL very helpful, while experienced users will continue to use the command line.

COMMAND SUMMARY

The following is a summary of the commands supported by the MS-DOS operating environment. For a fuller explanation of both commands and options, consult your system's PC/MS-DOS reference manual. The various commands are labeled internal or external, with external commands being accessible to the user only if the full filespec (drive and path) is given to were the appropriate command file resides. Improvements or additions in v4.0 are marked with a superscripted asterisk (*).

Command	*Explanation*
append*	External — sets a path that MS-DOS will search for files when they are not in the current directory. In v4.0 it can be told not to search already defined paths Example: append c:\wproc\docs searches the \wproc\docs directory on drive c: for files.
assign	External — assigns a drive letter to a different drive. Example: assign a=c allows all references to drive a: to go to the c: drive.
attrib [options]	External — sets or resets the 'read only' attribute and archive bit of a file, and displays the attributes of a file. Options: +r sets read-only mode of a file −r disables read-only mode +a sets the archive bit of a file −a clears the archive bit Example: attrib +R filespec

backup [options]*	External — backs up one or more files from one disc to another. In v4.0 it can also automatically format the destination disc Options: /s backs up sub-directory files to file in current directory /m includes files that have been changed since last backup /a adds files to be backed up to those already on the backup disc without erasing old files /d backs up only those files which were modified after a given date /l makes a backup log entry in a file called BACKUP.LOG Example: backup c:\ a:/s backs up all files on the c: drive onto the a: drive.
break	Internal — sets the Ctrl-C or Ctrl-Break option. Example: break ON
cd (or chdir)	Internal — changes the working directory to a different one. Example: cd\wproc\docs
chcp [nnn]	Internal — selects current code page for as many devices as possible. Omitting 'nnn' displays the current code page.
chkdsk [options]*	External — analyses the directories, files and File Allocation Table on the logged or designated drive and produces a disc and memory status report. In v4.0 it also reports the volume, serial number and disc allocation units Options: /f fixes any problems found during the check

	/v causes the display of filespecs as they are being processed Example: chkdsk a:/f/v
cls	Internal — clears the screen.
command [options]	External — starts the command processor. This is loaded into memory in two parts: the resident part and the transient part which can be overwritten by some programs in which case the transient part is reloaded. Options: /e specifies the environment size in bytes (default = 160 bytes) /p prohibits command.com from exiting to a higher level /c executes a following command Example command /c chkdsk a: starts a new command processor under the current program, runs the chkdsk command on the disc in the A: drive, and returns to the first command processor.
comp	External — compares two files and reports any differences. Example: comp file1 file2
copy [option]	Internal — copies one or more files to specified disc. If preferred, copies can be given different names. Option: /v causes the verification of data written on the destination disc. Example: copy *.exe a:/v copies all files with the .exe extension to the a: drive with verification.

ctty	Internal — changes the standard I/O console to an auxiliary (aux) console, and vice versa. Example: ctty aux moves all input/output from the current device (console) to an aux port such as another terminal. The command 'ctty con' moves I/O back to the console.
date	Internlal — enters or changes the current date.
del [option]*	Internal — deletes all files with the designated filespec. Option v4.0 only: /p displays filenames to confirm deletion Example: del a:*.txt deletes all files which have the extension .txt from the a: drive.
dir	Internal — lists the files in a directory.
diskcomp	External — compares the contents of the disc in the source drive to the disc in the destination drive.
diskcopy	External — copies the contents of the disc in the source drive to the disc in the destination drive.
dosshell*	External — a new v4.0 command which activates the front-end graphical interface
exe2bin	External — converts .exe files to binary format.
exit	Internal — exits the command processor and returns to a previous level.

fastopen [option]* — External — store in memory the location of directories and recently opened files on a specified drive.

Option:
/x allows use of expanded memory. If this option is used, then **/x** option must also be used with the **buffers** command.

fdisk* — External — sets up and partitions the fixed disc for use with MS-DOS and other operating systems. This command is also used to display and change the current active partition. V4.0 supports an 80-column screen. It also has improved user-friendly commands to allow disc partitioning in megabytes or percentages instead of cylinders

fdiskoff — External — could have another name, but it use is to park the fixed disc heads. This should be done before moving a computer equipped with a hard disc to prevent disc damage.

find [options] — External — searches for a specific string of text in a specified ASCII file or files.

Options:
/v displays all lines not containing the specified string
/c prints the count of lines containing the string
/n precedes each occurrence with the relative line number in the file.

Example: find "lost words" chap1

searches for the string 'lost words' (which must appear within full quotes) in the named file (chap1).

format [options]*	External — formats the disc in the specified drive. Options: /8 formats with 8 sectors per track /4 formats a double-sided disc with 40 tracks, 9 sectors per track for 360 kB in a high capacity (1.2 MB) disc drive per track /n specifies the number of sectors per track, i.e. /n:9 for nine sectors /t specifies the number of tracks, i.e. /t:40 for forty tracks /v allows a volume label to be given to the disc /s copies the system files from the logged drive V4.0 also uses: /f:size to specify disc capacity /v:label to allow specification of 'label' without prompting after formating Example: format a:/4/s
graftabl*	External — loads a custom designed, colour graphics font table into memory. In v4.0, the multilingual code page 850 is now supported
graphics*	External — v4.0 now supports EGA and VGA graphics modes to provide screen dumps to IBM Grapgics, Proprinters and compatibles
install*	External — a new v4.0 command which provides an improved method of loading memory-resident pop-up programs in addition to certain DOS commands
join	External — joins a disc drive to a specific path.
keyb [xx]	External — selects a special keyboard layout. Omitting xx returns the current status of the keyboard.

label — External — creates or changes the volume identification label on a disc.

md (or mkdir) — Internal — creates a new directory on the specified disc.

mem* — External — a new v4.0 command which reports the amounts of conventional, expanded and extended memory that are available

mode [options]* — External — sets the mode of operation on a display monitor, parallel/serial printer or the RS232C port. In v4.0 the keyboard repetition and autorepeat start delay time can be set. Also, it allows the setting of the number of rows to any of 25, 43 or 50 on the screen, and there is a wider range of serial-port configurations

Options:

Display: mode [n]

40 sets display width to 40 characters per line

80 sets display width to 80 characters per line

bw40 sets screen to black and white display with 40 characters

bw80 sets screen to black and white display with 80 characters

co40 sets screen to colour display with 40 characters

co80 sets screen to colour display with 80 characters

mono sets screen to monochrome with 80 characters

Printer: mode LPTi: [n][,[m][,p]]

i sets printer number with legal values from 1 to 3

n sets number of characters per line with legal values of 80 or 132

m sets the number of lines per inch with legal values of 6 or 8

p allows continuous reentries on a time-out error

Example: mode LPT1: 132,8

sets the printer in the fist parallel port to 132 characters per line and 8 lines per inch

Serial printer: mode LPTi: = COMj
This command redirects all output sent to one of the parallel printer ports to one of the serial (RS232C) ports. Before using this command, the serial port must be initialised using the 'p' option of the printer mode command
i sets printer number with legal values from 1 to 3
j sets the serial port with legal values of 1 or 2

more — External — reads data from the standard I/O and sends output to the console one screen-full at a time

Example: type read.me | more

displays the contents of the read.me file one screen at a time

nlsfunc — External — provides support for extended country information and allows the use of chcp command to select code pages for all devices defined as having code page switching support.

now — External — displays the current date and time

path — Internal — sets and displays the path to be searched by external commands or batch files

Example: path c:\;c:\dos;c:\comms

will search the root directory as well as the dos and comms sub-directories for files

print [options]	External — can be used to print text files in background mode, while other tasks are being performed. Using the command without options displays files already in the print queue Options: /d specifies the print device such as PRN or AUX /b sets size of internal buffer with legal values from 512 to 16384 bytes, speeding up printing /q specifies the number of files in the print queue, normally 10, with legal values from 4 to 32 /t allows cancellation of files in print queue
prompt	Internal — changes the command prompt Example: pg which allows the path of the current working directory to be displayed as the prompt
rd (or rmdir)	Internal — removes the specified directory
recover	External — recovers a file or an entire disc containing bad sectors
ren (or rename)	Internal — changes the file name Example: ren a:\docs\memo1 memo2 will rename the memo1 file, which is to be found in sub-directory docs on a disc in the a: drive, to memo2
replace [options]*	External — allows easy updating of files from a source disc to a target disc of files having the same name Options: /a also adds files that exist on the source disc but not on the target disc

/d replaces files in the destination disc only if the source files are newer
/p prompts the user before replacing
/r replaces read only files

V4.0 also uses:
/u updates files with a time and date on the source disc more recent than those on the destination disc

restore [options]

External — restores one or more files that were backed up using the 'backup' command

Options:
/s restores files in the specified directory and all files in any sub-directories of the specified directory
/p prompts user before overwriting an existing file by restoring

select*

External — sets the formats for date and time, currency sign and decimal point convention. In v4.0 there are full-screen installation and help routines

set

Internal — sets strings into the command processor's environment. The general form of the command is:

set [name=[parameter]]

Set by itself displays the current environment

share

External — installs file sharing and locking

sort [options]

External — reads data from the console or a file, sorts it and sends it to the console or file

Options:
/r sorts in reverse order

	Example: dir | sort sorts the output of the 'dir' command in alphabetical order
subst	External — allows substitution of a virtual drive for an existing drive and path Example: subst e: a:\wproc\docs will cause future reference to drive d: to be taken as replacement to the longer reference to a:\wproc\docs
switches*	External — a new v4.0 command which forces the conventional keyboard layout on to an enhanced keyboard
sys*	External — transfers the PC/MS-DOS system files from the logged drive to the disc in the specified drive. V4.0 allows the specification of source drive and path commands to transfer system files across a network
time*	Internal — displays and sets the system time. In v4.0 a 12- or 24-hour format is supported
tree*	External — displays the directory structure. In v.40 the display is shown graphically
type	Internal — displays the contents of a file on the console
vdisk	External — allows the setting up of RAM memory as a virtual disc. In v4.0 the /x option can be used to allow the virtual disc to reside in expanded memory
ver	Internal — displays the PC/MS-DOS version number
verify	Internal — allows the verify switch to be turned ON or OFF

Example: verify OFF

vol — Internal — displays the disc volume label, if it exists

xcopy [options] — External — copies files and directories, including lower level sub-directories, if they exist, to the destination drive and directory

Options:

/a copies source files that have their archive bit set

/d copies source files which were modified on or after a specified date

/e copies sub-directories even if they are empty — use this option in conjunction with the /s option

/m copies archived files only, but also turns off the archive bit in the source file

/p prompts the user with "(Y/N?)"

/s copies directories and their sub-directories unless they are empty

/v causes verification of each file as it is written

GLOSSARY OF TERMS

ASCII	It is a binary code representation of a character set. The name stands for "American Standard Code for Information Interchange".
AUTOEXEC.BAT	A batch file containing commands which are automatically executed on booting up the system.
BASIC	A high level programming language. The name stands for "Beginner's All-purpose Symbolic Instruction Code".
BIOS	The Basic Input/Output System. It allows the core of the operating system to communicate with the hardware.
Buffer	RAM memory allocated to store data being read from disc.
Byte	A grouping of binary digits (0 or 1) which represent information.
Cluster	A unit of one or more sectors. It is the minimum amount of space that can be allocated to a file on disc.
COMMAND.COM	The Operating System's Command Processor which analyzes what is typed at the keyboard and causes execution of appropriate commands.
CONFIG.SYS	A special file that allows the system to be configured closer to requirement.
Directory	An area on disc where information relating to a group of files is kept.
DOS	The Disc Operating System. A collection of small specialised programs that allow interaction between user and computer.

FAT	The File Allocation Table. An area on disc where information is kept on which space on disc has been allocated to which file.
File	The name given to an area on disc containing a program or data.
Filespec	File specification made up of drive, path, filename and a three letter extension.
FORTRAN	A high level programming language. It stands for FORmula TRANslation.
Memory	Part of computer consisting of storage elements organised into addressable locations that can hold data and instructions.
Prompt	The System prompt displayed on screen (usually A> or C>).
RAM	Random Access Memory. The micro's volatile memory. Data held in it is lost when power is switched off.
ROM	Read Only Memory. The micro's non-volatile memory. Data are written into this memory at manufacture and are not affected by power loss.
Root Directory	The main disc directory under which a number of sub-directories can be created.
Sector	Disc space, normally 512 bytes long.
SHELL	A front end to MS-DOS or an alternative Command Processor.
SYSTEM	A BASIC command which causes exit from BASIC and return to MS-DOS.
System	Short for computer system, implying a specific collection of hardware and software.
System Disc	A disc containing MS-DOS' three main files and other Utilities.

INDEX

ANSI.SYS 34
APPEND command 38, 51
ASCII files 25
AUTOEXEC.BAT file 5, 21, 35
ASSIGN command 51
ATTRIB command 51

BACKUP command 44, 52
BASIC 9, 38
Batch
- file commands 39
  - FOR 39
  - GOTO 40
  - IF 40
  - REM 40
  - SHIFT 40
- files 36
  - use of KEYBUK 36
  - use of CLS 37
  - use of ECHO 37
  - use of PATH 37
  - use of PROMPT 37
  - use of VER 37
  - use of % 39

BIOS 4
Booting up 4
BREAK command 6, 31, 32, 52
BUFFERS 31, 32
Built-in commands 6

CD command 6, 21, 22, 52
CHCP command 52
CHKDSK command 17, 52
CLS command 6, 53
CODEPAGE command 33
COM.SYS 34
COMMAND.COM 3, 18, 53
Command processor 3
Commands, built-in 6
COMP command 15, 53
Comparing discs 15
Comparing files 15
CONFIG.SYS 5, 31
Configure the system 31
COPY command 6, 14, 53

Copy mode (EDLIN) 28
COUNTRY 31, 32
Ctrl-Alt-Del 5
Ctrl-C 31
Ctrl-S 42
Ctty command 54
Current line (EDLIN) 26
DATE command 6, 11, 41, 54
DELete command 6, 16, 54
Delete mode (EDLIN) 28
DEVICE 33
DIR 4, 7, 54
  /P option 8
  /W option 8
  wildcard characters 9
Directing I/O 42
Directories 4, 19
Disc capacity 13
DISKCOMP command 15, 54
DISKCOPY command 15, 54
DOS
  SHELL command 48, 54
  Versions 1
  v4.0 48
DRIVER.SYS 34
ECHO 6, 37
Edit mode (EDLIN) 27
EDLIN 25
EGA.SYS 34
EXE2BIN command 54
EXIT command 54
Exit EDLIN 30
FASTOPEN command 55
FAT 17
FDISK command 4, 55
FDISKOFF command 55
FILES 31, 34
Filenames 8
Filters 43
FIND command 55
FORMAT command 11, 56
GRAFTABL command 56
GRAPHICS command 56
IBMBIO.COM 3
IBMDOS.COM 3

Input/Output (I/O) 2
Insert mode (EDLIN) 26, 27
INSTALL command 56
IO.SYS 3

JOIN command 56

KEYB command 56
KEYBOARD.SYS 36
KEYBUK command 36

LABEL command 57
LASTDRIVE 35
Line editor 25
List mode (EDLIN) 26

MD command 6, 21, 22, 57
MEM command 57
MODE command 57
MORE filter command 43, 58
MOUSE.SYS 43
Move mode (EDLIN) 28
MSDOS.SYS 3

NLSFUNC command 58
NOW command 58

Options with
- BACKUP 44
- CHKDSK 17
- COMMAND.COM 18
- COPY 14
- DEVICE 33
- DIR 8
- FORMAT 11
- PRINT 43
- RESTORE 44
- XCOPY 17

PATH command 6, 37, 58
PRINT command 43, 59
PRTSC command 44
PRN redirection 42
Prompt 4
PROMPT command 6, 21, 37, 59

Quit EDLIN 30

RAM 5
RD command 6, 21, 24, 59
RECOVER command 59

Redirection 42
RENname command 6, 16, 59
REPLACE command 59
Replace mode (EDLIN) 29
RESTORE command 44, 60
ROM 5
Root directory 19

Search mode (EDLIN) 28
SELECT command 48, 60
SET command 6, 41, 60
SHARE command 60
SHELL 35
SORT filter command 43, 60
Sub-directories 19
SUBST command 61
Switch 8
SWITCHES command 61
SYS command 13, 61
System disc 3

TIME command 6, 11, 41, 61
TREE command 61
TYPE command 41, 61
Text files 25

Utility disc 3

VDISK command 34, 61
VER command 6, 42, 61
VERIFY command 6, 61
VOL command 6, 42, 62

Wildcard characters 9

XCOPY command 17, 62

BERNARD BABANI BP232

A Concise Introduction to MS-DOS

- If you are a PC user and want to get the most out of your computer in efficiency and productivity, then you must learn its PC/MS-DOS operating system. With this CONCISE INTRODUCTION TO MS-DOS you will learn to do just that in the shortest and most effective way.

- The book was written with the non-expert, busy person in mind and, as such, it has an underlying structure based on "what you need to know first, appears first". Nonetheless, the book has also been designed to be circular, which means that you don't have to start at the beginning and go to the end. The more experienced user can start from any section.

- The book explains:

  How the DOS operating system is structured so that you understand what happens when you first switch on your computer.

  How to use the DOS commands to perform various house-keeping operations on your disc files.

  How to use the EDLIN line editor to fully configure your system by writing your own CONFIG.SYS and AUTOEXEC.BAT files.

  How directories and sub-directories can be employed to organize the way you keep files on your disc so that your system's efficiency is maximized.

  How to write batch files to automate the operation and use of your system.

  How to manage your system's environment.

- The book covers all versions of 3.x of both PC-DOS and MS-DOS as implemented by IBM and other manufacturers of "compatible" micro-computers, including the AMSTRAD PCs. Coverage extends to both the floppy disc-based and hard disc-based systems. A separate section deals with the DOS shell of version 4.0, which is a menu-driven graphical interface making this version of the operating system easier to use. Familiarity with this version of DOS will make migration to the new operating system OS/2 that much easier.

ISBN 0-85934-177-1

00295

£2.95

9 780859 341776