# INTRODUCING DOS




MARK STEVENS

SERIES EDITOR: RENZO E. D'ORAZIO

INFORMATION TECHNOLOGY SERIES

COMPUTING
OVERSIZE

005.
4469
STE

First published 1995 by
JOHN WILEY & SONS
33 Park Road, Milton, Qld 4064

Offices also in Sydney and Melbourne

© Jacaranda Wiley Ltd 1995

National Library of Australia
Cataloguing-in-Publication data

Stevens, Mark (Mark William), 1960– .
Introducing DOS.
Includes index.
ISBN 0 471 33648 3.

1. DOS (Computer operating system). 2. Operating systems (Computers). I. Title. (Series: Information technology series (Brisbane, Qld.)).

005.4469

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the publisher.

Clip art used from Corel Ventura Publisher.

Every effort has been made to trace the ownership of copyright material. Information that will enable the publisher to rectify any error or omission in subsequent editions will be welcome.

Series editor: Renzo D'Orazio, DCR Australia

Designed by Diana Eaton — Tulip Publications

Printed and bound in Australia by
Watson Ferguson & Co., Brisbane

10 9 8 7 6 5 4 3 2 1

DUNDEE CITY COUNCIL

| LOCATION | |
|---|---|
| C.T. | |
| ACCESSION NUMBER | |
| COO 212 101X | |
| SUPPLIER | PRICE |
| DAW | £9.99 |
| CLASS No. | DATE |
| 005.4469 | 21/12/01 |

# CONTENTS


**Preface** vii
The Information Technology Series vii
About this module ix
What you need to use this module ix
How to use this module x
Additional assessment material and solutions xi

**Chapter 1 — Operating systems and computer hardware** 1
What is a computer? 2
Mainframes and supercomputers 2
Mini-computers 2
Microcomputers 3
Computer organisation 3
Input devices 4
The keyboard 4
The mouse 4
The trackball 5
Output devices 5
Visual display units 5
Printers 5
Storage devices 6
Primary storage 6
Secondary storage 8
The central processing unit 9
The operating system 10
Chapter summary 11

**Chapter 2 — Operating systems and MS-DOS** 13
Operating system structure 14
Input-output management 14
Command processing 15
Utilities 15
Starting the computer 15
The boot process 16
System programs 16
Command.com 16
Auto execution of files 16
Prompt 17
Primary features of MS-DOS 18
Operating systems and application programs 18
Operating system versions 19
Chapter summary 20

**Chapter 3 — Directory and file management** 23
File names 24
Valid and invalid filenames 25
Standard filename extensions 26
Storing files on a disk 28
Disks and diskettes 28
Saving files 29
Reading files from a disk 30
The MS-DOS directory structure 30
Disk drives 30
Directories 31
Current directory 32
Paths to other directories 33
Default and active drives 34
The PATH command 35
File management 37
Command syntax 37
File manipulation commands 38
Copying files 38
Deleting files 40
Recovering deleted files 40
Renaming files 41
Displaying files 42
The pipe operator 43
Printing files 44
Displaying the directory contents 44
Manipulating directories 45
Change directory 45
Creating directories 46
Deleting directories 47
Displaying the directory structure 47
Absolute and relative pathnames 48
Wildcards 50
Single character matching 50
Multiple character matching 50
Chapter summary 51

**Chapter 4 — Getting help** 55
On-line help for MS-DOS commands 56
DOS command syntax 56
Using the command-line help 59
Using the MS-DOS Help system 59
Getting help — method 1 61
Getting help — method 2 61
Exiting MS-DOS Help 62
Chapter summary 62

**Chapter 5 — Hard disk drives** 65
Hard disk basics 66
Disk formatting 66
Disk storage capacity 67
Disk access times 67
Read/write heads 67
Cylinder organisation 67
Fixed and removable platters 68
Hard disk types 68
Disk organisation 69
Boot record 69
The file allocation table 69
The root directory 70
Formatting and labelling commands 70
Disk labelling 70
Disk formatting 72
Booting from the hard disk 74
Disk management 75
Disk management utilities 75
Chapter summary 78

**Chapter 6 — Floppy disk drives** 81
Disk basics 82
Floppy disks 82
Disk capacities 84
Disk organisation 85
Labelling disks 85
Creating bootable disks 88
Disk management 90
Chapter summary 90
**Answers to self-tests** 92
**Bibliography** 98
**Index** 99

# PREFACE

The following pages give a brief outline of *The Information Technology Series* as well as an overview of this module. Take a little time to read through this section as it explains things such as what the module is about, how it is organised and the recommended approach to working through it. It also explains the purpose of the different sections in the module and the typographical conventions used in the text, examples and exercises.

## THE INFORMATION TECHNOLOGY SERIES

*The Information Technology Series* is a self-paced, modular approach to computing. The series covers a broad range of fundamental and advanced topics in hardware and software as well as application packages. The modules deliver a mixture of theory and practical work and the series is integrated, with more advanced modules building and expanding on previous modules.

The series has modules at various levels of difficulty, from introductory concepts of application packages through to high level computing and information technology theory and applications.

The series is ideal for TAFE, university or industry training programs, as well as individual use by anyone interested in learning the basics of computing. The introductory modules are also suitable for high school computing curricula. Because of the series' self-paced approach, the modules may be studied individually in an open learning environment, as well as in the more traditional classroom environment.

### The project team

*The Information Technology Series* is the result of work by a national team of authors and production people. The authors come from the IT industry and a variety of tertiary institutions. The authors represent a cross-section of people with many years experience in both education and industry.

Renzo D'Orazio, the series editor, has over twenty years experience in education, publishing and industry. He is responsible for the coordination of the authors nationally and for the final technical content of the modules.

The series is continually evolving and expanding. Members of the project team are constantly improving the series. They welcome any feedback and suggestions which will improve any aspect of individual modules or the series as a whole. Comments and suggestions may be sent to:

The Tertiary Publishing Manager,
The Information Technology Series,
Jacaranda Wiley Ltd,
P.O. Box 1226,
Milton, Qld, 4064.

or on e-mail to:

renzo@thomas.DIALix.oz.au

## About the author

Mark Stevens graduated from the University of Western Australia with a degree in Electronic Engineering in 1980.

He then moved to the United Kingdom where he worked as a design engineer on military radar systems before returning to Perth in 1984 and completing a Masters preliminary thesis on logic gate simulation.

Since that time he has lectured in digital electronics, microprocessor hardware, assembly language programming and more recently micro-computer hardware, peripherals and operating systems.

He is currently based at the Advanced Manufacturing and Technology Centre where he is developing a series of computer networking and internetworking modules for the Associate Diploma of Technology — Computer Systems Engineering.

Mark has developed strong links with several companies in the computer networking and communications fields and provides technical support and advice on a consultancy basis.

## Acknowledgements

I would like to dedicate this book to my partner and best friend, Susan, who keeps me going when the going gets tough.

I would also like thank my very good friends and series editor, Renzo, and, his wife Catriona, D'Orazio for their patience, help and amazing capacity for work which has made this book and the series possible.

Special thanks also to my friends and mentors Gene and Helen Rockefeller who have taught me to look for opportunity and move quickly when it presents itself.

## ABOUT THIS MODULE

The intent of this module is to introduce you to the basic concepts of computers and operating systems driven by a *command line interface*. Although specific examples have been drawn from version 6.21 of MS-DOS, most examples are equally valid for earlier versions.

I have also tried to provide a little of the 'how the operating system does it' as well as the 'how to do it' so that you can begin to develop a feeling for how operating systems work in general.

I feel this is important, as it helps you to understand some of the more advanced features of MS-DOS as well as giving good grounding in understanding other operating systems.

In this module we begin by looking at some typical hardware for a microcomputer system and where the operating system fits into this picture. The next step is to break down the operating system function into its three component parts, that is:

- input-output management
- command processing
- utilities

and see how these components are implemented in MS-DOS.

Having established the purpose and extent of the operating system's function, the remainder of the module is dedicated to the specifics of how the operating system performs its various functions, such as booting and communicating with hardware as well as using the operating system in terms of file management, getting help and managing system resources such as hard and floppy disks.

## WHAT YOU NEED TO USE THIS MODULE

To work through this module you will need a number of things:

- your own copy of this book, as well as a workbook for notes and practical questions and exercises
- access to a personal computer, either stand-alone or networked
- Microsoft DOS 6.2 installed on your computer, though most of the commands will work with DOS 5.

While not essential, it is recommended that you refer to the *Microsoft DOS User's Guide*.

## HOW TO USE THIS MODULE

This module can be studied in a self-paced manner or as part of a more formal environment. Either way, there are a number of aspects of the module layout that you should understand.

Each chapter in the book consists of combinations of theoretical and explanatory material interspersed with self-test questions and practical exercises. At the end of each chapter you will find a number of review questions and exercises. Each of the different sections is identified by icons. The following is a brief explanation for each of these.



### Self-test

These questions occur throughout the module. They are designed to see if you have understood the preceding material. Brief answers to these questions are given in the back of the book. Use the answers carefully. If you do not understand the material, re-read the previous section(s) or consult your lecturer or supervisor.




### Exercise

This is a practical exercise on which you need to work. It may involve keyboard work or written answers to questions. It may combine material from a number of previous sections. It tests your ability to combine various aspects of what you have learnt.




This is a list of instructions which you are expected to follow. It is generally found when a point or sequence of events is being demonstrated. These instructions may be combined with explanations or examples.



### Review questions

These are similar to the self-test questions. They are a series of questions which review the theory in the chapter. They are found at the end of the chapter.



### Review exercises

These are also found at the end of the chapter. They are similar to the practical exercises within the chapter, but may combine several concepts. They generally test higher level practical learning outcomes.




### Challenge problems

*These are questions or problems which are found at the end of the chapter. They are generally of a higher degree of difficulty than the material in the module. They are designed to stimulate your thinking about material in the following chapters (or later modules) as well as developing higher order synthesis skills.*

## ADDITIONAL ASSESSMENT MATERIAL AND SOLUTIONS

If you are a lecturer or supervisor you may be able to obtain additional assessment material and solutions to review questions and exercises in this module. These are available from the series editor. Contact the publishers for further details (see page vii).

# CHAPTER 1

# OPERATING SYSTEMS AND COMPUTER HARDWARE

This chapter introduces you to computers, what they are and what they do. You will learn about the main hardware and software components which make up a computer system and how these components work together.

At the end of this chapter you will be able to:

- define the term computer
- describe two common input devices
- describe two common output devices
- define the term primary storage
- define the term secondary storage
- define the term CPU
- describe the relationship between the operating system and the computer hardware.

# WHAT IS A COMPUTER?

A computer can be defined as 'a device designed to accept and store input data, process data and produce appropriate output information under the control of a stored sequence of instructions'.

This definition is very broad. The reason it needs to be so broad is because computers are used for such a wide range of applications, and are constructed in so many different ways that any definition which was more specific would probably not include all types of computers.

Let us examine some examples of where computer technology is used. Computers are used in:

- research organisations to calculate planetary orbits
- breweries to control the automated brewing process
- personal digital assistants (PDAs)
- mobile phones
- video recorders
- singing greeting cards.

In fact, almost all consumer electronics have computers of one type or another embedded in them, even though we may not be aware of their function. However, the kinds of computers that *we* are interested in are those we can directly control or program ourselves. These fall into three categories:

- mainframes or supercomputers
- mini-computers
- microcomputers.



**Self-test**

1.1 In your own words, write a definition of what a computer is.

1.2 List four other examples of where computer technology is used.

## MAINFRAMES AND SUPERCOMPUTERS

**Mainframes** and **supercomputers** are generally designed to perform huge numbers of complex calculations. They support hundreds of users via other smaller computers called **front-end processors**. These types of computers are rarely seen nowadays and are usually operated only by research institutions such as universities or large government departments with offices in several locations.

## MINI-COMPUTERS

**Mini-computers** are the little brothers of the mainframes. They are designed to support many users, but they have to communicate with these users by themselves, without the aid of front-end processors. This type of computer is often found in large businesses.

## MICROCOMPUTERS

IBM coined the term **personal computer** (PC) for the microcomputer. Originally designed to be single user desktop computers, they have now become the most prolific of all the types of computers. Desktop, laptop, notebook and even palmtop computers all fall into this category.




### Exercise

1.1 List three types of computer which fall into the category of microcomputer.

1.2 Name three types of organisations who might still use a mainframe computer.

# COMPUTER ORGANISATION

Computers require three major components to function. These are:

- hardware
- an operating system
- application software.

The hardware has four basic components:

- input devices
- output devices
- storage devices
- the central processing unit (CPU).

The hardware operating system and application software all combine to form a complete computer system, whether it is a mainframe, minicomputer or microcomputer.

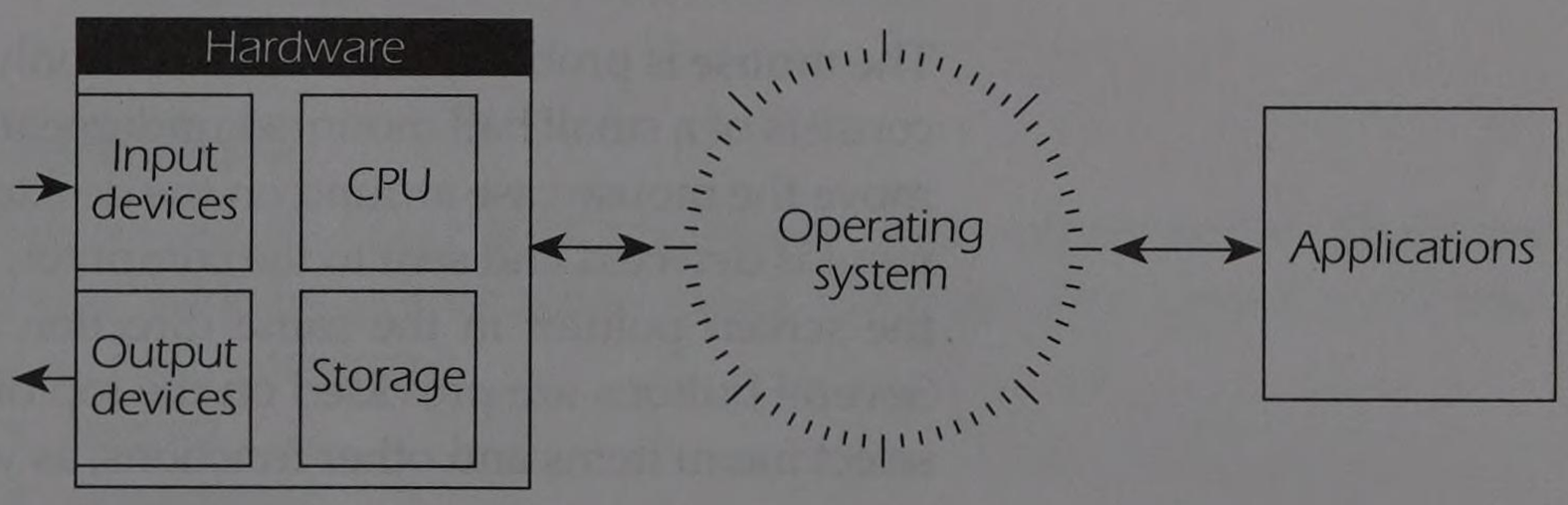



Figure 1.1 — Basic computer organisation

Because of the large numbers of microcomputers in the world today, we will concentrate on this type of computer when looking at the characteristics of input, output, CPUs and storage devices and how these interact with the operating system. The command line interface we use in this module will also concentrate on the workings of microcomputer systems.

# INPUT DEVICES

Input devices for microcomputers include:

- the keyboard
- the mouse
- trackballs.

You do not need to understand the inner workings of these devices, although some explanation is provided here to help you understand their advantages and limitations.

Input devices are used to interact with the computer through its operating system. They pass on instructions provided by you, the user, to the computer system.

## THE KEYBOARD

The **keyboard** is the most common device used to input information into a computer. The type most often used is the QWERTY keyboard with:

- an additional numeric keypad
- direction arrow keys for cursor control
- function keys for application software use.

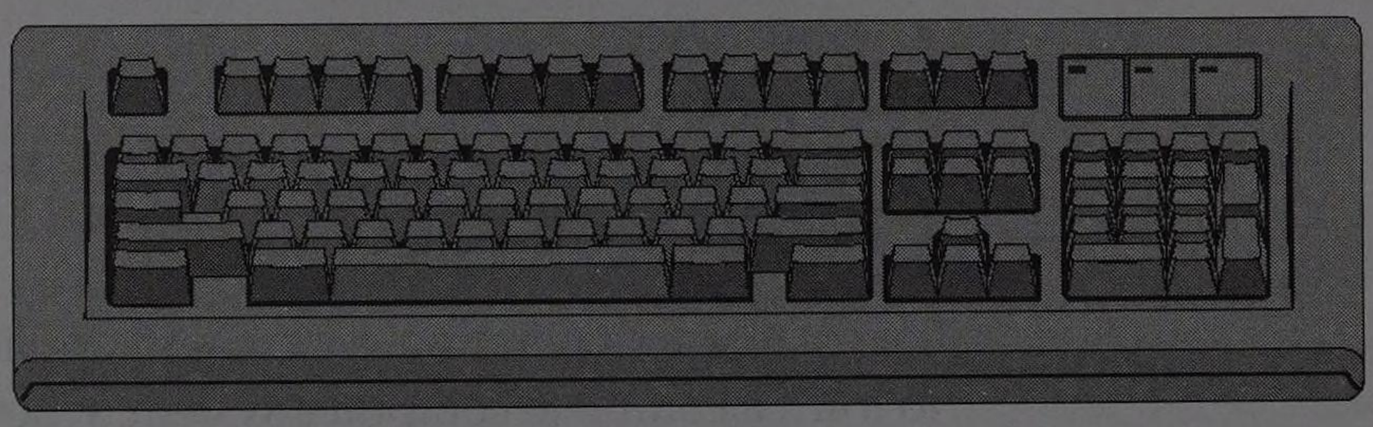


Figure 1.2 — Typical PC keyboard (101 key extended)

## THE MOUSE

The **mouse** is probably the most commonly used direct input device. It consists of a small ball mounted underneath a hand-sized case. As you move the mouse case around on the desktop, the ball rolls. This movement is detected and sent to the computer, which responds by moving the screen pointer in the same direction as the body of the mouse. Several buttons are provided on the top of the mouse so that you can select menu items and other functions, as you require.

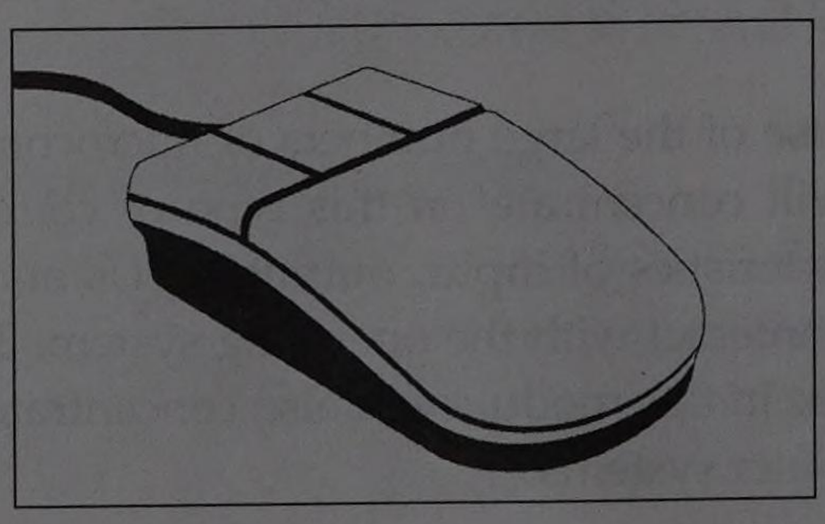


Figure 1.3 — The mouse

### THE TRACKBALL

A **trackball** is a little like an upside down mouse. The (track) ball is mounted in the keyboard or in a stationary box. Instead of moving the mouse body, you rotate the ball itself with your fingers, thus moving the cursor on the screen.



**Self-test**

1.3 List three different types of input devices.




**Exercise**

1.3 Describe the difference between a mouse and a trackball.

1.4 List all the input devices which are attached to your computer.

## OUTPUT DEVICES

There are many different types of output devices available, including visual display units, printers, microfilm, microfiche and plotters. These devices provide you with data and information from the computer, under the control of the operating system. Visual display units and printers are the most commonly used output devices.

### VISUAL DISPLAY UNITS

Visual display units (VDUs) give the user instant feedback. Originally, the primary purpose of VDUs was to display text. Most modern computers now use colour displays as they are ideal for graphics-based applications, where colour can play an important role, in addition to displaying text.

### PRINTERS

Printers are the next most commonly used output device. There are many different printer technologies, although they generally fall into two major categories:

- impact
- non-impact.

**Impact** printers are still the most common type of printer, although their numbers are declining. The main type of impact printer is the **dot matrix,** which forms each character as a pattern of dots. The print quality is dependent on the number of wires or pins in the print head. The more pins in the print head, the greater the dot density and hence the better the character and print quality.

**Non-impact** printers are generally more expensive to purchase, but faster than the impact printers when printing. There are only two types in general use:

- **inkjet** — this type of printer squirts tiny dots of ink onto the paper to form each character. These printers can achieve high resolutions (superior to dot matrix) and are much faster and quieter. Some of these printers are also able to print in colour by having several nozzles, each dispensing one colour. They are much cheaper to buy than laser printers.
- **laser** — these printers are the fastest and produce the best quality print of all the types examined here. They can handle graphics and virtually an infinite variety of fonts, while their cost is relatively low for the performance available.



**Self-test**

1.4 Which type of printer provides the highest quality printing?

1.5 Which of the printers discussed are impact and which are non-impact?



**Exercise**

1.5 Look around your laboratory or office and briefly describe the different types of printers you have access to.

# STORAGE DEVICES

Computer systems have two main types of storage; primary and secondary. Each type has particular uses.

## PRIMARY STORAGE

**Primary storage** is often simply referred to as **memory**. To understand how memory works, we must first understand the way in which data is organised. The smallest unit of data is the **bit**. A group of eight bits makes a **byte**, which often represents one character. A group of characters makes up a file, which may be an executable program or simply data. When dealing with memory, we generally concern ourselves only with characters and files. These are basic mechanisms which the operating system uses to store data.

Memory is also where the **operating system** resides. The operating system is a program which gives the user interactive control over the computer. It also controls the access to peripheral devices by user programs. We will discuss these aspects in more detail later.

When running a program (whether it is an operating system function or an application program under the control of the operating system) the computer loads the program and the data being manipulated into memory. It then transfers the data to, and from, the **input/output** (I/O) devices and secondary storage as required.



### Self-test

1.6 How many bits in a byte?

1.7 What is the main purpose of primary storage.




### Exercise

1.6 Describe what is held in memory during program execution.

1.7 Find out how much main memory your computer has. (If you are unfamiliar with computers at this stage, one way to do this is to look at the messages displayed by the computer when it starts up.)

## Types of memory

There are basically two types of memory:

- **read-only memory** (ROM)
- **random access memory** (RAM).

ROM is pre-loaded with programs or data and cannot be altered by the computer. It has the important feature of being **non-volatile**; that is, when power to the computer is turned off, ROM does not lose its contents. Its major use in modern computer systems is to hold a small program which the computer executes when it is first turned on. This program tells the computer:

- how to read the keyboard
- how to use the VDU
- how the hard and floppy disks are configured
- where (on which disk) to look for the operating system
- how to load the operating system in order to start up the computer.

These aspects of the computer and operating systems are described in detail in later modules.

Random access memory can be both written to and read from. It is also known as **read/write memory** for this reason, although it is commonly referred to (somewhat incorrectly) as RAM.

The computer uses RAM to temporarily hold programs and data. RAM is **volatile**, which means that when power to the computer is turned off, the RAM contents are lost. The operating system uses and manages RAM for the following purposes:

- input storage areas where data is held temporarily until processing takes place

- working storage areas where intermediate results of processing are held during processing
- output storage areas where finished results are kept before being released to the screen, printer or secondary storage
- program storage areas which hold the step-by-step processing instructions.



### Self-test

1.8 Define the term non-volatile memory.




### Exercise

1.8 Describe the differences between RAM and ROM and list the uses of these two types of memory.

1.9 If you have access to an old or unused computer, locate the RAM and ROM chips on the motherboard. The ROM is usually a fairly large chip and has information about the BIOS contained with in it stamped on a label attached to its top. RAM may be in two forms;

- a row of chips plugged directly into the motherboard which is known as the cache, and
- small plug-in circuit boards, or single in-line modules (SIMS), which usually have three or nine chips mounted on them. This is the dynamic RAM (DRAM) which is what we talk about when we say 'a computer has four megabytes of memory' and is the memory which is used for the purposes descibed above.

How do the number of RAM chips relate to the total RAM that is installed on your computer? (Be careful if you are looking inside a working machine. Make sure the power is disconnected and that you do not damage any components with static electricity.)

## SECONDARY STORAGE

You can see that the contents of primary storage are either:

- unchangeable and non-volatile (ROM), or
- changeable, but lost when the power is turned off (RAM).

Often we want to create or change files of information and keep them for later use. This is where we use secondary storage. Secondary storage devices can store large amounts of data and retain the data even when the computer is turned off. They can be read from, and written to, many times. The data can be erased or changed whenever we like. An exception to this is **CD-ROM** (compact disk — read-only

memory), which is a read-only data storage device. In modern computer systems, most of our data and programs (including the operating system) are kept in secondary storage. They are only loaded into memory (RAM) when needed. In the case of the operating system, the loading is done at system start-up time, or **boot** time.

The most common forms of secondary storage are:

- hard disks
- floppy disks
- magnetic tape
- CD-ROM.



### Self-test

1.9 What is the main purpose of secondary storage?




### Exercise

1.10 List all the secondary storage devices installed on your computer along with their storage capacities in megabytes.

## THE CENTRAL PROCESSING UNIT

The **central processing unit**, or CPU, is the heart of the computer. The CPU:

- performs all the manipulations on data
- controls the movement of data within the computer system.

The CPU consists of:

- registers for temporary storage of data and addresses
- a control unit which controls the CPUs internal processes
- an arithmetic logic unit, or ALU, which performs all the data manipulations required.

When a CPU is built on one silicon chip, it is known as a microprocessor. The original definition for a microcomputer was 'a computer which uses a microprocessor as its CPU'.



### Self-test

1.10 What are the three main elements of a CPU?




### Exercise

1.11 If you have access to an old computer, locate the CPU chip on the motherboard. Can you determine what type of processor it is?

# THE OPERATING SYSTEM

The **operating system** (OS) is the software which controls the *computer hardware* and how it interfaces with the *computer user*.

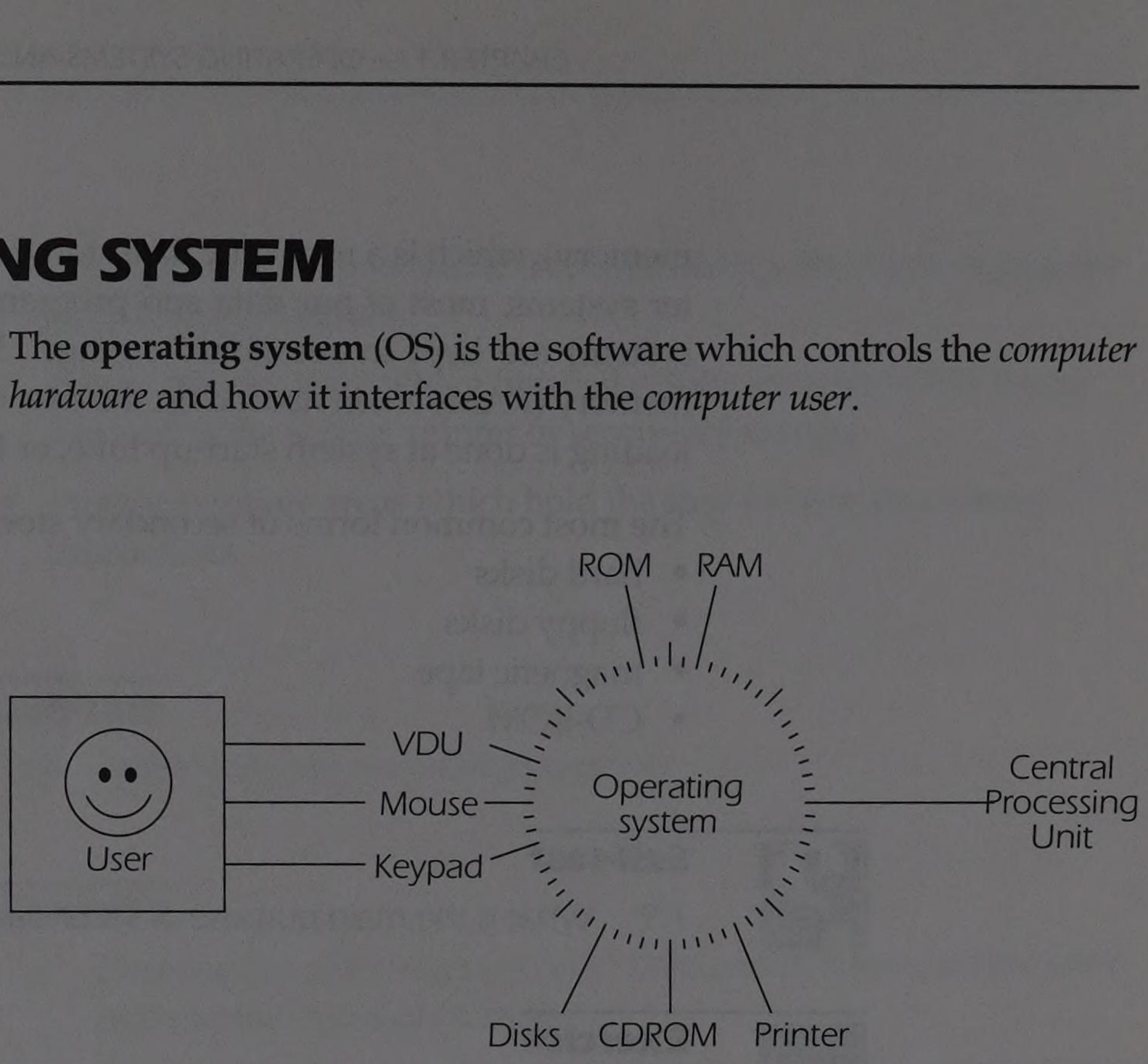



Figure 1.4 — The computer-user interface

The operating system:

- controls how devices interact with the CPU
- allows data to be transferred between memory and secondary storage devices
- allows data to be copied from a file in primary or secondary storage to output devices such as VDUs, plotters and printers
- accepts commands from the user, through a keyboard or a mouse, and processes them.

Operating systems, of whatever type, provide the following general facilities:

- organisation of storage for data and programs in files and directories, on secondary storage devices
- file manipulation — create, display, print, copy, rename
- a communication interface between the user and the computer.

There are literally hundreds of different types of operating systems in existence today. They are used on thousands of different makes and models of computers. In the world of PCs, a number of operating systems have become very popular. They can be classified as:

- single user — only one user at a time can use the computer
- multi-user — many users can use the same computer at the same time
- network — single user and/or multi-user operating systems communicate over a common medium. These are not operating systems in the strict sense, but they do offer typical operating system facilities, over a network.

Some examples of common operating systems today are:

- single user — MS-DOS, Windows, PC-DOS
- multi-user — Unix, Pick, VMS
- network — Novell, Appletalk, Banyan Vines.

Operating systems are currently available in either of two main interface types:

- command line interface — MS-DOS, Unix (shell), PC-DOS, VMS, Pick
- graphical user interface — Windows, X-Windows (X-11), Motif, Open Look, Apple Macintosh (Mac).

This module introduces you to **Microsoft DOS** (MS-DOS, or simply DOS). This is a single-user operating system, using a command line interface. It has been the most common operating system for PCs and is likely to remain so in the short term. This position is changing rapidly with **Microsoft Windows** now becoming very widespread in the PC area. Windows is the most popular GUI-based operating system.

The remainder of this module covers the various facilities that DOS offers and explains how you can use these to create files and run applications on your PC.



### Self-test

1.11 How important is the operating system to the computer?




### Exercise

1.12 List three types of command line-based operating systems and briefly describe what type of computers they are generally found on (e.g. PCs, business computers).

1.13 List three types of graphical user interfaces and what type of computers they are generally found on.

## CHAPTER SUMMARY

In this section we described the overall structure of a computer. We looked at the three major categories of computers — mainframes, mini-computers and microcomputers. We also examined the four basic elements of the computer's hardware and described a number of the more common peripheral devices.

Most importantly, we introduced the concept of an operating system; in particular the DOS single-user operating system. We looked at its role in controlling the computer's hardware and providing an interface between the user and the computer.



## Review questions

1.1 What are the four basic components of a computer system?

1.2 Name the three categories of computer.

1.3 What kind of input device has a ball that is moved by your fingers?

1.4 What does VDU stand for?

1.5 Describe the difference between ROM and RAM.

1.6 In a CPU, what does the ALU do?

1.7 What is the purpose of an operating system?

1.8 What is the most common command line-based operating system for PCs today?

1.9 What is the most common GUI-based operating system for PCs today?



## Review exercises

1.1 List five applications where you might find a computer.

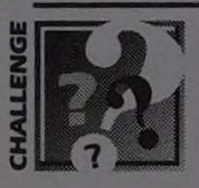
1.2 Give a definition of a microcomputer.

1.3 Describe how memory is organised.

1.4 List three uses of RAM.

1.5 Describe how a PC starts up and loads the operating system.

1.6 List and briefly describes four main functions of an operating system.




## Challenge problems

1.1 If you required a computer that had to perform thousands of very fast calculations and was to be shared by many people, what type of computer would you choose and why?

1.2 Why would a trackball be better than a mouse as an operating system interface for a laptop computer?

# CHAPTER 2

# OPERATING SYSTEMS AND MS-DOS

This chapter introduces you to the components of an operating system, with particular reference to the Microsoft Disk Operating System (MS-DOS). We will examine what these components are, what they do and how they work together.

At the end of this chapter you will be able to:

- define the term operating system
- describe the two major functions of an operating system
- describe what the BIOS does in MS-DOS
- define the function of COMMAND.COM in MS-DOS
- define the term utilities
- describe the steps involved in the boot process
- list the major features of MS-DOS
- describe how application programs interact with the operating system.

# OPERATING SYSTEM STRUCTURE

As we saw in Chapter 1, the **operating system** coordinates the activity between you, the user, and the microcomputer. Through the operating system you tell the computer to:

- run your applications
- store and retrieve your data
- perform any of several other tasks.

The operating system has two major functions:

- controlling the computer's operations
- file management.

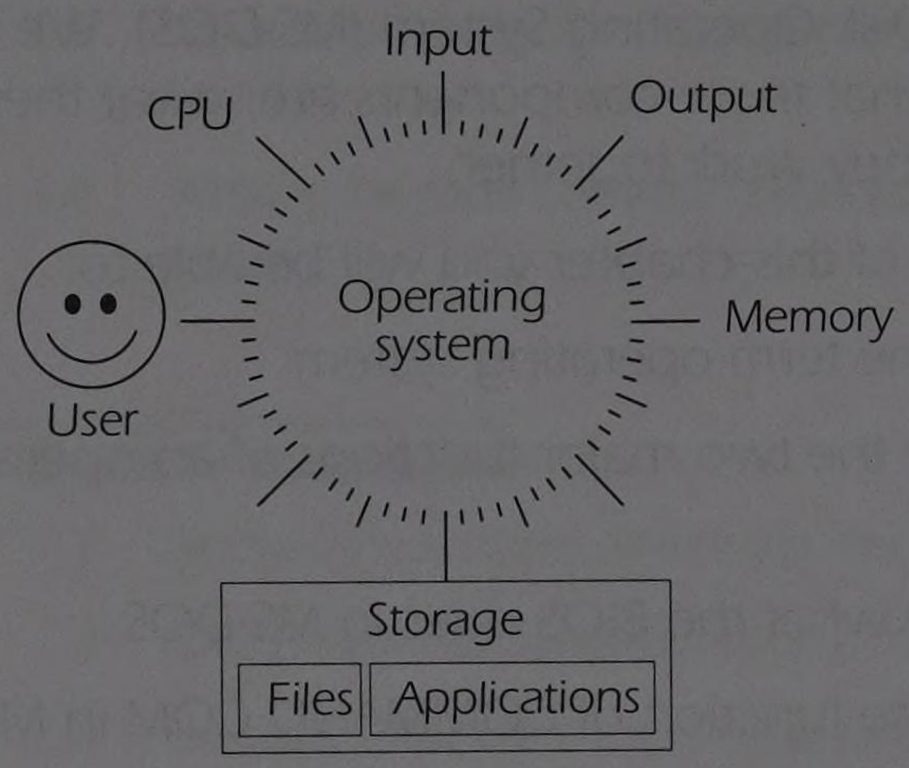



Figure 2.1 — The operating system provides an interface between the user and the computer

In practice, file management is linked so closely with these control functions that it is difficult to separate the two processes.

The control of computer operations may be subdivided into several parts.

## INPUT-OUTPUT MANAGEMENT

**Input-output management** is the process of coordinating the computer's data flow; that is, CPU's communication with the input and output devices and secondary storage. This includes:

- fetching data input from the keyboard
- sending display data to VDU terminals
- controlling read and write operations to disk drives.

In MS-DOS this is handled by the **BIOS,** or basic input output system, which is stored in ROM.




### Exercise

2.1 Describe the main functions of the BIOS in MS-DOS.

2.2 Next time you start the computer, list the devices that are defined in the BIOS. They should appear on the screen as part of the startup process.

## COMMAND PROCESSING

**Command processing** is the part of the operating system which:

- interprets commands entered via the keyboard or other input devices
- implements the hardware operations necessary to comply with these commands.

Performing these tasks often involves controlling the input-output manager (BIOS). In MS-DOS, this function is carried out by the COMMAND.COM.

## UTILITIES

**Utilities** are relatively small programs which are part of the operating system. They perform such tasks as:

- formatting disks
- copying files
- listing directory contents and tables of directories
- checking for viruses
- providing on-line help.

Most file management tasks fall into this category.

MS-DOS operating system commands fall into two categories:

- **internal commands** — which are built into COMMAND.COM
- **external commands** — which are utilities stored in the DOS directory of the boot disk drive (the one from which the operating system was loaded).

We will discuss this in more detail in the next section.



### Self-test

2.1 What are the two major functions of an operating system?

2.2 What is the difference between an internal and an external command?

# STARTING THE COMPUTER

When power is first applied to a computer system, it knows nothing about the input and output devices attached to it or what it is supposed to do. To get from here to a point where the computer is interacting with the user via the operating system is known as the **bootstrap** or **boot** process.

The term **cold boot** is used where we turn on the power to the computer to start it.

The alternative to this is called a **warm boot**. Here we re-boot the system after the computer has already been on. There is a slight functional difference between these two processes, but the details are beyond the scope of this book. The boot process will be slightly different depending on the operating system used. However, most operating systems follow the general pattern of the MS-DOS boot process described here.

## THE BOOT PROCESS

When the power is first turned on, the CPU reads a specific memory location to find the location of the BIOS in ROM. The startup portion of the BIOS instructs the computer to run diagnostic tests, including checking the computer's memory, the disk and keyboard.

## SYSTEM PROGRAMS

Once the diagnostic tests are finished, the computer usually, by default, tries to read drive A. Drive A is the first of the two floppy drives on the computer. If a floppy disk containing the DOS operating system has been inserted in drive A, the two system programs — IO.SYS and MSDOS.SYS — are loaded and executed. If there is no disk in the first drive, or the information is not on the disk, the computer will look for the DOS operating system on the hard disk. The disk on which DOS is ultimately found is known as the **system disk** or **boot disk.**

In more modern BIOSs the order in which disks are read is configurable, so your computer may go directly to drive C (the hard disk) to look for DOS.

## COMMAND.COM

After the two system programs are executed, the computer looks on the disk for a program called COMMAND.COM. This program is then executed. Executing COMMAND.COM causes a copy of the operating system kernel (but not all of the utilities) to be loaded into RAM where it remains until the power is turned off.

## AUTO EXECUTION OF FILES

The computer now looks for two files:

- CONFIG.SYS
- AUTOEXEC.BAT.

If the computer successfully finds these files, it will execute the commands contained in them. We can use these files to customise the system to suit our particular requirements.

CONFIG.SYS is mainly used to add programs (drivers) to control memory or peripherals which the computer's BIOS does not know about, (e.g., CD-ROM drives) or to give better control over devices which the BIOS recognises. For example, the **smartdrive** driver (SMARTDRV.EXE) speeds up the operating system's access to disk drives.

ATOEXEC.BAT is more concerned with customising operating system features, for example, the way the prompt apprears. It can also be used to start application programs automatically each time we turn on the computer.

If there are no CONFIG.SYS or AUTOEXEC.BAT files, the computer will ask you to set the date and time. PCs have a real-time clock which keeps track of the time even when the computer is switched off. This is to ensure that files can be correctly dated when they are saved. Dating our files is useful. If we have several copies of the same document, we can always find out which file contains the latest update by looking at the date on the file.

### PROMPT

After the computer has gone through the steps listed above, the prompt will appear. The prompt tells us which is the active drive. If we had the DOS disk in drive A, the prompt will be:

**A>** or something similar, depending on how the prompt has been set up.

If we booted off the hard disk, the prompt would be:

**C>** or something similar, again depending on how the prompt has been set up.



### Self-test

2.3 What is the first thing the BIOS does when it starts during the boot process?

2.4 What is the difference between a warm boot and a cold boot?




### Exercise

2.3 Describe what happens during the boot process if there is no system disk in drive A.

2.4 Briefly describe the role of each of the following files in the boot process:

- CONFIG.SYS
- AUTOEXEC.BAT

## PRIMARY FEATURES OF MS-DOS

MS-DOS provides the user with a **command line interface** method of controlling the computer system. This means that we are able to execute application programs and activate internal or external commands by typing commands at the DOS prompt. The commands and utilities available fall into the following general areas:

- memory configuration and management
- file management
- application program execution
- printing
- text file creation
- peripheral and hardware control.

MS-DOS also has a number of advanced features which include:

- automatic optimisation of memory usage
- recovery of accidentally deleted files
- anti-virus utilities
- data compression for disk storage
- automatic backup facilities.

In later chapters we will look from a practical perspective at the key elements and concepts involved with the basic features of MS-DOS.



**Self-test**

2.5 List six main features of MS-DOS.

## OPERATING SYSTEMS AND APPLICATION PROGRAMS

Because the operating system controls the computer hardware, application programs 'call' the relevant parts of the operating system whenever they need to communicate with the computer hardware. For example, the operating system obtains input from the keyboard and passes it to the application software. The application software passes output to the operating system which in turn passes it on to a printer to be printed.

Because different operating systems communicate with the hardware in different ways, the application software must be designed to work with each particular operating system. For this reason different versions of application programs are available to work with different operating systems. For example, you would need a different version of Microsoft Word to work with DOS than you would with the Macintosh.

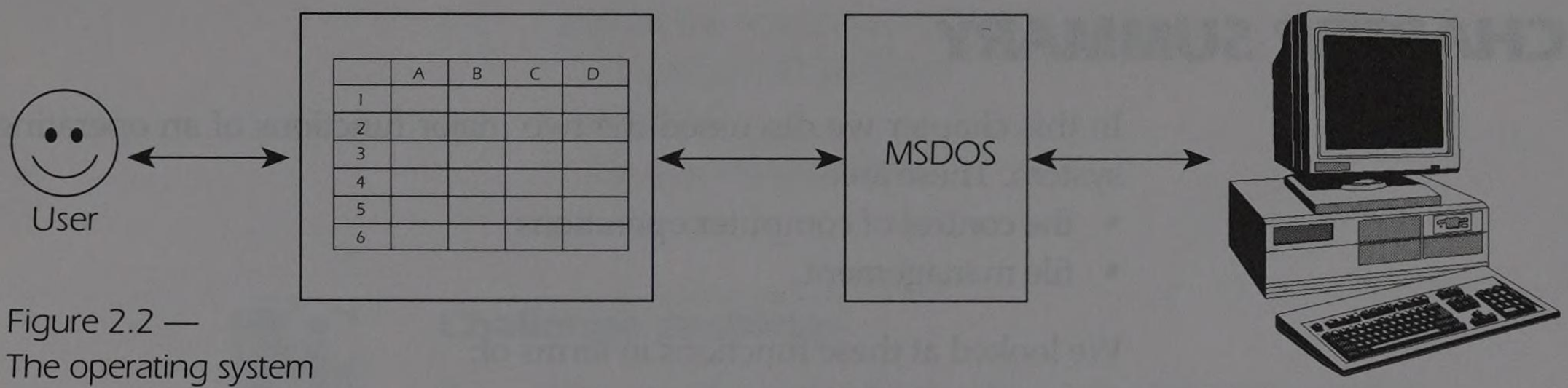



Figure 2.2 — The operating system controls the execution of application programs such as word processors and spreadsheets

## OPERATING SYSTEM VERSIONS

As new hardware is released for particular computers, the BIOSs and operating systems are revised to take advantage of the new hardware and to fix known 'bugs' or faults in the system. These revisions are released as different **versions**. For instance MS-DOS has been released in versions 1.1, 2.1, 3.0-3, 4.0, 5.0 and 6.0-6.22.

Generally, revisions are made in such a way that application written to work with older versions of the operating system will also work with new versions. Unfortunately, applications developed to run with new versions of the operating system often take advantage of additional features not available in older versions.

In these cases, new versions of these application packages cannot run under an earlier operating system. The operating system version which can be used on a system is usually stated in the application software documentation.

A typical statement would read something like this: 'Wordstar version 4.4 will run under MS DOS versions 3.1 or later'.



### Self-test

2.6 How does an application program communicate with the computer hardware?




### Exercise

2.5 Explain why Word for DOS would not run on a Macintosh computer.

2.6 List the main application programs running under DOS on your computer.

2.7 Find out what version of DOS is running on your computer.

# CHAPTER SUMMARY

In this chapter we discussed the two major functions of an operating system. These are:

- the control of computer operations
- file management.

We looked at these functions in terms of:

- input-output management
- command processing
- utilities.

We defined internal and external commands and described the boot process. We have also briefly described the primary features of MS-DOS which will be covered in detail in later chapters. Finally, we looked at the way in which application programs interact with the operating system.



## Review questions

2.1 What are the two main operating system functions?

2.2 What are the three major components of computer operations control?

2.3 List three functions performed by utilities.

2.4 What does the boot process do?

2.5 What is a system disk?

2.6 What is a cold boot?

2.7 List four primary and four advanced features of MS-DOS.



## Review exercises

2.1 Describe the purpose of the COMMAND.COM in MS-DOS.

2.2 Briefly describe the difference between internal and external commands.

2.3 List the steps involved in the boot process.

2.4 Give two reasons why it is important to know which version of an operating system is running on the computer.

2.5 Describe the relationship between:

- the operating system
- an application program
- the user.

2.6 Describe the interaction between:
- an application program
- the BIOS
- the operating system.




## Challenge problems

2.1 Why would a piece of software have a statement such as: 'Word for DOS version 6.0 requires DOS version 5.0 or later' in its documentation?

2.2 Why would a computer generally boot from a hard disk rather than a floppy disk?
(Hint: think about what you learnt in Chapter 1 about floppy and hard disks.)

# CHAPTER 3

# DIRECTORY AND FILE MANAGEMENT

File management is one of the primary functions of the operating system. In this chapter we will examine the features and characteristics of MS-DOS in this role.

At the end of this chapter you will be able to:

- identify the characteristics of a legal file name
- describe the process of reading files from a disk drive
- describe the process of saving files to a disk drive
- define the terms default drive and active drive
- describe a typical MS-DOS directory structure
- set up directory paths
- describe the use of wildcards
- define the term current directory.

You will also learn how to use the following file management commands:

- copy
- delete
- undelete
- rename
- type
- print
- dir
- chdir
- mkdir
- rmdir
- tree
- path.

## FILE NAMES

Data and programs are stored in **files** on the computer system. Each file has its own unique name to identify it. Some files are named by the operating system or application packages. However, we generally assign names to data files and programs which we create ourselves. We should give our files meaningful names — names that reflect the content of the file in question. This will help identify the files more easily.

When we assign names to our files, we must check the naming convention used by the operating system. The convention for naming files can vary between different operating systems. Some operating systems allow only short filenames. Early systems, for example, used six characters for the filename plus a three character extension. Other systems allow much longer names.

MS-DOS, like all operating systems, has its own rules on how to assign filenames. Filenames in MS-DOS are made up of:

- the **name** of the file, which may be one to eight characters long, and
- an **extension** of zero to three characters.

There are certain characters which cannot be used for MS-DOS file names. MS-DOS interprets these 'special' characters in specific ways. For instance, the period (.) is used as a filename **separator** — it separates the filename from the extension. There are a range of suitable characters for use as MS-DOS filenames. They are listed in Table 3.1.

Table 3.1 — The set of characters used in filenames and extensions

| Characters | Description |
|---|---|
| A — Z | Upper case alphabetic characters |
| a — z | Lower case alphabetic characters |
| 0 — 9 | Digits |
| _ | Underscore |
| ^ | Caret |
| $ | Dollar |
| ~ | Tilde |
| ! | Exclamation |
| # | Number sign |
| % | Per cent |
| & | Ampersand |
| - | Hyphen |
| { } | Braces |
| ( ) | Parentheses |
| @ | 'At' sign |
| ' | Apostrophe |
| ` | Grave accent |

## VALID AND INVALID FILENAMES

When naming files we need to:

- use only the characters from Table 3.1
- have a minimum of one character and a maximum of eight characters in the filename
- have no more than three characters in the extension.

Some examples of MS-DOS file names are shown below.

Table 3.2 — Typical valid and invalid filenames

| Valid filenames | Invalid filenames |
|---|---|
| a.exe | a.exet |
| Boss.f | arapahoes.elm |
| wdstar.com | ala carte |
| &mmate | yes:no.doc |
| fred(s).txt | aux.. |
| 12345678.123 | A\B\C.D |
| 12345678 | A B C.D |
| x.y | 123456789.123 |
| AuToExEc.BaT | 123.1234 |
| AUTOEXEC.BAT | .123 |

### Important point

Remember that MS-DOS filenames are **case insensitive**, so it does not matter if you mix upper and lower case characters. This is not true for all operating systems however.

In addition to the rules outlined above, the following filenames are reserved for specific purposes by MS-DOS, so they are not available for general use:

- CLOCK$
- CON
- AUX
- COM*n* (where n = 1-4)
- LPT*n* (where n = 1-3)
- NUL
- PRN

These will be discussed in more detail in later modules.



### Self-test

3.1 What is the maximum and minimum length for a filename?

3.2 What is the maximum and minimum length for the extension?




### Exercise

3.1 Give reasons why each of the filenames in the Invalid filenames list in Table 3.2 are invalid.

## STANDARD FILENAME EXTENSIONS

Some filename extensions have special meanings, either to the operating system (MS-DOS) or to the application packages running in that operating system. The extension is used to help identify file types. Although the operating system (or the application package) does not always require us to follow these conventions, using standard extensions will help us to identify the type and use of our files. This is especially important when files are shared between different people or applications.

Some examples of these standard extensions are shown in Table 3.3. This is not a complete list as many of the extensions are application dependent, but it will give you an indication of the more common types.

Table 3.3 — Some common file extensions

| Extension | Typical meaning |
|---|---|
| .EXE, .COM | Operating system program or **executable** file |
| .SYS | **System** file which contains information about the computer's hardware |
| .BAT | **Batch** file — which contains a series of operating system commands that can be executed in one 'batch' |
| .TXT | **Text** file which contains standard printable characters with no special formatting |
| .DOC | **Document** file usually created by a word processor |
| .BAS | A **BASIC** language program |
| .INI | An **initialisation** file for an operating system or an application |
| .BAK | A **backup** file containing a copy of the contents of some other file |
| .HLP | A file containing **help** information |
| .DAT | A **data** file |
| .PCX, .BMP, .GIF | Three examples of extensions of files containing **graphics** data |

## The importance of extensions

Batch (.BAT) files are an important part of MS-DOS. These are files which contain a sequence of commands that can be executed at some later stage. These commands are written in the same way as we would type them in at the keyboard. The operating system recognises whether a file is a batch file by the .BAT extension. It will execute the associated commands in this file just as if we were typing these commands into the system ourselves. This is a very convenient mechanism for executing often repeated procedures. We will look at batch files in greater detail in another module in this series.

Application programs will often add extensions of their own to filenames specified by the user. For example, the word processor will add the .DOC extension to indicate a document file.

There are many valid combinations that can be used when assigning filenames. However, you should make the filename descriptive to reflect what the file contains, and you should use a three character extension following the conventions described above. These guidelines will result in filenames that make file identification an easy task.

The examples below demonstrate the importance of using standard extensions in operating system and application package files.



### Self-test

3.3 What would you expect the following files to be?

CHESS.EXE

RESUME.DOC

PAYROLL.INI

MEMO9410.DOC

DIARY.TXT

COUNTRY.SYS

FLGHTSIM.EXE

SALES94.DAT

GAMES.HLP




### Exercise

3.2 Create valid filenames for the following file descriptions:

- an application program that calculates salaries
- a document which is written with a word processor that describes Russia
- a file containing a sequence of executable commands which creates an on-screen menu for the user
- sales data for the financial year 1994/1995.

# STORING FILES ON A DISK

Files are created by the operating system and user applications. These files are stored as permanent data on secondary storage devices — generally hard disks and diskettes.

When a file is created by a word processor, for example, the computer holds it in its internal memory (RAM). At this stage there is no permanent record of the data within the file. A power failure or other computer problem at this point could cause the data to be lost. To ensure that you do not lose your data, you need to **save** your file.

When you save a file, the computer makes a copy of that file onto a disk. You are able to work on your file again when you need to, by reading the file back from the disk into memory. The application you are running can then access it. The process of reading the file back in is called **retrieving** or **opening** and is usually done from within the application.

## DISKS AND DISKETTES

To understand how MS-DOS saves or stores files onto a disk, we need to briefly look at how a disk holds data.

A disk is a circular piece of non-magnetic material coated on both sides by a thin layer of magnetic particles. Hard disks are made of aluminium, while floppy disks are made of a flexible material called mylar. Data is stored on these disks as tiny north and south magnetised areas where each magnetised area represents a **one** (1) or a **zero** (0), a **binary digit**.

**Binary data** is made up of a series of binary digits, with a one having the opposite polarity to a zero. For instance, a north magnetised area may represent a binary 1 while a south magnetisation may represent a binary 0. Numbers, characters and other data are stored as combinations of these binary digits or **bits**. For example, the number 42 can be stored as the following bit sequence:

```
00101010
```

These sequences of ones and zeros form the basis of the data in our files. These sequences are stored as groups of eight bits; each group of eight bits forming **one byte**. The bytes of data are stored in concentric rings or **tracks** on the disk. This is depicted in Figure 3.1.

Each track is divided up into **sectors**. Our files are stored in groups of sectors. However, a file does not necessarily have to be stored in consecutive sectors. The actual location of the data on the disk depends upon which sectors are free at the time that our files are written to the disk.

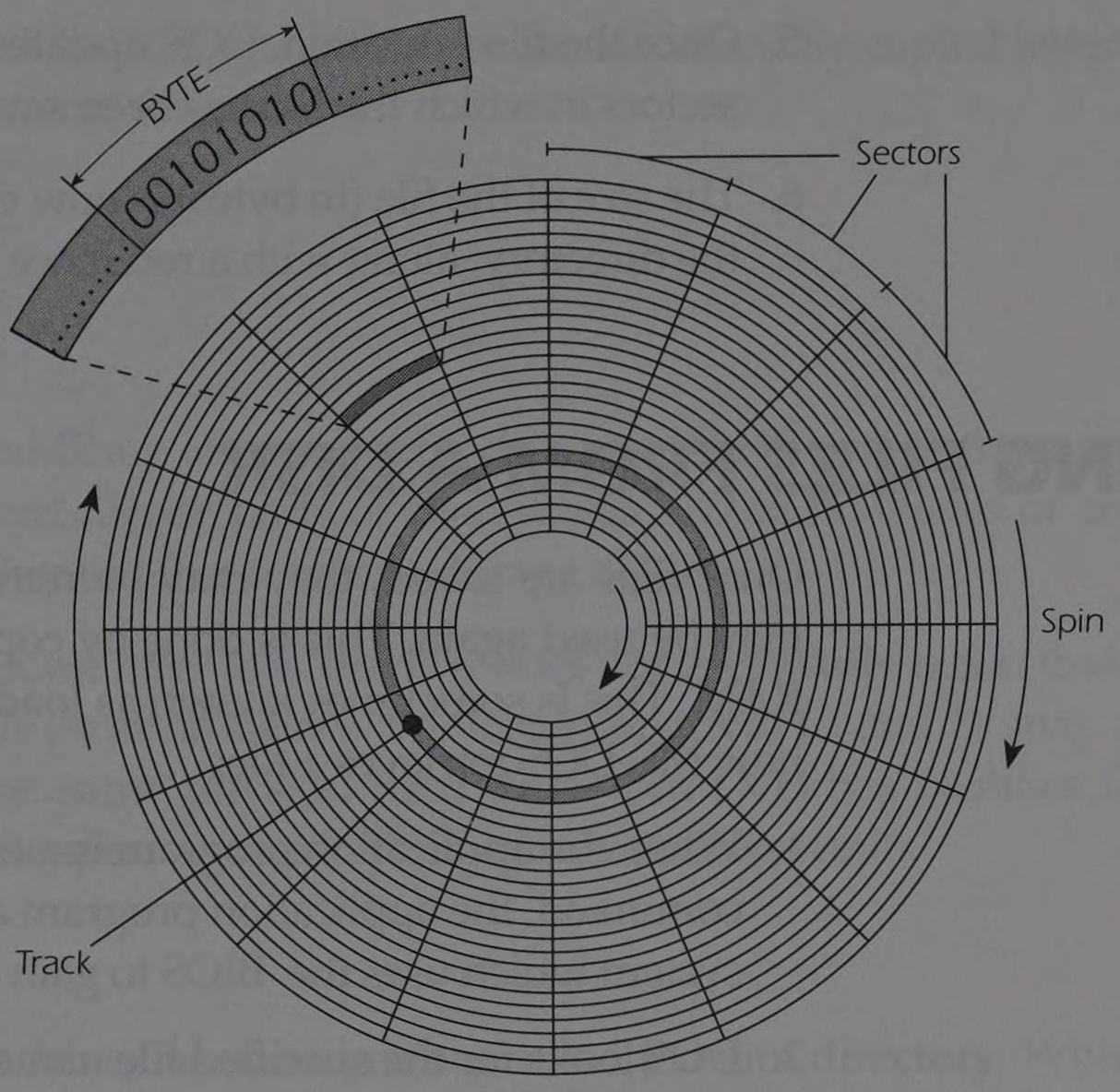



Figure 3.1 — Storing binary data on a disk track

## SAVING FILES

Using this basic understanding of how disks function, we will now examine how MS-DOS saves a file. This is done in a series of steps.

1. When we issue a **save** command from within an application program such as a word processor or spreadsheet, the application program accesses the relevant part of the operating system that controls the disk drive selected via the BIOS. If a floppy drive is selected, the BIOS orders the disk drive mechanism to physically clamp the disk and spin it at 360 revolutions per minute (rpm). This prepares the disk so it is ready to have the file written to it. On the other hand if we wish to write to a hard disk, it will be already spinning, so the BIOS just prepares to perform the write.

2. The operating system stores:
   - the file name (including the extension)
   - the current date and time

   in an area on the disk called the **directory**. The directory is like an index in a book. The operating system uses the directory to keep track of what is stored on the disk.

3. Next, the operating system reads the **file allocation table** (FAT) on the disk. This contains information about which sectors are used or unused as well as which sectors are used for which file. It looks for an unused sector as a starting point for writing the file.

4. The disk drive's **read/write head** moves to the required sector and begins writing a copy of the file. If the disk is relatively empty, the file will be stored in consecutive sectors. When a disk becomes nearly full, this is generally not possible, so the file may be scattered across different sectors all over the disk.

5. Once the file is written, DOS updates the FAT to include the sectors in which the file has been saved.
6. The size of the file (in bytes) is now entered next to the filename in the directory, along with a reference to the FAT.

## READING FILES FROM A DISK

Once files are saved, they may be retrieved at any time so that they may be used again. This is done by copying the data from the disk to RAM. This is sometimes known as **loading** or **opening** the file. This is how MS-DOS retrieves a file:

1. When the application program issues an open or retrieve command, the application program again accesses the operating system which uses the BIOS to gain control of the disk drive.
2. DOS looks for the specified file name in the directory of the disk and references the FAT.
3. Next, DOS reads the FAT to get information about which sectors the file is stored in.
4. The disk drive's read/write head moves to the indicated tracks and sectors on the disk and begins retrieving the file.

Later we will examine commands like:

- COPY
- DEL
- TYPE
- RENAME.

These commands (and others) use either or both of the algorithms just described for reading and writing files.

## THE MS-DOS DIRECTORY STRUCTURE

If we simply kept all our files on disks without any form of organisation, it would soon become very difficult to find anything. This is just the same as keeping all the mail we received for a year (including junk mail) in one big box. If we then tried to find the letter from great-aunt Giacinta, which arrived in April and announced an inheritance, we would not be able to find it! To help overcome this problem, MS-DOS provides us with a system of **drives** and **directories** to group related files in a logical and systematic manner.

### DISK DRIVES

A **disk drive** is for computer files what a filing cabinet is for documents. It allows a large number of related data to be stored together.

In an MS-DOS system drives are identified by capital letters followed by a colon, for example:

- A:
- B:
- C:
- L:

A: and B: are always floppy drives, while C: and D: through to Z: may be hard drives, parts of hard drives, CD-ROM drives or any one of a number of other forms of storage devices.

When we refer to a 'drive' it does not necessarily mean that we have a whole physical device. For instance, drives C: and D: may actually be on the same physical hard drive, but DOS has facilities for making them appear as two logically *distinct* storage units.

## DIRECTORIES

Each drive has one directory called the **root directory**. Within the root directory there are usually other directories. Directories are like the drawers in a filing cabinet; they let us subdivide the information stored on the drive. Within each directory there may be sub-directories which are like suspension files in a filing cabinet. These sub-directories may have sub-directories of their own, and so on. These directories allow us to store data in a logical and organised way. This makes the data easier to find and work with.

As an example, Figure 3.2 shows the directory structure for the hard disk of a computer system with MS-DOS and *Windows* installed on it. Also installed on this disk is the *Word for Windows* word processor. This computer is used by 'Mark' who keeps data in sub-directories under his own name.

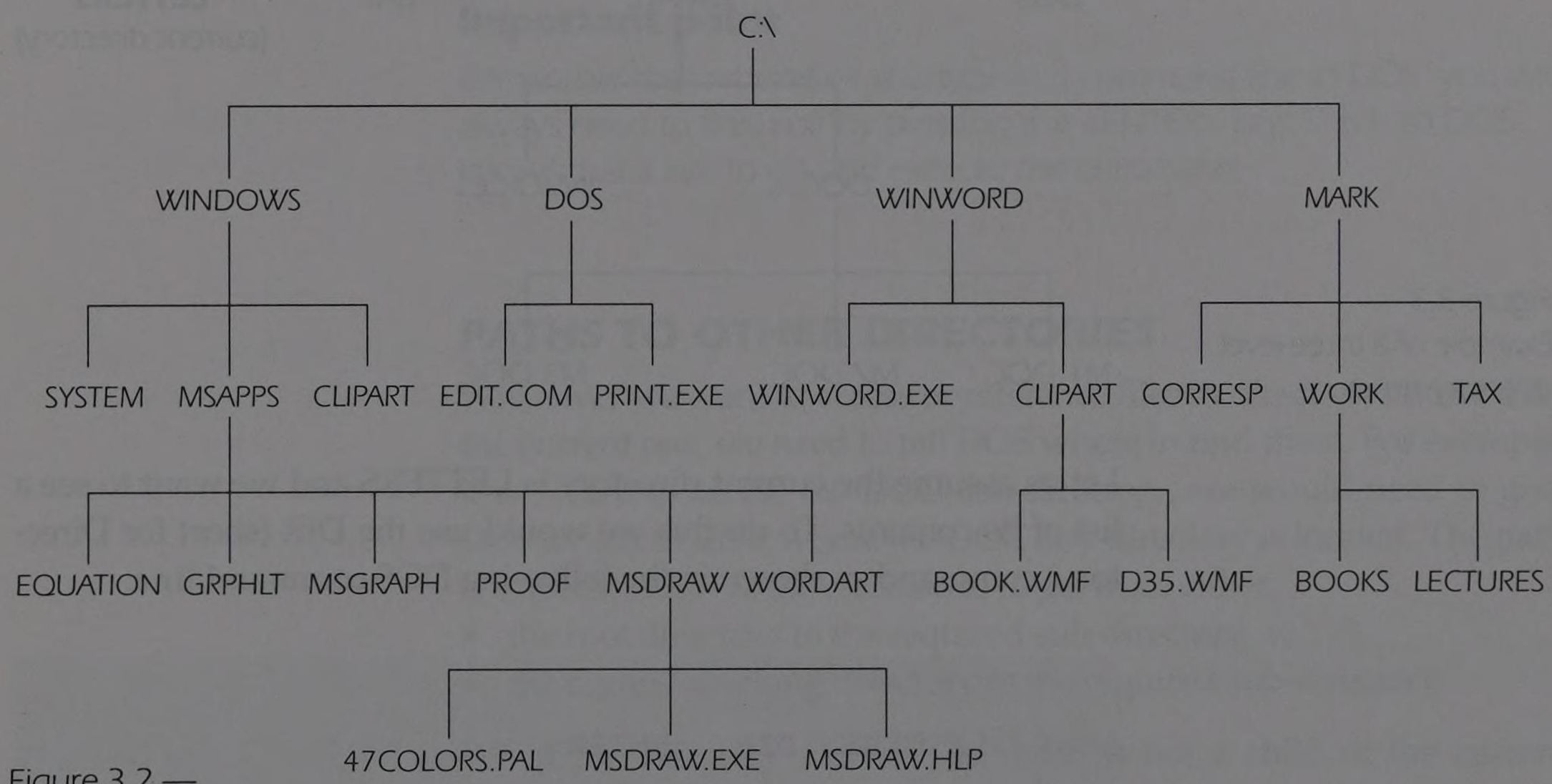



Figure 3.2 — Typical MS-DOS directory structure

We often describe the relationships between directories using the terms **parent** and **child.** When there is a directory/sub-directory relationship, the directory is referred to as the parent and the sub-directory is referred to as the child. In Figure 3.2, for example, the WINDOWS directory is the parent. The SYSTEM directory within that is the child.



**Self-test**

3.4 Refer to Figure 3.2. In the case of the MARK and WORK directories, which is the parent and which is the child?

3.5 What are all the child directories (or sub-directories) of the WINDOWS directory in Figure 3.2? Which of these child directories (if any) have sub-directories of their own?

## CURRENT DIRECTORY

When we are looking at files in a directory, it is like opening a drawer in a filing cabinet. The 'open drawer' is called the **current directory**. Just as we can see only what is in the open drawer of a filing cabinet, the computer can see only the files within the current directory.




- Look at the directory structure in Figure 3.3.

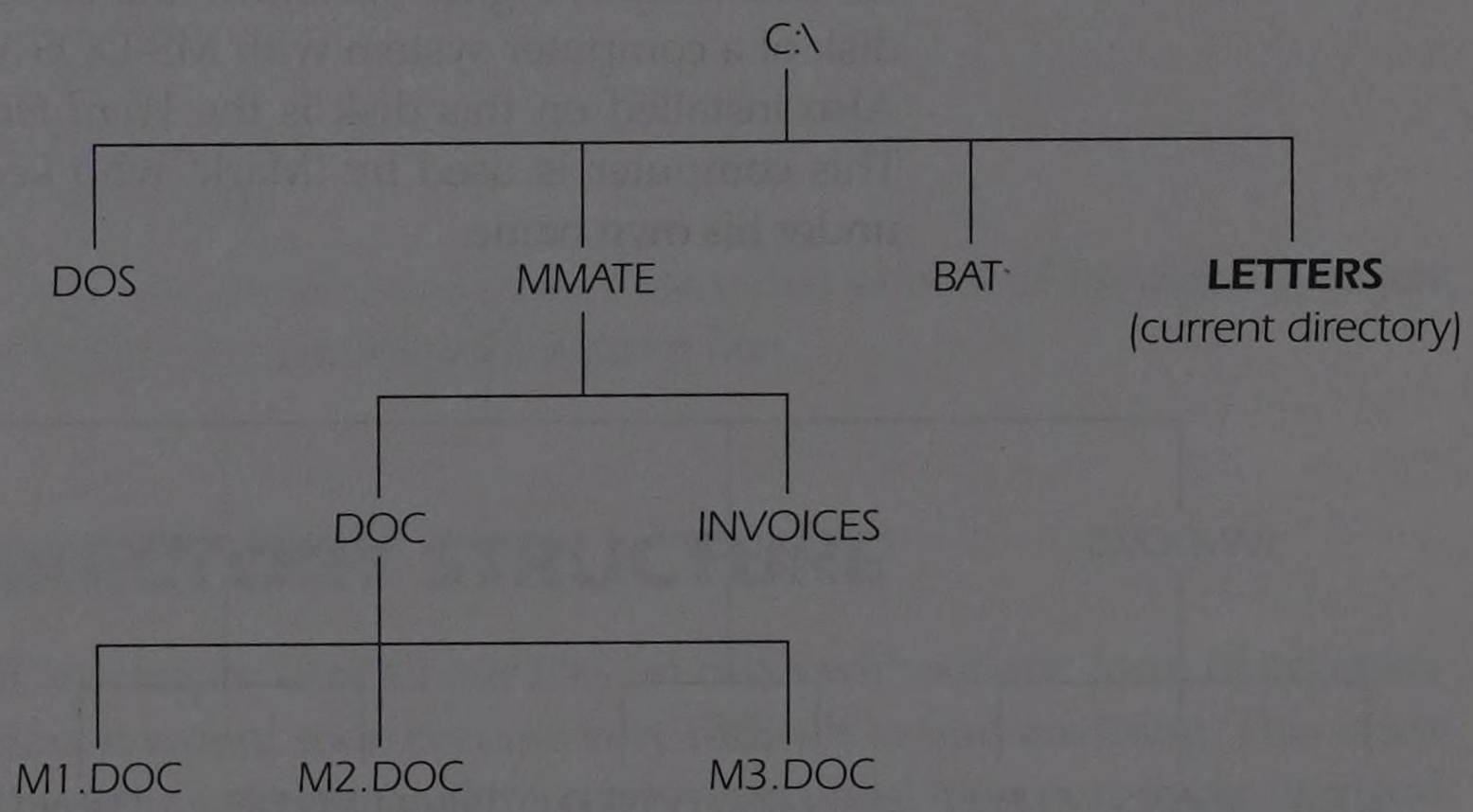



Figure 3.3 — Example of a three-level directory structure

Let us assume the current directory is LETTERS and we want to see a list of its contents. To do this we would use the DIR (short for Directory) command as shown in the following DOS **command line**.

```
C:\LETTERS> DIR  <ENTER>
   |        |     |      |
   1        2     3      4
```

We will discuss DOS command structures in more detail a little later. For the moment, note the following points about the DOS command line:

1. This is the disk drive we are currently working from. This is the **C:** drive hard disk.

2. This is the directory we are working from. This, together with (1) is called the **command line prompt** or, simply, the **DOS prompt**. The DOS prompt is displayed by the operating system — *you do not type it in!*

### Important point

The DOS prompt can take several forms, depending on how it is customised in AUTOEXEC.BAT, as mentioned previously. The form shown assumes that the command PROMPT $P$G was included in AUTOEXEC.BAT.

3. This is the **DIR**ectory command typed in by you, the user. We will use the convention of highlighting in **bold** typeface all commands typed in by the user. Information displayed by the computer will be shown in normal typeface.

4. When you have finished entering the command, press the <ENTER> key to tell DOS to go and execute it, i.e. to carry out the command instructions. The <ENTER> key is *always* required at the end of a DOS command line, so for the sake of clarity we will not include it in our command lines from here on.

### Important point

Remember that whenever you type in a command line in DOS, you will always need to finish off by pressing the <ENTER> key. This is so DOS knows that it has to go and execute the command.

## PATHS TO OTHER DIRECTORIES

Whenever we want to access or reference files in directories other than the current one, we need to tell DOS where to find them. For example, if we want to access the DOC sub-directory, we would need to give DOS a path to tell it where the DOC sub-directory is located. The **path** is a specification of the directories to get from either:

- the root directory to the required sub-directory, or
- the current working directory to the required sub-directory.

When the directory we wish to reach is not a child of the current directory, the simplest way to write the path is to specify the drive on which the directory is to be found, followed by the name of each

sub-directory from the root directory to the sub-directory we want to reach. The sub-directory names are separated by backward slashes (\). Using this method, the LETTERS and INVOICES sub-directories would be referenced as:

C:\LETTERS, and

C:\MMATE\INVOICES

respectively, regardless of the current directory.

### Example

To list all files in the DOC sub-directory, we combine the DIR command with the full path specification for the DOC sub-directory:

```
DIR C:\MMATE\DOC
```

When the directory we wish to reach is a child of the current directory, the path is simpler. In this case we only have to specify the path from the current directory down to the child. Let us assume the current directory is root and we want to see a list of contents of the DOC sub-directory again. We would still use the DIR command. The command would be

```
DIR MMATE\DOC
```




### Exercise

3.3 Using the example directory structure above and assuming the current directory is INVOICES, write the commands to list the contents of the following directories:

- BAT
- DOC

3.4 Assuming you are located in the MMATE sub-directory, write the command to list the contents of the following sub-directories:

- DOC
- INVOICES.

## DEFAULT AND ACTIVE DRIVES

Just as the computer needs to know the path to take when we want to access files in a directory, it also needs to know which disk drive and directory to look in to find MS-DOS commands and utilities.

DOS has a number of mechanisms to do this. The first is the concept of a **default drive**. This is the drive we initially booted up from. When we first boot up, the computer automatically knows that the default drive is the one we want to look on for files even if we do not specify a drive letter in the path.

The **active drive** is also the drive on which the computer will look for files, without having to specify a drive letter in the path. Initially, the default drive is the same as the active drive. However, the active drive may be changed by simply typing in the name of the new active drive. For example, to change from the A: drive to the C: drive simply type C:

Even though the terms *active drive* and *default drive* mean different things, they are often used somewhat interchangeably, so make sure you understand the context.

## THE PATH COMMAND

This command is very important in MS-DOS. It tells MS-DOS which directories to look in and in what order to look for programs to execute. This saves us having to type in the path to each program whenever we wish to run that program.

The PATH command is usually included in AUTOEXEC.BAT so that the command path is set up when the computer is started. The path may also be set by typing the command directly at the DOS prompt. The order in which directories are listed in the path is the order in which they are checked to see if the named command or program is there. The directories listed are separated by a semicolon (;). A typical PATH command might be something like this:

```
PATH C:\DOS;C:\WINDOWS;C:\WINWORD;D:\PROGS\MYAPPS
```

### Important point

*When MS-DOS is installed it will automatically insert the line PATH C:\DOS into the AUTOEXEC.BAT file. If this line is deleted, MS-DOS external commands will not work unless you specify the path to them.*

### Example

*If we wanted to run Windows we would need to execute the WIN command. After typing WIN on the command line, MS-DOS will search for the program WIN.EXE:*

- *firstly in the current directory*
- *then in C:\DOS*
- *then in C:\WINDOWS*

Because WIN.EXE exists in the C:\WINDOWS directory, DOS will not search any further. As soon as it finds the program it is looking for, DOS will execute it. If WIN.EXE was not present, MS-DOS would go on to search in:

- C:\WINWORD
- then D:\PROGS\MYAPPS

before reporting the message 'File not found'.

**Important point**

Remember that when you are trying to execute a program or application, DOS can only execute the program or application if it can find it. DOS can only find these in either of two ways:

- the command, program or application that you want executed is located in a directory specified by the PATH command, or
- you specify the full PATH to the command you want to execute.



**Self-test**

3.6 Assuming the following path is set up:

PATH C:\DOS;C:\WINDOWS;C:\WINWORD;D:\PROGS\MYAPPS

- what command lines do you need to type in order to execute the following commands:
  - the DOS command, FDISK?
  - an application program called PAYROLL.EXE located in MYAPPS?
  - a flight simulation game called HORNET located in a GAMES directory on C: drive?

3.7 Assume the PATH is the same as for 3.6 and there is a command called REPORT.COM located in D:\PROGS.

You then type in the following line at the DOS prompt:

- REPORT <ENTER>
- Will DOS execute your report command? Give reasons.




**Exercise**

3.5 Type in the PATH command at the DOS prompt. Write down exactly what is displayed on the screen and explain what it means.

# FILE MANAGEMENT

In order to manage our files successfully, we need to understand the facilities that MS-DOS provides us with via its:

- internal commands
- external commands and utilities.

### Important point

When referring to DOS commands, people often refer to them in various ways (as we have deliberately done).

You may see the DOS commands and utilities referred to as:

- commands
- programs
- utilities
- batch files
- executable files
- command files
- applications

and perhaps even other names!

Don't let this worry you. Generally, these naming conventions are not a real problem, as anything that is executed at the DOS prompt will be executed in a similar way.

This loose terminology is an unfortunate historical fact, and as a computer user it is best to understand this early on as it may otherwise cause considerable confusion.

## COMMAND SYNTAX

Each command has a correct **syntax**. The command syntax is the method for specifying:

- the correct name of the command
- what the command will act on
- how it will do it.

It is not our purpose to examine every option of every DOS command and describe the syntax of each of these in detail. Those details are best left to the:

- *MS-DOS Users Guide*
- *MS-DOS Reference Manual*
- MS-DOS on-line HELP which we will examine in a later chapter.

Our purpose here is to look at the most commonly used DOS commands and give simple, practical examples of their use. To do this effectively, let us first group file management commands into two major categories:

- file manipulation commands
- directory manipulation commands.

For the following examples we will use the sample directory structure shown in Figure 3.4. on the following page.

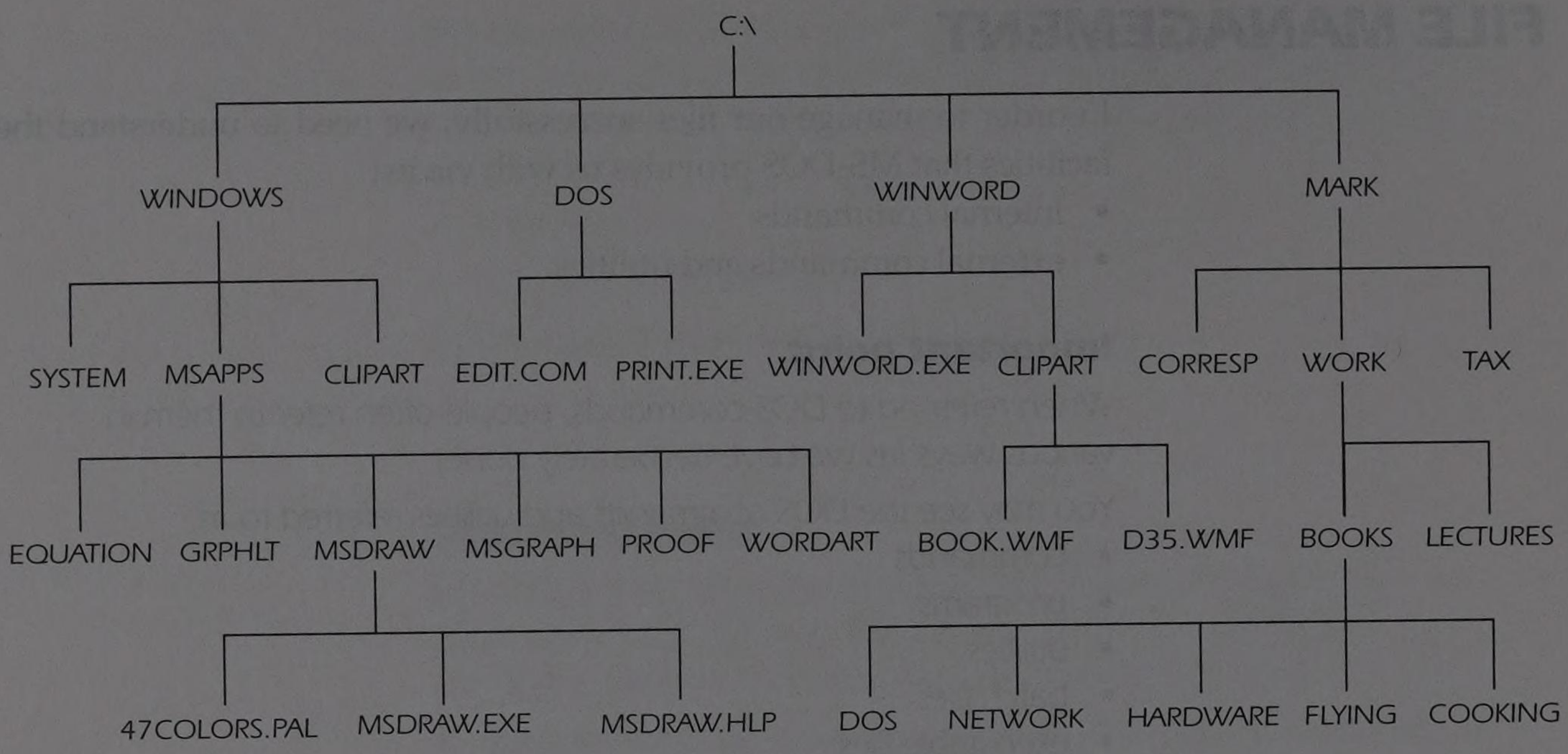



Figure 3.4 — Example of a five-level directory structure

# FILE MANIPULATION COMMANDS

The main commands which are used to manipulate files are:

- COPY — make copies of files
- DELETE (or DEL) — delete existing files
- UNDELETE — retrieve deleted files
- RENAME — change the name of a file
- TYPE — display the contents of a file on the screen
- PRINT — print a file on the printer
- DIR — get listings of files contained in the directories.

Let us examine each of these in turn.

## COPYING FILES

COPY copies one or more files to new location. This may be to:

- a different drive
- a different directory, and/or
- to a different name.

The basic syntax is:

```
COPY file1 file2
```

A more detailed syntax is:

```
COPY sourcepath\source_filename destinationpath\destination_filename
```

The rules for paths that we outlined earlier apply to the command syntax.

### Example

To copy the file AUTOEXEC.BAT stored in the root directory of C: drive to a file named AUTO.OLD in the DOS directory we could type in the following command:

```
COPY C:\AUTOEXEC.BAT C:\DOS\AUTO.OLD
```

If our current directory were already the C:\DOS directory, we could omit the *destination path* and shorten the command to:

```
COPY C:\AUTOEXEC.BAT AUTO.OLD
```

If our current directory were C:\DOS and we wanted perform the copy but keep the filename as AUTOEXEC.BAT we could omit the *destination path* and *destination filename* and could shorten the command still further to:

```
COPY C:\AUTOEXEC.BAT .
```

where the period (.) is the DOS short-hand representation of 'current directory'. We could shorten this command still further by omitting the destination directory altogether:

```
COPY C:\AUTOEXEC.BAT
```

This is because DOS always takes a file (or files) to the current directory by default.



### Exercise

3.6 Write the copy command needed to copy a file called OBSCURE.DOC from the WORDART directory to the CORRESP directory, keeping the same filename. Assume the current directory is C:\ (Remember that we are referring to the directory structure in Figure 3.4.)

3.7 Assuming you are located in C:\WORK, write the command to copy the file CHAP1.DOC to CHAP1.BAK located in the BOOKS sub-directory.

## DELETING FILES

DEL is the command for deleting files. Its syntax is very simple:

```
DEL path\filename
```

### Example

To delete the file named AUTO.OLD in the DOS directory we could type in:

```
DEL C:\DOS\AUTO.OLD
```



### Exercise

3.8 Write the command to delete a file called OBSCURE.DOC from the WORDART directory. Assume the current directory is WORDART.

3.9 Write the command to delete a file called OBSCURE.DOC from the WORDART directory if you are located in C:\MARK\WORK\BOOKS\NETWORK.

## RECOVERING DELETED FILES

UNDELETE is the command for recovering files previously deleted using the DEL command. It will generally work only if no other files have been written to the disk since the file was deleted. Its syntax is very simple:

```
UNDELETE path\filename
```

UNDELETE has some useful options, the most important ones being:

- if no *path\filename* is specified it will attempt to recover all deleted files in the current directory
- if the command is followed by /LIST it will display a list of files which may be able to be recovered in the current directory.

### Example

To undelete the file named AUTO.OLD in the DOS directory we could type in:

```
UNDELETE C:\DOS\AUTO.OLD
```

To get a list of all files which could be undeleted we type in:

```
UNDELETE /LIST
```




### Exercise

3.10 Write the command to undelete a file called OBSCURE.DOC from the WORDART directory. Assume the current directory is C:\

## RENAMING FILES

The RENAME (or REN) command is for changing the name of a file. Its syntax is also quite simple:

```
RENAME old_filename new_filename
```

The more general command format is:

```
RENAME old_path\old_filename new_filename
```

### Example

To rename the file named AUTO.OLD in the DOS directory we could type in:

```
RENAME C:\DOS\AUTO.OLD AUTO.2
```

or

```
REN C:\DOS\AUTO.OLD AUTO.2
```



### Exercise

3.11 Write the command to rename a file called AUTO.OLD to AUTOEXEC.BAT in the DOS directory. Assume the current directory is:

C:\WINDOWS\MSAPPS\WORDART

## DISPLAYING FILES

TYPE is the command for displaying the contents of text files on the screen. If used on files other than text files (for example, DOS programs), the results may be unpredictable. Again, its syntax is very simple:

```
TYPE filename
```

or, more generally:

```
TYPE path\filename
```

### Example

As .SYS files contain text data, we may TYPE them. To TYPE the file named CONFIG.SYS in the root directory we could enter the following command:

```
TYPE C:\CONFIG.SYS
```

To display the AUTOEXEC.BAT startup command file, we could enter:

```
TYPE C:\AUTOEXEC.BAT
```

## THE PIPE OPERATOR

When displaying a very long text file with the TYPE command, it will scroll off the screen very rapidly and we will not be able to see most of it. To avoid this problem we can **pipe** (or pass on) the output of the TYPE command to the MORE command as follows:

```
TYPE C:\CONFIG.SYS | MORE
```

One of the functions of the MORE command is to **paginate** any screen output, i.e. it will cause any data sent to the screen to be displayed one screen at a time. In effect we have two commands working here:

- TYPE — to display the contents of a file
- MORE — to paginate the output from TYPE.

The two commands a joined together with the **pipe operator**. This is the vertical bar character ( | ) — not to be confused with the exclamation point (!).

### Important point

If not done so already, you may need to add the line:

```
SET TEMP=C:\DOS
```

to the AUTOEXEC.BAT file to make this mechanism work properly.




### Exercise

3.12 Write the command to display the contents of a long file called SYSTEM.INI in the WINDOWS directory. Assume the current directory is MSDRAW.

## PRINTING FILES

As the name suggests, PRINT is the command for printing files. Its syntax is:

```
PRINT filename
```

or, more generally:

```
PRINT path\filename
```

### Example

To print the file named AUTOEXEC.BAT in the root directory we could type in:

```
PRINT C:\AUTOEXEC.BAT
```




### Exercise

3.13 Write the command to print a file called OBSCURE.DOC from the WORDART directory. Assume the current directory is WORDART.

## DISPLAYING THE DIRECTORY CONTENTS

As we have already seen, DIR is the command used to list the files in a directory. Its syntax has four common forms:

```
DIR
```

```
DIR filename
```

```
DIR path
```

```
DIR path\filename
```

### Example

To list the files in the root directory we could type in:

```
DIR C:\
```

To look for the file DISKS.DOC in the HARDWARE directory of Mark's directory structure, we would type in:

```
DIR C:\MARK\WORK\BOOKS\HARDWARE\DISKS.DOC
```



### Exercise

3.14 Write the command to list the files from the WORDART directory.

## MANIPULATING DIRECTORIES

There are far fewer commands for manipulating directories than for manipulating files. The main commands are:

- CHDIR or CD — change directory
- MKDIR or MD — create a new directory
- RMDIR or RD — delete or remove an existing directory
- TREE — display the directory tree structure.

### CHANGE DIRECTORY

CHDIR (or CD) is the command which lets us change the current directory. To use it we need to specify the directory we want to make the *new*, current directory. Its syntax is:

```
CHDIR path
```

or

```
CD path
```

### Example

To change current directory to the GRPHLT directory we type in:

```
CD C:\WINDOWS\MSAPPS\GRPHLT
```



### Exercise

3.15 Write the command to change current directory from the WORDART directory to the MSDRAW directory.

## CREATING DIRECTORIES

MKDIR (or MD) is the command which lets us create a new directory. To use it we need to specify the directory we want to make or create, and the path to it. Its syntax is:

```
MKDIR path\new_directory
```

or

```
CD path\new_directory
```

### Example

To create a new sub-directory called NEW in the GRPHLT directory we type in:

```
MD C:\WINDOWS\MSAPPS\GRPHLT\NEW
```



### Exercise

3.16 Write the command to make a new sub-directory in the WORDART directory called CLIP.

## DELETING DIRECTORIES

To delete existing directories we us the RMDIR (or RD) command. To use it we need to specify the directory we want to delete, and the path to it. MS-DOS will not let you delete a directory which still has files or sub-directories in it. The syntax is:

```
RMDIR path\directory
```

or

```
RD path\directory
```

### Example

To delete the NEW directory which we created earlier in GRPHLT, we type in:

```
RD C:\WINDOWS\MSAPPS\GRPHLT\NEW
```




### Exercise

3.17 Write the command to delete the directory WORDART\CLIP.

## DISPLAYING THE DIRECTORY STRUCTURE

TREE lets us see the relationships between directories. It displays, in character based graphical form, all the child directories of the directory specified. Its syntax is:

```
TREE path\directory
```

If no *path\directory* is specified, the child directories of the current directory will be displayed.

### Example

To see the sub-directories of the MSAPPS directory, we type in:

```
TREE C:\WINDOWS\MSAPPS
```



### Exercise

3.18 Write the command to see the sub-directories of the root directory.

## ABSOLUTE AND RELATIVE PATHNAMES

When we referred to sub-directories several levels down the directory tree, we saw that the pathnames could be quite long, e.g.:

```
C:\MARK\WORK\BOOKS\HARDWARE
```

This referencing method is referred to as **absolute** pathnames. The pathname is described absolutely, i.e. in its entirety from the root directory — C:\ in this case. This method of referencing can be quite tedious at times.

There is an alternative method which is often useful. Files and directories can be referred to with **relative** path/filenames; i.e., names which are relative to the current working directory.

### Example

Assume that Mark is located in his HARDWARE directory but wants to work in his BOOKS directory. We have seen that he can CD to it by:

```
CD C:\MARK\WORK\BOOKS
```

That is fine, but tedious. After all, if you refer to Figure 3.4 again, you can see that Mark simply wants to go 'up one directory'. In other words, Mark wants to CD to the parent directory. DOS has a special name for this — the double period ( .. ). Mark can achieve the same result as above by typing:

```
CD ..
```

This is a relative pathname — it is specified relative to where the user is currently located, not from the root directory. Taking this concept further, Mark can CD to the COOKING sub-directory, from the HARDWARE sub-directory as follows:

```
CD ..\COOKING
```

which is a lot simpler than typing:

```
CD C:\MARK\WORK\BOOKS\COOKING
```

Relative referencing can be applied to filenames and other DOS commands. For example, Mark could access the files CHAP.1 CHAP.2 and CHAP.3 in the NETWORK sub-directory, from his current directory in COOKING, by commands such as:

```
TYPE ..\NETWORK\CHAP.1
PRINT ..\NETWORK\CHAP.2
TYPE ..\NETWORK\CHAP.1 | MORE
DEL ..\NETWORK\CHAP.1
```

and so forth.



### Self-test

3.8 What is the difference between relative and absolute paths?
Classify the following path/filenames as absolute or relative:

```
C:\DOS
C:\DOS\FDISK.EXE
DOS\FDISK.EXE
FDISK.EXE
D:\PROGS\ACCTS\PAYROLL.EXE
..\NETWORK\CHAP.1
..\..\CORRESP
D:\
```




### Exercise

3.19 Assume you are located in Mark's NETWORK directory. Using relative path/filenames, write the commands to:

- display the contents of CHAP.7 in HARDWARE
- delete WEEK.1 in LECTURES
- copy WEEK.2 IN LECTURES to CHAP.2 in the current directory
- change directory to COOKING.

# WILDCARDS

Often when using MS-DOS we may wish to work with a group of files rather than a single file. For instance if Mark wanted to copy all his document files from directory C:\MARK\CORRESP to drive A:\ROOT, MS-DOS would allow him to do this easily by using **wildcards**. Wildcards are special characters which may be used in place of other characters or groups of characters. MS-DOS has two types of wildcards:

**?** — matches exactly one character
* — matches zero or more characters.

## SINGLE CHARACTER MATCHING

The question mark wildcard symbol ( **?** ) is used to match a single character, so if Mark had a series of files called:

MARK1.DOC
MARK2.DOC
MARK3.DOC
MARK4.DOC
...
MARK9.DOC

and he wanted to COPY them from drive C: to drive A: then he could then type:

```
COPY C:\MARK\CORRESP\MARK?.DOC A:
```

The group of files in the above example are all those files whose names start with 'MARK', followed by one other symbol (which could be anything) and having a .DOC extension.

## MULTIPLE CHARACTER MATCHING

The asterisk ( * ) wildcard symbol matches a group of characters of any length. For example, if we wanted to list all documents in the current directory with the .DOC extension, we could type in:

```
DIR *.DOC
```

As we can see, this saves a considerable amount of typing when manipulating similarly named files.



### Self-test

3.9 What is the difference between the ? and * wildcard symbols?

3.10 CHAP.* will match which of the following files?

| | |
|---|---|
| CHAP.1 | CHAPS.1 |
| CHAPTER.123 | CHAP.123 |
| CHAP.DOC | CHAP |
| CHEP.TXT | CHOP.XX |

3.11 CH?P.??? will match which of the following files?

| | |
|---|---|
| CHAP.1 | CHAPS.1 |
| CHAPTER.123 | CHAP.123 |
| CHAP.DOC | CHAP |
| CHEP.TXT | CHOP.XX |

3.12 CH?P.* will match which of the following files?

| | |
|---|---|
| CHAP.1 | CHAPS.1 |
| CHAPTER.123 | CHAP.123 |
| CHAP.DOC | CHAP |
| CHEP.TXT | CHOP.XX |




### Exercise

3.20 Write the command to list all the files in the DOS directory which have the extension .EXE.

3.21 Write the command to list all files in the current directory which begin with 'A' and have three characters in the filename and any extension.

3.22 Write the command to list all files in the current directory which have exactly two characters in the filename and have extensions beginning with 'D'.

## CHAPTER SUMMARY

In this chapter we looked at many of the aspects of file management. We began with file naming conventions and how MS-DOS saves and reads files. We then looked at the features MS-DOS provides which allow us to organise our files on our disk drives and in our directories. We also looked at how a directory structure is laid out.

This led us to the concept of paths, which is the way in which we navigate the directory structure. We also looked at default and active drives which add convenience to the process. Finally, we examined a number of common file management commands and the use of relative filenames and wildcards with those commands.



## Review questions

3.1 What are the legal lengths for filenames and extensions in MS-DOS?

3.2 What are the valid characters for filenames other than the alphabetic characters and the digits?

3.3 Why can a '?' character not be used within a filename?

3.4 What filenames are reserved for system use by DOS?

3.5 How is a bit stored on a disk?

3.6 What is a sector?

3.7 What does DOS use the FAT for?

3.8 What is the purpose of directories on a disk?

3.9 What is the minimum number of directories on a drive?

3.10 What is a path?

3.11 What command is used to tell MS-DOS where to look for programs to execute?

3.12 What is the difference between the default drive and the active drive?

3.13 List the common commands which are used to manipulate files.

3.14 List the common commands which are used to manipulate directories.

3.15 What is the DOS shorthand notation for:
- current directory?
- parent directory?

3.16 What command will copy all the files in the root directory of C: with the extension .BAT to the root directory of A: ?

3.17 Does the copy command always require a destination? Explain your answer.

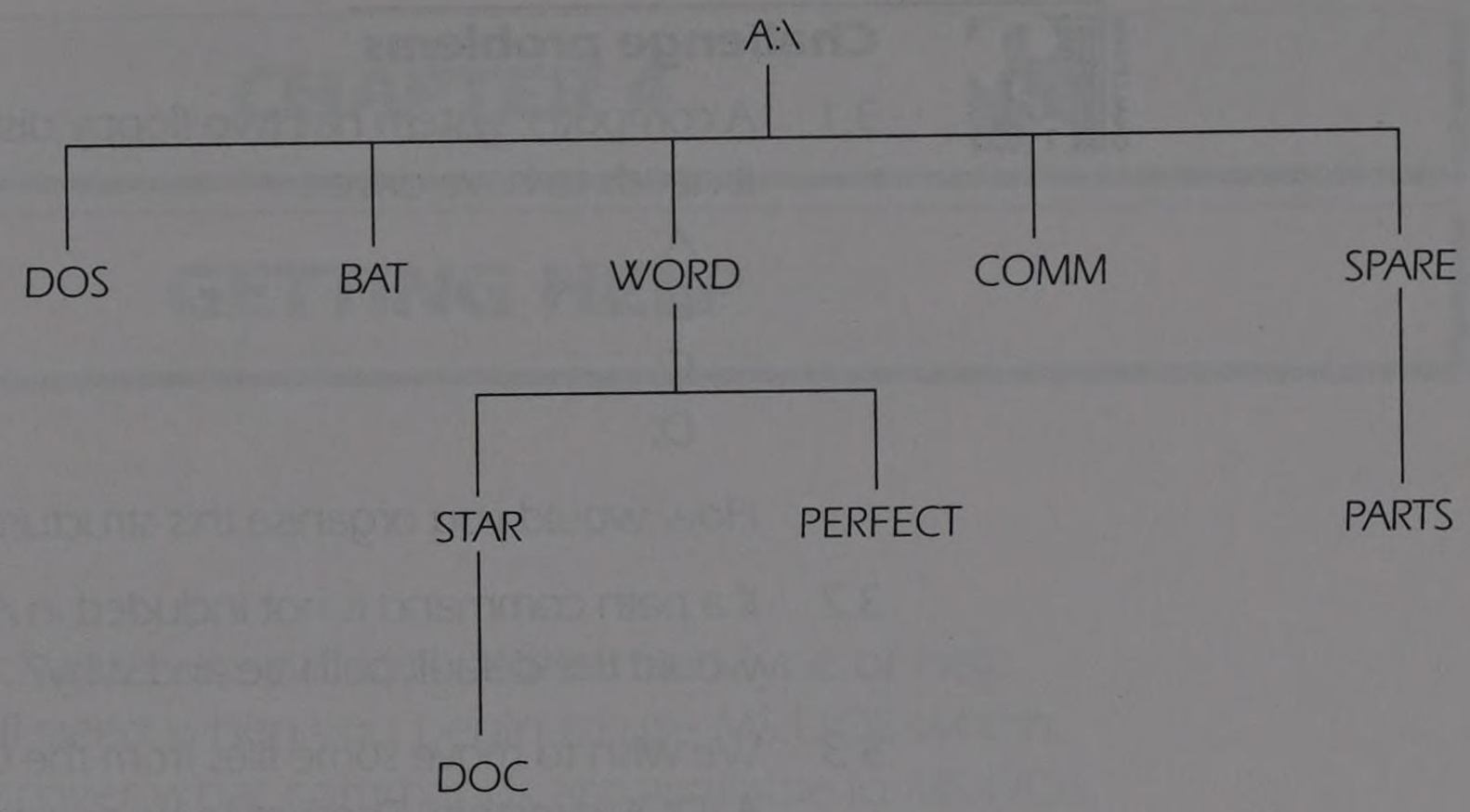
3.18 Describe the differences between the two wildcards.

3.19 What is the essential difference between a relative path and an absolute path?



## Review exercises

The following exercises are intended to be done on an MS-DOS computer, using the following directory structure. You will need to create the following directory structure before you begin these exercises.




3.1 Create the directory structure shown above on a blank, formatted floppy disk.

3.2 Copy the file AUTOEXEC.BAT from the root directory of drive C: to the root directory of drive A: and write down the command used.

3.3 Use wildcards the copy all the files with the extension .SYS from the DOS directory on drive C: to the DOS directory on drive A:. Write down the command used.

3.4 Check that the file transfers performed in Review exercise 3.3 took place and write the command required to check this.

3.5 Change drives from C: to A: and write down the command used.

3.6 Rename the AUTOEXEC.BAT file in the root directory of A: to AUTO.BAT and write down the command used.

3.7 Write down the commands to perform the following functions:
- print the AUTO.BAT file of Review exercise 3.6 if your current directory is C:\DOS
- delete the DOS directory on A: (Hint: there may be more than one command required.)

3.8 Create a directory TEMP in the C: directory on your hard drive. Copy some files from your diskette to this directory.

3.9 Using a spare floppy disk and the temporary directory of the hard disk, practise working with the following commands:
- COPY
- REN
- CD
- DEL
- RD
- UNDELETE

Use the TREE, DIR and TYPE commands as appropriate, to check the results of commands that you execute. Briefly note the results of your commands.




## Challenge problems

3.1 A computer system has two floppy disks and one hard disk. It needs to have drives:

A:
B:
C:
D:

How would you organise this structure?

3.2 If a path command is not included in AUTOEXEC.BAT what would the default path be and why?

3.3 We wish to move some files from the C:\DOS directory to the A:\DOS directory. The files to be moved must:

- end in .EXE or .COM
- have five-letter file names
- start with the letter 'M'.

What is the simplest way to achieve this?

# CHAPTER 4

# GETTING HELP

In this chapter we will look at the main type of help you will need when you begin to use MS-DOS which is to discover what commands are available in MS-DOS and how to use them. This type of help is supplied by the on-line help system which is part of MS-DOS. To use this system effectively, you need to have an understanding of the conventions used when commands and options are described.

At the end of this chapter you will be able to:

- access the MS-DOS command line help
- interpret the elements of the on-line help command syntax
- use the MS-DOS on-line help system to explain how to use of the command syntax, command notes and examples.

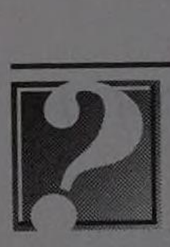
# ON-LINE HELP FOR MS-DOS COMMANDS

MS-DOS version 5.0 and earlier was distributed with very comprehensive manuals which included a detailed description of each command available, as well as chapters on a number of other topics. The manuals from version 6.0 and later are much smaller in a number of key areas. These manuals also lack a reference section for commands.

Versions 6.0 and later rely on an on-line help system for command usage which contains information similar to the command reference section in the earlier manuals. Unfortunately, not all the information available in the manual for version 5.0 has been included in the on-line help in version 6.0 on, so it is still useful to have a copy of the MS-DOS version 5.0 manual if you can get one.

The on-line help is available in two forms:

- command line help — this displays the syntax for the specified command and then returns you to the DOS command line
- MS-DOS Help — this is an interactive help system. It includes syntax information, notes and examples for each MS-DOS command.



**Self-test**

4.1 Why is the on-line help important to using DOS?

# DOS COMMAND SYNTAX

One of the most difficult things to understand when first accessing help is the notation used to describe how to use the commands. Let us examine this notation and then look at each type of help in turn.

**Syntax** refers to the order and spelling of the words and characters that we use with MS-DOS commands. This includes any **parameters** and **switches** which follow the command. A parameter is a piece of information which the command will act upon, such as a filename, while a switch is a command option which changes the way the command will act. For the following explanations and exercises, we will use the imaginary command 'SAMPLE' to illustrate the use of the various elements of DOS command syntax.

When you see a command, in capital letters, you must type it exactly as it appears in the syntax line. Elements which appear in italics are **placeholders** representing information which you, the user, will need to supply. The placeholders in this example are:

- drive:
- path
- filename
- options.

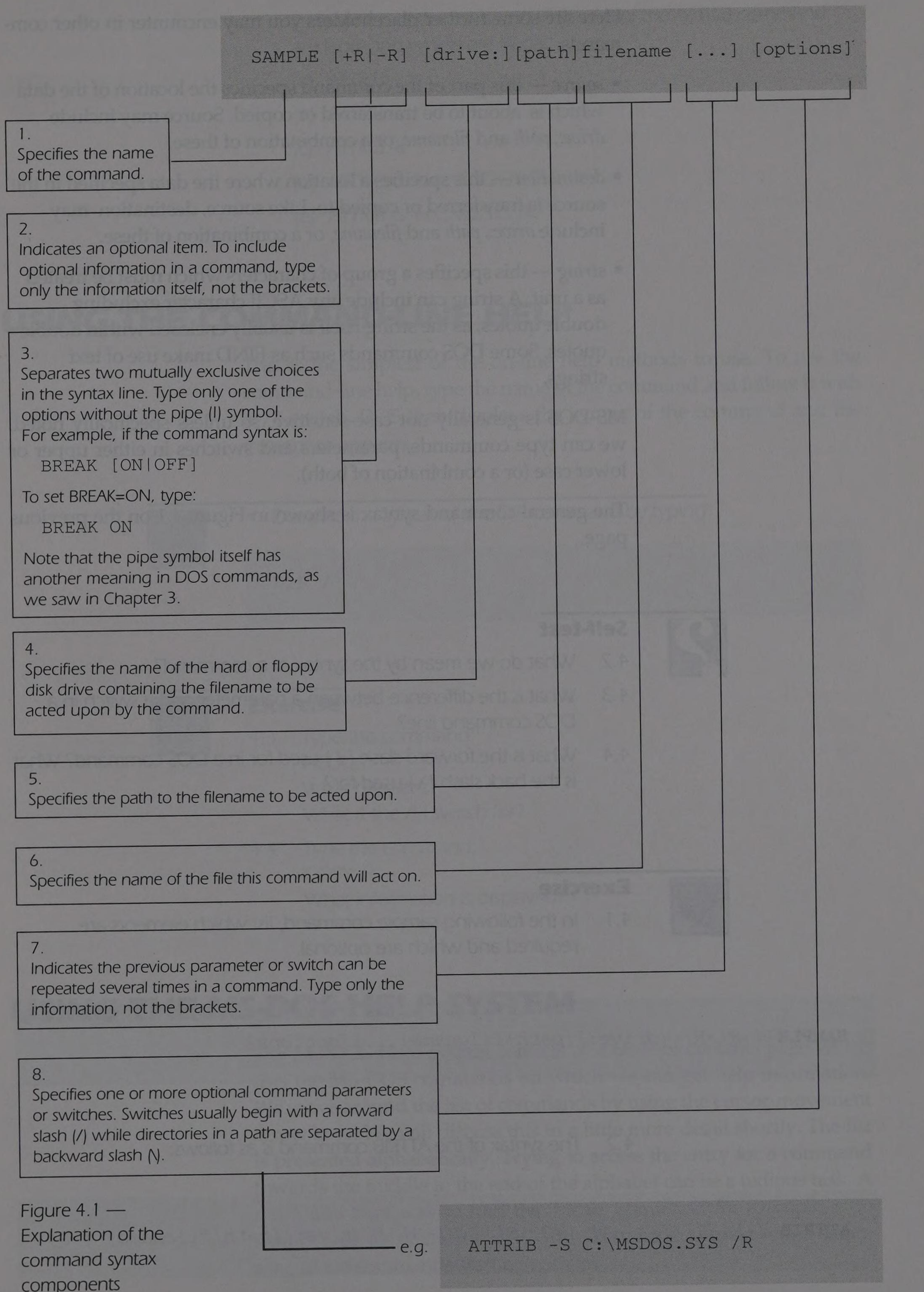



Figure 4.1 — Explanation of the command syntax components

Here are some further placeholders you may encounter in other commands:

- *source* — this part of the command specifies the location of the data which is about to be transferred or copied. Source may include *drive:*, *path* and *filename*, or a combination of these.
- *destination* — this specifies a location where the data specified in the source is transferred or copied to. Like source, destination may include *drive:*, *path* and *filename*, or a combination of these.
- *string* — this specifies a group of characters which must be treated as a unit. A string can include any ASCII character excluding double quotes, as the string itself is usually enclosed within double quotes. Some DOS commands such as FIND make use of text strings.

MS-DOS is generally not case-sensitive, so unless specifically noted, we can type commands, parameters and switches in either upper or lower case (or a combination of both).

The general command syntax is shown in Figure 4.1 on the previous page.



### Self-test

4.2 What do we mean by the syntax of a command?

4.3 What is the difference between a parameter and a switch in a DOS command line?

4.4 What is the forward slash ( / ) used for in a DOS command? What is the back slash ( \ ) used for?




### Exercise

4.1 In the following sample command, list which elements are required and which are optional.

```
SAMPLE [+R|-R] [drive:][path]filename[...][options]
```

4.2 The syntax of the ATTRIB command is as follows:

```
ATTRIB [+R|-R] [+A|-A] [+S|-S] [+H|-H] [[drive:][path]filename] [/S]
```

State which of the following instances of the ATTRIB command are legal and which are illegal:

(a) ATTRIB

(b) ATTRIB -R

(c) ATTRIB -R *.*

(d) ATTRIB -S -H C:\MSDOS.SYS

(e) ATTRIB -S -H C:\IO.SYS /R

## USING THE COMMAND-LINE HELP

This is the simplest of the on-line help methods to use. To use the command-line help, type the name of the command and follow it with a /? switch. DOS then displays the syntax of the command and the prompt re-appears.




- Examine the syntax of the DIR command by typing:

```
DIR /?
```




**Exercise**

4.3 Type the command:
HELP /?
What happens?
What is the /H switch for?

4.4 Type the command:
COPY /?
What information is displayed?

## USING THE MS-DOS HELP SYSTEM

The MS-DOS Help system consists of a table of contents page which lists the MS-DOS commands on which we can get help information. We move around the list of commands by using the cursor-movement arrow keys. We will discuss this in a little more detail shortly. The list is presented alphabetically. Trying to access the entry for a command towards the middle to the end of the alphabet can be a tedious task. A quick and easy way to find the 'Move' command for example, is to simply type the character 'm'. This will bring the cursor to the beginning of the commands starting with 'm'.

When you first access help for any of the commands enclosed in the < > symbols, the system displays a **Syntax page** for that item. Most syntax pages have a **Notes page** and an **Examples page** attached to them. The relationship is shown below.

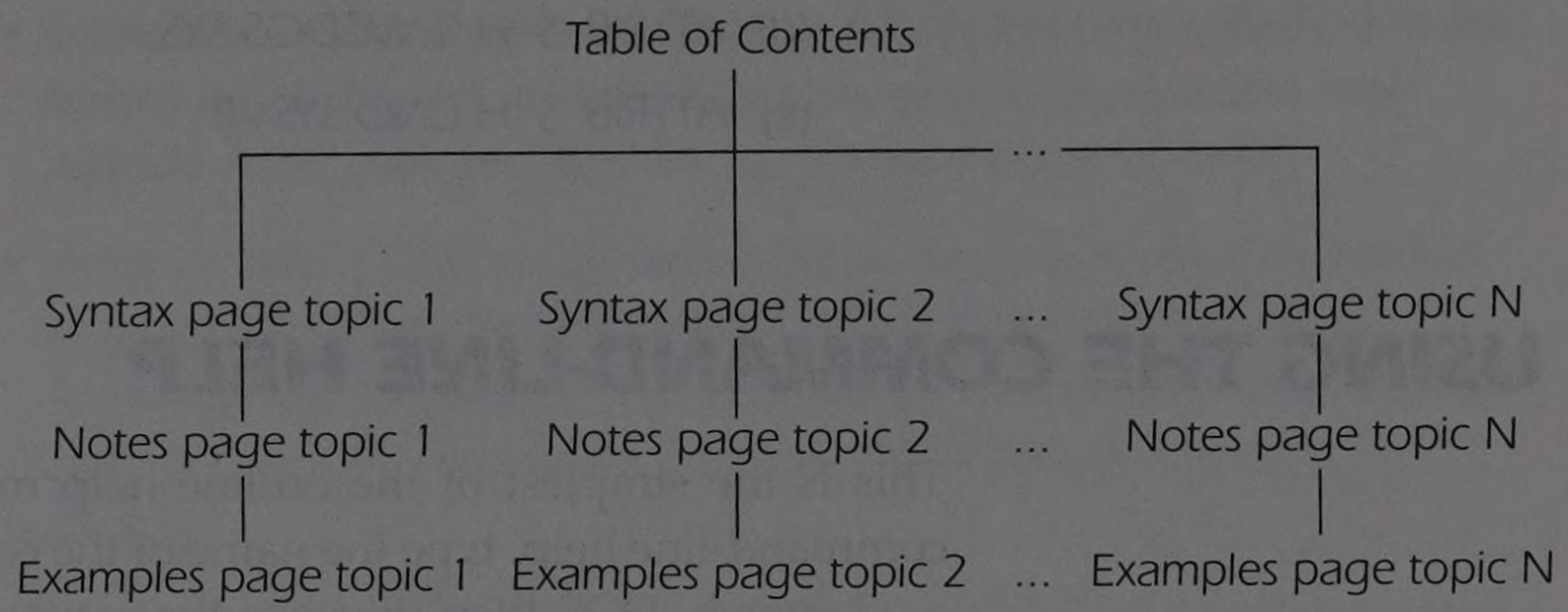



Figure 4.2 — Structure of the DOS help system

The syntax pages:

- contain a description of the command
- list the correct syntax for that command.

The examples pages give you some examples of how to use the command.

The notes pages provide any additional or extra information about the command that is useful or necessary for you to know.

You can move between the pages using the following keystrokes:

- <ALT> + <C> — (hold down the <ALT> key and the <C> key at the same time) this returns you from whatever page you are on, back to the contents page
- <ALT> + <B> or <ESC> will take you to the last page you were viewing
- <ALT> + <N> will take you to the next page; that is, from the syntax page to the notes page, or from the notes page to the examples page.

When you have scrolled through the syntax, notes and examples pages for a topic, pressing <ALT> + <N> will take you to the syntax page, for the next help topic available on the list.




- Use the above commands to practise scrolling through the help screens.

## GETTING HELP — METHOD 1

The MS-DOS help system can be entered in two ways. The first is by typing 'HELP' at the command prompt. This will bring up the MS-DOS help table of contents. We can access the following two pull-down menus from every screen that we open from here:

- File — the File menu allows us to exit MS-DOS Help, or get a printout of the help information available on the current topic
- Search — the SEARCH menu allows us to search for all occurrences of a particular string within every topic available. It also allows us to repeat our last search.

As mentioned previously, there is a syntax page for every item listed on the Help screen and enclosed in the < > symbols. We can choose a topic by using the:

- up
- down
- left
- right

arrow keys. This will position the cursor on one of the Help items. Once we have come to the item we require, we press <ENTER>.

If a mouse is connected to the DOS system, we simply position and click on the desired topic. When the syntax page appears, we will see the syntax information displayed for that particular command. In the top left-hand corner of the page we will see the headings <Notes> and <Examples>. Choosing these items moves us directly to the page specified. Alternatively, we can use <ALT> + <N>. Similarly, when at the notes page, the headings <Syntax> and <Examples> appear. These allow us to move freely between pages.

## GETTING HELP — METHOD 2

The second method for entering MS-DOS Help bypasses the table of contents page and takes us directly to the syntax page of a particular command. To use this method we must know the name of the command we want help on. We can then type:

```
HELP command_name
```

at the DOS prompt. The syntax page for the particular command will appear. We can now perform any of the operations we have already discussed.

When referencing information on a particular Help command, we may find another Help topic enclosed in < > brackets within the body

of information. This indicates that there is additional or related information in this second topic. This cross-referencing of commands is a useful feature of MS-DOS Help. Choosing this item will **jump** you to the syntax page of the second command. The <ALT> + <B> keys will return you to whichever screen you were on before the jump.

## EXITING MS-DOS HELP

From any page we may exit MS-DOS Help by choosing Exit from the File menu or by typing:

```
<ALT> + <F>  <X>
```



### Self-test

4.5 What are the two ways to enter the MS-DOS Help system?

4.6 What are the advantages and disadvantages of each method?




### Exercise

4.5 Briefly describe what information can be found in the following:

- Syntax page
- Notes page
- Examples page.

# CHAPTER SUMMARY

In this chapter we looked at the two main types of on-line help that are available when using MS-DOS. These were:

- the command-line help, and
- the MS-DOS Help system.

We saw how to interpret the syntax of commands and looked at the ways in which command-line help and the MS-DOS Help system may be invoked. Additionally, we looked at the internal structure of MS-DOS Help and how to move around inside it to find the required information.



### Review questions

4.1 Describe what happens when you type the command 'ATTRIB /?'.

4.2 Describe what happens when you type the command 'HELP DIR'.

4.3 What are the different ways of obtaining on-line help from DOS?

4.4 Where would you look for additional help if you could not find what you need in the on-line help?

4.5 What is the purpose of the following placeholders in a DOS command line:

- drive: ?
- path?
- filename?
- options?



## Review exercises

Start up the MS-DOS Help and get some help information on the following commands. For each command:

- describe its purpose, in your own words
- record its syntax.

Summarise your notes in a table like that shown in the table below.

| No. | Command | Syntax | Purpose |
|---|---|---|---|
| 4.1 | CLS | | |
| 4.2 | XCOPY | | |
| 4.3 | DATE | | |
| 4.4 | DBLSPACE | | |
| 4.5 | DELTREE | | |
| 4.6 | DISKCOPY | | |
| 4.7 | DISKCOMP | | |
| 4.8 | FIND | | |
| 4.9 | MOVE | | |
| 4.10 | VERIFY | | |




## Challenge problems

4.1 There are certain topics in the MS-DOS Help system which do not follow the standard format outlined in this chapter. Use MS-DOS Help to examine the topic Batch commands and describe how the format of this topic varies from the typical command structure.

4.2 Unlike the original manuals, MS-DOS Help does not tell you which commands are internal and which are external. From your knowledge of MS-DOS, can you work out a way to find out this information?

# CHAPTER 5

# HARD DISK DRIVES

Nearly all modern PCs base their secondary storage on magnetic hard disk systems. From the point of view of the operating system, the hard disk is a place from where the operating system is loaded and where applications programs and data are stored. In this chapter we will examine the features and characteristics of MS-DOS which relate to managing and utilising hard disks.

At the end of this chapter you will be able to:

- describe the two most common types of hard disks
- describe the process of formatting and labelling a hard disk drive
- define the role of the hard disk in the boot process
- describe disk management
- use the hard disk related commands.

# HARD DISK BASICS

A hard disk drive consists of:

- an aluminium **platter** which is coated with a thin layer of magnetic material, usually cobalt or iron oxide, on which the information is stored as a pattern of magnetic bits
- a **read/write head** which detects the magnetic alignment of the coating when reading, or magnetises the coating in the appropriate way when writing
- a **spindle motor** which rotates the platter at about 3600 - 5400 RPM. The whole unit is usually hermetically sealed to prevent the entry of dust. This type of disk, known as a **Winchester** disk, is the type generally used in microcomputers today.

## DISK FORMATTING

A new disk is **unformatted**. This means that the magnetic material on the platter has no magnetic patterns on it at all. The process of formatting is to divide up the disk surface into **tracks** and **sectors**. Tracks are concentric rings on the disk surface and sectors (or blocks) are small segments of the track. When data is stored on the disk it is divided up into **blocks**. Each block of data is capable of holding a set number of bytes and is designed so a block of data fits into one sector on the disk. Often, the terms *sector* and *block* are used interchangeably.

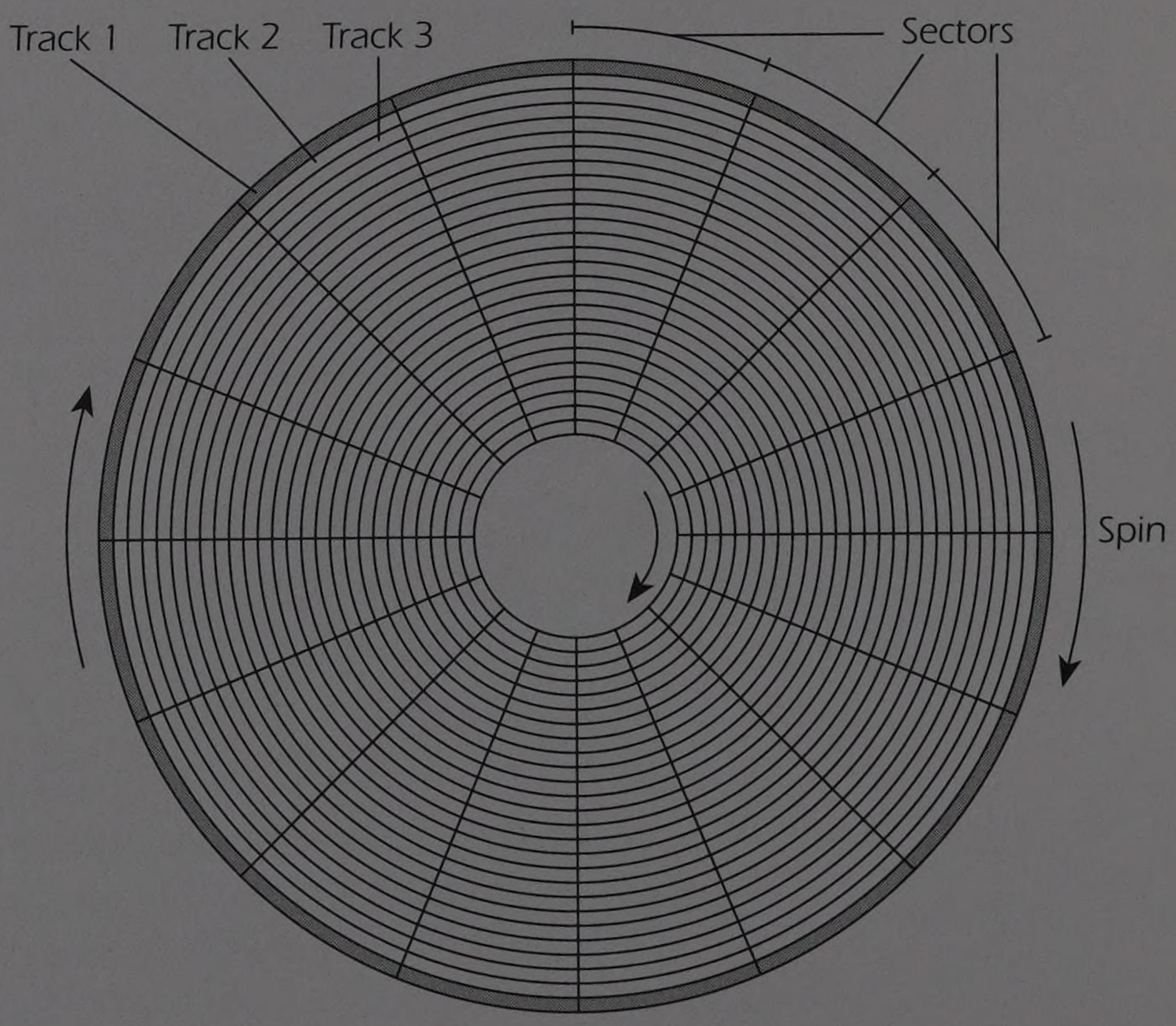



Figure 5.1 —
Disk tracks and sectors

## DISK STORAGE CAPACITY

The disk capacity is the amount of data which may be stored on the disk. This is determined by:

- the number of tracks on the disk
- the number of sectors in each track
- the number of disk platters or surfaces.

Bigger disks must have more tracks on the disk and/or the tracks must be closer together. Large capacity drives may also have more than one platter or surface in order to increase the storage area. Common drives for microcomputers range from tens of megabytes to several gigabytes. (A megabyte is $2^{20}$ bytes — just over 1 million bytes. A gigabyte is $2^{30}$ bytes — just over a billion bytes!)

## DISK ACCESS TIMES

Disk access time is the time taken from the instant the computer requests the information from the disk drive until the disk drive makes the information available to the application. Typical times for modern drives are between 9 milliseconds (ms) and 20 ms.

The method by which the heads are moved back and forth across the disk surface makes a significant difference to access time. The fastest, most common access mechanisms today use a **voice coil** assembly not unlike the magnetic coil and fixed magnet used to move a loudspeaker coil.

## READ/WRITE HEADS

The disk has a read/write head for each surface of each platter. They are mounted very close to the surface of the disk but do not touch. This allows a very high disk speed but means that the distance must be carefully controlled, hence the rigid platter. Small particles of dust colliding with the head would be enough to cause the head to contact the surface of the disk resulting in damage to both the head and the disk surface. This is called a **head crash** and is a serious and expensive failure.

## CYLINDER ORGANISATION

Where multiple platters are used in a disk, there is a read/write head for each platter's surface. The heads are all mounted on the same movement system. This means that when the head on platter 1, side 1, is on a specific track, the heads on platter 2, sides 1 and 2, platter 3, sides 1 and 2 and so on, are all aligned on the same track. This collection of tracks is called a **cylinder**. We can see that it would be faster to store a file on one cylinder (same track on different platters) than on several tracks of one platter as the heads only have to be positioned once.

## FIXED AND REMOVABLE PLATTERS

The single platter, hermetically sealed units used for microcomputers are fixed platter drives (hence the term **fixed disk**). Some larger, multi-platter drives have one or more removable platters to allow fast **backup** or transfer of large quantities of data.



**Self-test**

5.1 What is a sector?

5.2 What is a track?

5.3 What is a cylinder?

5.4 What determines the access time of a disk?

5.5 Why is a fast disk important to an operating system such as DOS?

## HARD DISK TYPES

The two most common types of hard disks in use in PC systems today are:

- IDE — Intelligent Drive Electronics
- SCSI — Small Computer Systems Interface drives.

These names describe the type of interface between:

- the hard disk drive
- the computer's hardware
- the BIOS.

The IDE interface is the most common and is used on drives up to around 600 megabytes. An IDE controller can normally support up to two drives, although more specialised controllers are available which support up to four drives. The operating system communicates with IDE drives generally via the BIOS.

SCSI drives start at about 250 megabytes and range up to several gigabytes in capacity. The SCSI interface allows up to seven drives to be connected but requires a special driver to be used as SCSI drives are not supported by BIOS.



**Self-test**

5.6 Which type of disk drive communicates to the operating system through the BIOS?

# DISK ORGANISATION

As mentioned before, the process of **formatting** defines a series of concentric tracks on the disk surface. It divides those tracks into sectors, each of which holds a fixed number of data bytes. In MS-DOS the number of bytes per sector is 512. As well as defining the tracks and sectors, the operating system does two other things:

- it allocates a name or label to the disk for future identification
- it records, on the disk, information about the data stored.

MS-DOS does this by writing:

- a boot record
- a file allocation table (FAT)
- a root directory

onto the newly formatted disk surface. Let us now look at the function of each of these elements.

## BOOT RECORD

The boot record always occupies:

- the first sector,
- on the first track,
- on the first side,
- of the first disk.

The boot record sector contains a short program which tells the computer:

- how to load the operating system
- where to load the operating system from.

In addition to this, a number of key statistics about the disk are stored here. These are:

- the number of bytes per sector
- the total number of sectors on the disk
- number of sectors per track
- the number of heads on the disk
- the disk label.

## THE FILE ALLOCATION TABLE

The **file allocation table** (FAT), is a table which keeps track of which sectors are in use and which file they belong to. (Note that the files are not necessarily stored in sequential sectors.) The FAT also records any bad tracks on the disk and will prevent data being written to these bad tracks.

### THE ROOT DIRECTORY

The **root directory** maintains a list of the files stored on the disk. It keeps such information as:

- filename
- extension
- file attributes
- date and time
- starting FAT entry
- file size.



**Self-test**

5.7 What is the purpose of:
- the boot record?
- the FAT?
- the root directory?

## FORMATTING AND LABELLING COMMANDS

There are three different commands which relate to formatting a disk and three which relate to labelling.

### DISK LABELLING

The MS-DOS has three commands that can work with disk labels:

- LABEL
- DIR
- VOL.

The DOS command which can physically write a disk label is, LABEL. It can also be used for reading the label of the disk. The syntax for labelling a disk is:

```
LABEL drive: disklabel
```

If we want to read the label of a disk, we type:

```
LABEL drive:
```

MS-DOS will respond with:

```
Volume in drive <drive> is  <disklabel>
Volume label (11 characters, ENTER for none)? _
```

If you do not want to re-label the disk at this point, press <ENTER>.
MS-DOS will respond with:

```
Delete current volume label (Y/N)?_
```

You should respond 'N' to this.




### Exercise

5.1 Write down the command to name a computer system's drive D: as 'WORK_DRIVE'.

Another way to read the label of a disk is to use DIR. This gives the volume label as well as directory information. The DIR command is used in the same way as we saw in Chapter 4.

The other way to read the disk label is to use the VOL command. The syntax for the VOL command is:

```
VOL drive:
```

### Example

The following command gets the C: drive label:

```
VOL C:
```

DOS responds with something like the following:

```
Volume in drive C: is SCOTTIE_540
Volume Serial Number is 1D41-0357
```




- Type in the command to get the volume details of your hard disk.
- Do the same for your second drive, if your PC has one.



### Self-test

5.8 Do you expect the serial number on different disks to be the same? What about volume labels?

## DISK FORMATTING

There are three commands which relate to formatting a disk:

- FDISK
- FORMAT
- UNFORMAT.

## FDISK

FDISK is a complex MS-DOS utility which will allow you to perform several functions on the disk drive specified. One of its most important tasks is to **partition** the disk. This is a process of allocating different parts of the disk to different logical drives. This allows, for example, two logical drives, C: drive and D: drive, to be created on the same physical disk.

Alternatively, partitioning may be used to allow MS-DOS to use part of the disk and some other operating system (Unix, for example) to use another part of the same disk.

FDISK will also identify bad sectors — areas of the disk which will not reliably hold data. It will then write a **bad sector map** to the disk.

### Important point

Sophisticated drives such as IDE and SCSI devices do their own bad sector management as well as some other high level functions which are also not compatible with FDISK. Hence, FDISK should not generally used with these types of drives other than to create partitions if required.

## FORMAT

FORMAT formats a disk or partition so it can be used by MS-DOS. To do this the format command:

- creates a new root directory on the disk
- creates the file allocation table for the disk
- checks for bad areas on the disk
- deletes all data on the disk if the /U option is used.

MS-DOS cannot use a new disk unless it has been formatted. The basic syntax of the command is:

```
FORMAT drive:
```

If a disk drive has already been used for MS-DOS, then formatting the drive will mean that the files which were already on the disk can no longer be accessed. This is not a good idea unless we intend to re-install MS-DOS, all our applications and our data files. If we accidentally format a disk, we can recover the data providing:

- we have not written anything to the disk after reformatting it
- we did not user the /U option of the format command.

The most important option for the format command when formatting hard disks is the /S option. Disks formatted with this option can be used to boot the computer. Note that a disk that has not been formatted in this way can not be used to start up the computer. For more information on the options available for this and other commands see MS-DOS help as described in Chapter 4.

### UNFORMAT

UNFORMAT lets us restore a disk that was erased by using the FORMAT command. MS-DOS can do this for two reasons:

- The format command does not actually delete data from the disk — it simply creates a new root directory and FAT, so the data is still there even though we cannot see it.
- When we use FORMAT (without the /U option), MS-DOS makes a copy of the old root directory and FAT before replacing them. UNDELETE simply replaces the newly created root directory and FAT with the original ones.

The simple form of the syntax for this is:

```
UNFORMAT drive:
```



### Self-test

5.9 What command and option is used to create a disk which can participate in the boot process?

5.10 What option must you not use with FORMAT, if you want to be able to UNFORMAT the disk at a later stage?




### Exercise

This will require access to a hard disk with no valuable data on it. DO NOT format any hard disk without your supervisor's permission.

5.2 Either format a spare hard disk or ask your supervisor to demonstrate the process on a spare machine.

5.3 What information was reported when the formatting finished?

5.4 How big was the disk you formatted?

## BOOTING FROM THE HARD DISK

When a PC system boots, it looks for a boot record (or sector) on the boot disk containing a short 'boot program'. As we saw in Chapter 3, this 'boot program' contains information which the computer uses to tell it how and where to load the rest of the operating system. Only disks formatted with the /S option have this 'boot program' installed.

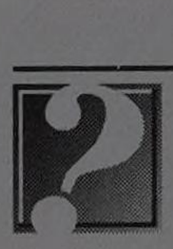
The PC will normally look first at the A: drive (which is a floppy drive) for this boot record. If it cannot find a boot record there, it will look at C: drive — the first hard disk drive on the system. Normally this is where it will find the boot record. In some modern PCs the BIOS can be configured to look on the C: drive first.

The boot record tells the computer to:

- run the BIOS programs IO.SYS and MSDOS.SYS
- load the file COMMAND.COM
- execute the commands in CONFIG.SYS
- execute the commands in AUTOEXEC.BAT.

As the latter two files are executed during the boot process, it allows us to customise the operating system to the installed hardware. All of these files must be in the root directory of the boot drive.

External commands such as ATTRIB, FDISK, LABEL etc. are normally kept in the \DOS sub-directory of the boot drive. They can be located elsewhere, provided reference is made to their location in the PATH command in AUTOEXEC.BAT.



### Self-test

5.11 What is the purpose of the boot process?

5.12 List the essential files required in the root directory of the boot drive.




- Locate the AUTOEXEC.BAT file on your boot drive.
- Display its contents.
- Locate the CONFIG.SYS file on your boot drive.
- Display its contents.



**Exercise**

5.5 Get a printout of AUTOEXEC.BAT and CONFIG.SYS.

5.6 Identify two entries in CONFIG.SYS and explain the hardware these entries are customising.

5.7 Identify four entries in AUTOEXEC.BAT and explain how these entries customise the hardware and/or the operating system.

## DISK MANAGEMENT

Over a long period of time there are several problems which can afflict hard disk drives. These are:

- **surface degradation**, where sectors on the disk cease to store data without error. Fortunately, most modern hard disks perform automatic bad sector re-mapping.

- **fragmentation**, which occurs when files are written to and erased from a disk many times. This causes single empty sectors or small groups of empty sectors to be scattered across the disk. When a file subsequently written to the disk, it will be scattered (fragmented) across many tracks. This slows disk performance significantly.

- **data corruption**, which can occur due to power fluctuations during the write process, or operating system error. This is the worst type of error as some data is usually lost.

Disk management is the process of minimising the effect of, or correcting, the above problems. MS-DOS provides several utilities to do this.



**Self-test**

5.13 List three common hard disk problems.

5.14 Does disk fragmentation cause data loss?

5.15 Why would fragmentation slow down disk performance?

## DISK MANAGEMENT UTILITIES

The three principal disk management utilities are:

- DEFRAG
- CHKDSK
- SCANDISK.

## DEFRAG

DEFRAG examines the disk for fragmentation and, if it is significant, it will rewrite the data on the disk so that files occupy **contiguous** sectors — there will be no more small clusters of unused sectors between files.

This will result in increased disk performance. This can speed up the disk read and write processes dramatically, particularly on a disk drive which is more than about 80 per cent full.

The syntax for this command is:

```
DEFRAG drive:
```

### Example

The C: drive on a particular computer is running very slowly and is 95 per cent full. To run DEFRAG we type:

```
DEFRAG C:
```

DEFRAG will:

- analyse your disk
- show you a picture of the sector usage on your disk
- report on its condition before defragmenting it.




### Exercise

5.8 From the MS-DOS prompt type:

```
DEFRAG C:
```

Look at the picture and report, then exit DEFRAG without defragmenting.

## CHKDSK

CHKDSK checks the status of a disk and displays a status report. It can also fix disk errors. The status report shows errors found in the MS-DOS filing system, which consists of the file allocation table and directories. CHKDSK also displays a summary of disk usage.

CHKDSK will fix a number of problems but you should note the following:

- CHKDSK does not actually verify that the information in your files can be accurately read. It only checks the integrity of the file as far as DOS is concerned.
- If errors exist on the disk, CHKDSK alerts you with a message.
- The SCANDISK program is now the preferred method of fixing disk drive problems. It is more modern and more powerful than CHKDSK.

The CHKDSK syntax is:

```
CHKDSK drive:
```



**Exercise**

5.9 Run CHKDSK on drive C: and note the results.

Now run it on drive D: (if you have one) and note the results.



**Self-test**

5.16 What disk parameters does CHKDSK return information on after it runs?

5.17 What memory details does CHKDSK report?

## SCANDISK

SCANDISK is a complete:

- disk checking tool
- disk analysis tool
- disk repair tool

that checks a drive for errors, including surface degradation. It can correct most problems that it finds. It is also reversible so that any corrections that cause subsequent problems can be reversed. It is beyond the scope of this text to cover all the functions of SCANDISK. However, the utility is menu-driven and is very simple to use. The disk may be checked for errors by using the following syntax:

```
SCANDISK drive:
```

You can obtain further information on SCANDISK from the on-line help or the DOS manuals.

# CHAPTER SUMMARY

In this chapter we introduced the basics of hard disks, their use and their management. We examined the most common types of disks and looked at basic parameters such as disk capacity and access time. Formatting and labelling of hard disks was covered along with the related commands of:

- LABEL
- FDISK
- FORMAT
- UNFORMAT.

The participation of hard disks in the boot process and issues of disk management using the:

- DEFRAG
- CHKDSK
- SCANDISK

utilities were also examined.



## Review questions

5.1 At what speed do hard disks typically rotate?

5.2 What is meant by access time?

5.3 What is a 'head crash'?

5.4 What is the relationship between a sector, track and cylinder?

5.5 List the two main types of hard disks found in PCs.

5.6 What is the purpose of the BIOS with respect to disk accesses?

5.7 Why is the boot record needed on a hard drive?

5.8 What is the purpose of a disk label?

5.9 What can LABEL do that VOL cannot?

5.10 What three things are written on a disk during formatting?

5.11 What is the function of a FAT?

5.12 What is FDISK used for?

5.13 Under what conditions will UNFORMAT work?

5.14 List and briefly describe the function of the commands used for disk management.



## Review exercises

5.1 Use DEFRAG to examine and, if necessary, defragment a hard disk.

5.2 Run SCANDISK and list the facilities available.

5.3 Relabel a:
- hard disk drive
- one of your diskettes.

5.4 Use CHKDSK on the hard disks on your PC and note down the details reported.

5.5 If you are in a classroom, ask your supervisor to demonstrate the FDISK utility. Note down the size and names of the partitions on the hard disks on your PC. If you are working on your own PC, take care that you do not accidentally damage the data on your disk.

5.6 Use the on-line help to get additional information on the format command. What are the different options available for formatting the different types of disks?




## Challenge problems

5.1 A disk drive is 90 per cent full and is very slow to access. What is the most likely problem and what is the solution?

5.2 Draw a directory structure for a typical bootable hard disk and show the essential files in each directory.

5.3 The DIR command shows that the file CHAPT4.DOC is located in the current directory. However, when we try to edit the file, the message 'error opening file' is displayed. What is the likely problem and how could we recover from it?

# CHAPTER 6

# FLOPPY DISK DRIVES

Nearly all modern PCs load their operating systems from and store their data on magnetic hard disk systems. This is the most convenient and fastest method. However, there are times when removable magnetic media such as a floppy disks can be very useful. For example, if:

- the hard disk suffers data corruption and the operating system will not load
- we wish to transport data to another computer system
- we want to keep a spare copy of the data (a backup) in case the hard disk fails.

In this chapter we will look at how these needs can be satisfied and examine the features and characteristics of MS-DOS which relate to managing and using floppy disks.

At the end of this chapter you will be able to:

- describe the two most common floppy disk types and the data capacities available on each type
- describe the process of formatting and labelling a floppy disk
- create a bootable disk
- describe disk management as related to floppy disks
- use the floppy disk related commands.

# DISK BASICS

The floppy disk drive has a **read/write head** which detects the magnetic pulses of the coating when reading, or it magnetises the coating in the appropriate way when writing. It has a **spindle motor** which rotates the platter at about 360 rpm.

MS-DOS reserves the drive letters A: and B: for use by floppy drives and C: onward for hard disks and other devices.

## FLOPPY DISKS

The terms *diskette*, *floppy disk* and *flexible disk* are used interchangeably. As we have already seen, floppy disks are made of a flexible mylar substrate (hence, the term floppy) with a ferro-magnetic coating similar to that used on hard disks. There are two common coating compositions in use: one for **double density** drives and the other for **high density** drives. The magnetic properties of the two types are different, as the high density drives need to be able to store around twice as much data on the same surface area.

New disks are unformatted. This means that the magnetic material on the platter has no magnetic patterns on it at all. The formatting process divides up the floppy disk surface into **tracks** and **sectors,** in a similar way to hard disks. Like the hard disk, a sector of a floppy disk in DOS is capable of holding 512 bytes of data.

Floppy disks are commonly available in two diameters: 3.5" and 5.25".

We will now look at the important features of each type.

### The 5.25" disk

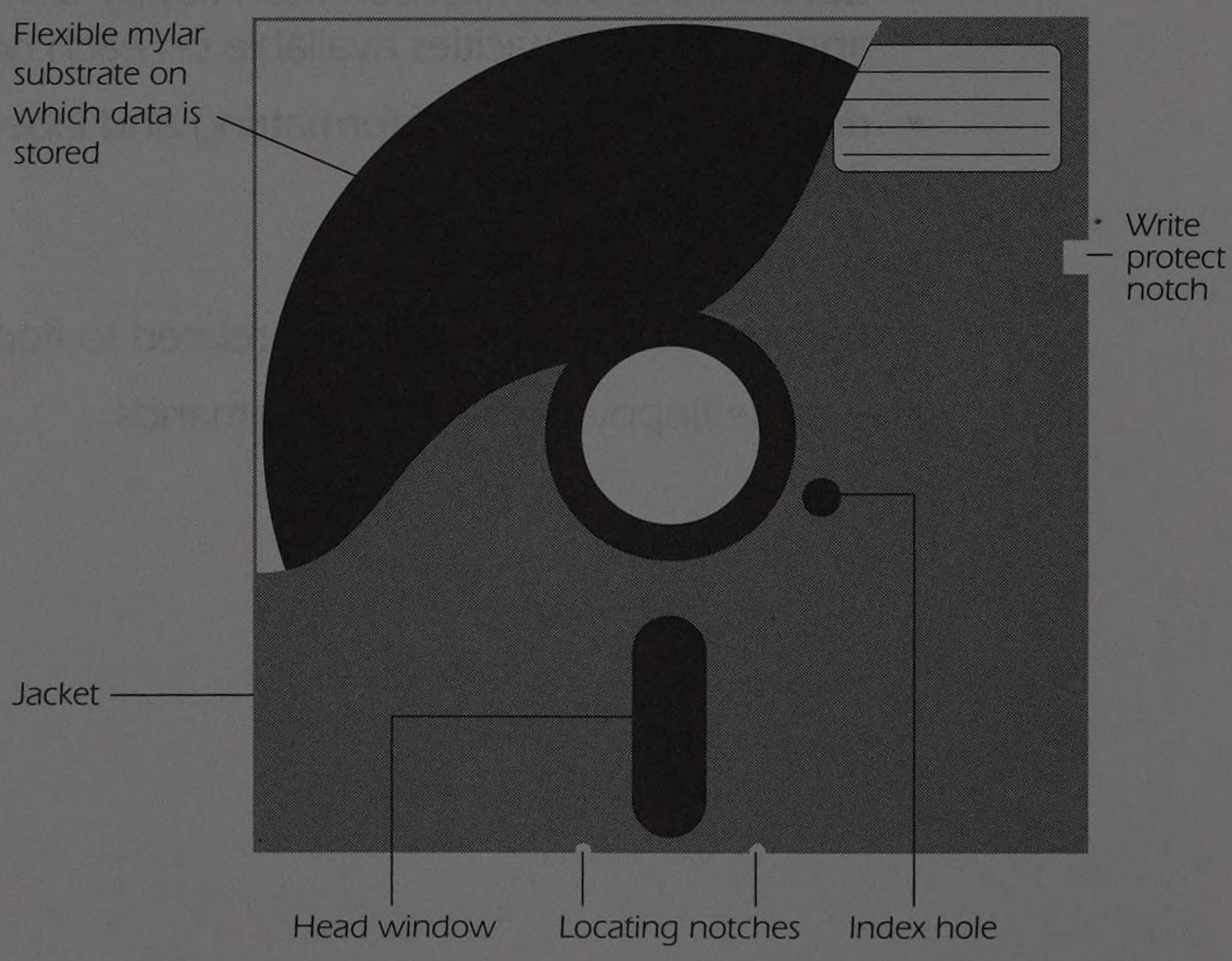



Figure 6.1 — The construction of a 5.25" disk

The main features of a 5.25" disk are that:

- the disk is enclosed in a furry plastic jacket designed to clean dust off the disk's surface as it rotates
- the disk has an index hole for timing and synchronisation purposes
- a write protect notch is used to indicate whether the disk may be used for read/write or read-only operations
- the disk has a reinforced hub ring where the drive mechanism grips it
- there are two locating notches which position the disk jacket correctly within the drive unit
- the head window is a slot in the jacket, through which the read/write heads make contact with the disk
- when formatted, the surface is divided into concentric tracks separated by **guard bands**
- the number of tracks laid down during the formatting process and the density of bits on the track determine the disk capacity.

### The 3.5" disk

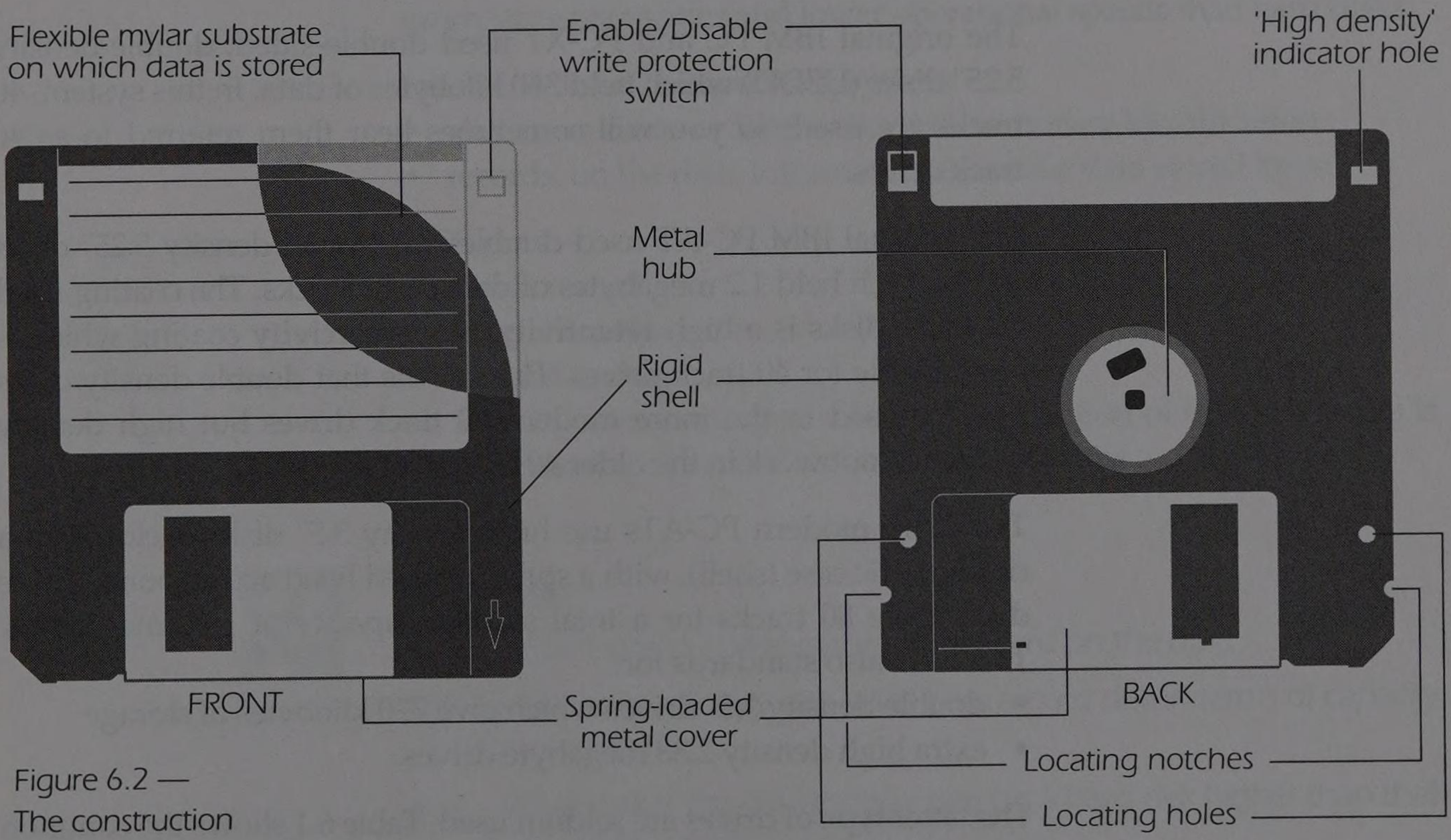



Figure 6.2 — The construction of a 3.5" disk

The main features of a 3.5" disk features are that:

- the 3.5" disk has a rigid plastic shell
- a switch in the top left-hand corner of the disk is used to enable/disable write protection
- a square hole is present in the top right-hand corner of a high density disk to distinguish it from a double density one

- the disk has a metal hub where the drive engages the disk to rotate it. This hub also provides positioning for indexing and sectoring.
- a spring-loaded metal cover conceals the head window slot when the disk is not in the drive
- two holes and two notches are provided in the hard case for locating the disk correctly within the drive unit.



### Self-test

6.1 What is the difference between double density and high density disks?




### Exercise

6.1 Obtain a 3.5" floppy disk and a 5.25" floppy disk which are no longer useful for storing data. Open the envelope or case and identify the parts described above. (If you do not have any damaged disks, your supervisor may be able to obtain some.)

## DISK CAPACITIES

The original IBM PC and PC-XT used double-sided, double-density 5.25" disks (DSDD) which held 360 kilobytes of data. In this system, 40 tracks are used, so you will sometimes hear them referred to as 40 track drives.

The original IBM PC-AT used double-sided quad-density 5.25" disks (HD) which held 1.2 megabytes of data on 80 tracks. The coating used for these disks is a high **retentivity**, high **coercivity** coating which is not suitable for 40 track drives. This means that double density disks may be used in the more modern 80 track drives but high density disks will not work in the older 40 track drives.

The more modern PC-ATs use high density 3.5" disks enclosed in a rigid plastic case (shell), with a spring-loaded head access point. These disks have 80 tracks for a total storage capacity of 1.44 megabytes. There are also standards for:

- double density 3.5" drives which give 720 kilobytes of storage
- extra high density 2.88 megabyte drives.

The latter type of drives are seldom used. Table 6.1 shows the common disk formats in use today for MS-DOS, as well as other operating systems.

Table 6.1 — Comparison of common floppy disk formats

| Disk type | Tracks | Sides | Sectors/track | Size |
|---|---|---|---|---|
| 5.25" | 40 | 2 | 8 | 320 kb |
| 5.25" | 40 | 2 | 9 | 360 kb |
| 5.25" | 80 | 2 | 15 | 1.2 Mb |
| 3.5" | 80 | 2 | 18 | 1.44 Mb |



**Self-test**

6.2 Using the data from table 6.1, what do you think would be the parameters for a 3.5" disk of 2.88 megabyte capacity?




**Exercise**

6.2 Assuming that:

- a floppy disk spins at 360 rpm
- a sector contains 512 bytes of data
- and using the information from table 6.1, calculate the theoretical data transfer rate (in kilobytes/second) for a 5.25", 1.2 megabyte disk.

## DISK ORGANISATION

As in hard disks, the process of formatting defines on the disk surface a series of concentric tracks and divides those tracks into sectors which hold a fixed number of data bytes. The only functional difference between the floppy disk and the hard disk is that floppy disks have much lower capacities and lower operational speeds than hard disks.

Like hard disks, the operating system:

- allocates a name or label to the disk for future identification
- records, on the disk, information about the data stored there.

DOS puts:

- a boot record
- a file allocation table
- a root directory

on a newly formatted disk surface. The function of these elements is the same as described in Chapter 5 for hard disks.



**Self test**

6.3 What is the purpose of the boot record on the disk?

6.4 How does a floppy disk differ from a hard disk in terms of capacity and speed?

6.5 What makes the data transfer rate on a hard disk higher than that on a floppy disk?

## LABELLING DISKS

DOS uses the same command to label floppy disks and hard disks. We will look at these commands, noting in particular options which apply to floppy disks. There are three commands which relate to labelling and two commands which relate to formatting a disk. Note that in the

case of removable media such as floppy disks, the label is the identification of the disk, not of the disk drive. Thus, reading drive A: may identify a disk labelled DOS_WORK and then by changing the disk we might see SARAS_DOCS.

## LABEL

LABEL is the MS-DOS command for labelling a disk and can also be used for reading the label of the disk. The syntax for labelling a disk is:

```
LABEL drive: disklabel
```

If we want to read the label of a disk, we type:

```
LABEL drive:
```

MS-DOS will respond with:

```
Volume in drive <drive> is  <disklabel>
Volume label (11 characters, ENTER for none)? _
```

If we do not want to re-label the disk at this point, we should press <ENTER>. MS-DOS will respond with:

```
Delete current volume label (Y/N)?_
```

to which we should respond 'N'.

### Reading the disk labels

There are two other and better ways of reading the label of a disk:

- the first is to use the DIR command which gives the volume label as well as directory information
- the second is use the VOL command.

VOL reads disk labels in the same way as hard disk labels. The syntax for reading a disk label with VOL is:

```
VOL drive:
```




### Exercise

You will need two formatted MS-DOS disks for this exercise.

6.3 Insert one disk into drive A: and read the volume name.

6.4 Write it down exactly - including case and underline characters.

6.5 Now change the label of the disk to TEST1.

6.6 Insert the second disk and read its volume name.

6.7 Now reinsert the first disk, read its volume name again and then change it back to its original name.

The commands which relate to formatting a disk are FORMAT and UNFORMAT.

## Formatting disks

FORMAT formats a disk for use by MS-DOS. The operation of this command is the same for hard disks and floppy disks. The syntax for floppy disks requires some special switches:

```
FORMAT drive:/option
```

There are two options for the format command which are particularly important when formatting floppy disks. These are the:

- /S option, which writes the boot record and the files IO.SYS, MSDOS.SYS and COMMAND.COM onto the disk so it can take part in the boot process.
- /F: option, which allows a particular disk format to be selected. For example, 360 kilobytes or 1.2 megabytes on a 5.25" disk. If no /F option is specified, the drive will attempt to format the disk to the highest data density it can.

Let us look at examples of how these switches are used.

### Example

The command to format a 720 kilobyte disk in a 3.5" high density drive is:

```
FORMAT A: /F:720
```

Note that the /F: option requires that the size of disk to be formatted is specified in kilobytes, so a 1.44 megabyte disk formatted in a 2.88 megabyte drive would require the command:

```
FORMAT A: /F:1440
```

### Unformatting disks

UNFORMAT lets us restore a disk that was erased by using the FORMAT command. This command also uses the same mechanism as that for hard disks.

The simple form of the syntax is:

```
UNFORMAT drive:
```



**Self test**

6.6 What command and option is used to create a disk which can participate in the boot process?

6.7 Under what circumstances is it not possible to UNFORMAT a disk?

## CREATING BOOTABLE DISKS

As we have seen, when a PC system boots, it looks for a boot record on the boot drive. Only disks formatted with the /S option have this program.

The PC will normally look first at the A: drive (which is a floppy drive) and if it cannot find a boot record there, it will look at the drive C: — the first hard drive. Normally this is where it will find the boot record. If the hard disk has no operating system on it, or has been corrupted for some reason, we have two options:

1. Obtain an original copy of the MS-DOS installation disks and re-install MS-DOS by booting from the first installation disk inserted in the A: drive. This is the preferred option.

2. If no installation disks are available, we can start the computer by inserting a floppy disk into the A: drive which has been formatted with the /S option.

If we need to choose the second option, the computer will start with the A: drive as the default drive and will look in the A:\DOS directory for its external MS-DOS commands. A disk which has been formatted with the /S option and has had nothing else done to it will have no \DOS directory, so while the computer will start up, we will have access only to the DOS internal commands such as:

- CD
- COPY
- MD
- RD
- PATH
- TYPE
- DIR.

These commands form part of COMMAND.COM.

The simple answer to this problem would appear to be create a \DOS directory on the bootable floppy disk and copy all the MS-DOS external commands into this \DOS directory. Unfortunately, MS-DOS is now so large that this is not practical. The best we can do is to copy a selection of the most important commands onto the A: drive. Some recommended commands are:

- ATTRIB.EXE
- CHKDISK.EXE
- SCANDISK.EXE
- SCANDISK.INI
- HELP.COM
- HELP.HLP
- QBASIC.EXE
- MOVE.EXE
- FDISK.EXE
- FORMAT.COM
- UNDELETE.EXE
- UNFORMAT.COM.

These commands will help in examining and rebuilding a faulty C: disk drive. They will allow us to perform the basic operations on the system. If the \DOS directory in C: drive is not corrupted, the PATH command may be used after boot up to access the MS-DOS external commands from that (C:) drive.

### Example

To use the external commands from the C: drive, after booting from the A: drive, type in the command:

```
PATH C:\DOS
```

To summarise, creating a bootable floppy disk requires that we:

- format the disk using the command FORMAT A: /S
- create a \DOS directory on the floppy disk (from the A:> prompt, this command would be 'MD DOS')
- copy the commands recommended above to the A:\DOS directory.



**Self test**

6.8 List the essential files required in the boot drive's root directory.

# DISK MANAGEMENT

The same problems which affect hard disks also afflict floppy disks. These problems are:

- surface degradation
- fragmentation
- data corruption.

Surface degradation and data corruption tend to be more significant on floppy disks because of the less precise nature of the media. Fragmentation is less important as far fewer files may be stored on a floppy disk compared to a hard disk.

The same utilities which are available for the management of hard disks also function well on floppy disks. These are:

- DEFRAG
- CHKDSK
- SCANDISK.

For a description of their syntax, refer back to Chapter 5.

# CHAPTER SUMMARY

In this chapter we introduced the basics of floppy disks, their use and management. We examined the two most common types of disks: the 3.5" and the 5.25". We also looked at double and high density drives. Formatting and labelling of floppy disks were also covered along with the related commands:

- LABEL
- DIR
- VOL
- FORMAT
- UNFORMAT.

Finally, we examined the creation of a bootable floppy disk and looked at issues of floppy disk management.



## Review questions

6.1 At what speed do floppy disks rotate?

6.2 What are the two common types of floppy disk available?

6.3 What capacities of floppy disk are commonly available?

6.4 What capacity is a high density 3.5" disk?

6.5 How is write protection controlled on a 3.5" disk?

6.6 List three ways we may read a label on a disk.

6.7 What three things are written on a disk during formatting?

6.8 What does the /S option do when formatting?

6.9 What is required to format a 360 kb floppy disk in a high density drive?

6.10 What are the commands used for disk management?



## Review exercises

6.1 Create a boot floppy containing the recommended commands.

6.2 Re-boot your computer using your boot floppy. Check that the main commands work, then re-boot the machine using your hard disk as normal.

6.3 Run SCANDISK and list the facilities available.

6.4 Relabel a disk drive and use VOL to check it.




## Challenge problems

6.1 Can a 720 kb formatted disk be used in a high density drive as a boot disk? Format one and try it.

6.2 Draw a directory structure for a typical bootable floppy disk and show the essential and recommended files (including hidden files) in each directory.

6.3 When booting from the C: drive the message 'COMMAND.COM not found' appears. What is the likely problem and how could we recover from it?

# ANSWERS TO SELF-TESTS

## Chapter 1

1.1 A computer is a device designed to process and move data in order to produce a desired output under the control of a program or sequence of instructions.

1.2 Engine management systems, washing machines, intelligent lifts, automated teller machines.

1.3 Keyboard, mouse, trackball.

1.4 Laser.

1.5 Dot matrix is an impact printer, laser and inkjet printers are non-impact.

1.6 Eight bits make up a byte.

1.7 The main purpose of primary storage is to hold the operating system, executable programs such as application packages, and data which the programs manipulate in order to produce the desired results.

1.8 Non-volatile memory is memory which does not lose the information stored in it when the power to the computer is turned off.

1.9 The main purpose of secondary storage is to hold data and programs which you want to keep for later use.

1.10 The three main elements of a CPU are: registers, control unit and the ALU.

1.11 The operating system is vital to the operation of a computer. It is the program which allows the user to control the computer's hardware and to run software.

## Chapter 2

2.1 The two major functions of the operating system are:
- controlling the computer's operations
- file management.

2.2 The difference between an internal and an external command is that an internal command is a short sub-program which is part of COMMAND.COM while external commands are utilities stored in the DOS directory of the disk drive.

2.3 The first thing the BIOS does during the boot process is to run diagnostic tests on the memory, disk and keyboard.

2.4 The difference between a cold and warm boot is that:

- a cold boot occurs when the computer is first turned on
- a warm boot occurs when we re-boot the computer after it has been running and without turning off the power.

2.5 The main features of MS-DOS include:

- memory configuration and management
- file management
- application program execution
- printing
- text file creation
- peripheral and hardware control
- automatic optimisation of memory usage
- recovery of accidentally deleted files
- anti-virus utilities
- data compression for disk storage
- automatic backup facilities.

Any six of these would be acceptable in answer to this question.

2.6 Applications programs communicate with computer hardware by 'calling' the parts of the operating system which in turn drive the hardware directly.

## Chapter 3

3.1 The maximum length for a filename is eight characters and the minimum is one.

3.2 The maximum length for an extension is three characters and the minimum is zero.

3.3 I would expect the files to be:

| | |
|---|---|
| CHESS.EXE | Computer chess program |
| RESUME.DOC | Someone's resume created by a word processor |
| PAYROLL.INI | An initialisation file for a payroll program |
| MEMO9410.DOC | Memo created by word processor in October 1994 |
| DIARY.TXT | Diary entries created by text editor - no control codes embedded |
| COUNTRY.SYS | Operating system file (containing country information) |
| FLGHTSIM.EXE | Flight simulator program |
| SALES94.DAT | Data file of sales figures for 1994 |
| GAMES.HLP | Text file used by a help program on games |

3.4 MARK is the parent directory and WORK is the child.

3.5 The child sub-directories of the WINDOWS directory are SYSTEM, MSAPPS and CLIPART. MSAPPS has sub-directories of its own.

3.6

| To execute the programs: | the commands are: |
|---|---|
| MS-DOS command FDISK | FDISK |
| program PAYROLL.EXE in MYAPPS: | PAYROLL |
| Flight simulator HORNET in GAMES: | C:\GAMES\HORNET |

3.7 The program REPORT.COM will not be executed if you simply type REPORT as the path does not include the directory D:\PROGS, even though the path does include its child D:\PROGS\MYAPPS.

3.8 Classification of absolute and relative paths:

| | |
|---|---|
| C:\DOS | Absolute |
| C:\DOS\FDISK.EXE | Absolute |
| DOS\FDISK.EXE | Relative |
| FDISK.EXE | Relative |
| D:\PROGS\ACCTS\PAYROLL.EXE | Absolute |
| ..\NETWORK\CHAP.1 | Relative |
| ..\..\CORRESP | Relative |
| D:\ | Absolute |

3.9 The difference between the **?** and * wildcards is that the **?** is used to match *exactly* one character, while * is used to match any number of characters.

3.10 The files matched by CHAP.* are indicated by a Y or N

| | | | |
|---|---|---|---|
| CHAP.1 | Y | CHAPS.1 | N |
| CHAPTER.123 | N | CHAP.123 | Y |
| CHAP.DOC | Y | CHAP | Y |
| CHEP.TXT | N | CHOP.XX | N |

3.11 The files matched by CH?P.??? are indicated by a Y or N

| | | | |
|---|---|---|---|
| CHAP.1 | N | CHAPS.1 | N |
| CHAPTER.123 | N | CHAP.123 | Y |
| CHAP.DOC | Y | CHAP | N |
| CHEP.TXT | Y | CHOP.XX | N |

3.12 The files matched by CH?P.* are indicated by a Y or N

| | | | |
|---|---|---|---|
| CHAP.1 | Y | CHAPS.1 | N |
| CHAPTER.123 | N | CHAP.123 | Y |
| CHAP.DOC | Y | CHAP | Y |
| CHEP.TXT | Y | CHOP.XX | Y |

## Chapter 4

4.1 On-line help is important to using MS-DOS as it lets us discover what commands are available and how to use them.

4.2 The syntax of a command is the order and spelling of words and characters used in the command.

4.3 A *parameter* is a piece of information which the command will act upon, such as a filename, while a *switch* is a command option which modifies the way in which the command will act.

4.4 The forward slash (/) identifies a switch, while a backward slash separates elements in a path, i.e. drive letters, directory names, sub-directory names and/or file names.

4.5 The two ways to enter MS-DOS help are to type HELP at the MS-DOS prompt, or type HELP *command_name* at the MS-DOS prompt. You can only use the second method if you know the name of the command you want help on.

4.6 The advantage of the first method is that you enter MS-DOS Help at a table of contents page so you can see what commands are available. The second method has the advantage of being faster if you know precisely what you want help on.

## Chapter 5

5.1 A sector is a small segment or fractional part of a track.

5.2 A track is a circular path on a disk surface on which information is written as the disk rotates.

5.3 A cylinder is a grouping of all the tracks on different platters, which can be accessed when the head mechanism is in a given position, without moving the head.

5.4 The access time of a disk is determined primarily by two things:

- the time it takes the head to move over the required track
- the delay while the disk rotates so that the required sector moves under the read/write head.

5.5 A fast disk is important to an operating system such as MS-DOS as we often refer to the disk to get parts of the operating system such as external commands. The faster the disk, the faster the operating system responds in these situations.

5.6 IDE drives communicate with the operating system through the BIOS.

5.7 The boot record sector contains a short program which tells the computer:

- how to load the operating system
- where to load the operating system from.

The file allocation table (FAT), is a table which keeps track of which sectors are in use and to which file they belong. The FAT also records any bad tracks on the disk and will prevent data being written to these bad tracks.

The **root directory** maintains a list of the files stored on the disk.

5.8 Volume labels and serial numbers must be different on different drives, otherwise MS-DOS will get confused.

5.9 FORMAT /S is the command which creates a boot disk

5.10 Do *not* use the /U switch if you want to be able to unformat the disk at a later stage.

5.11 The purpose of the boot process is to load the operating system and hence get the computer running.

5.12 The essential files required in the root directory of the boot drive are:

- IO.SYS
- MSDOS.SYS
- COMMAND.COM

5.13 Three common hard disk problems are:

- surface degradation
- fragmentation
- data corruption.

5.14 Disk fragmentation does *not* cause data loss.

5.15 Fragmentation slows down disk performance as files end up distributed over several tracks. Thus, the heads need to be moved several times during the reading of a single file which increases the access time, i.e. slows down the access.

5.16 CHKDISK returns information on:

Volume name
Volume ID

Errors

- *number* of bytes in total disk space
- *number* of bytes in *number* of hidden files
- *number* of bytes in *number* of directories
- *number* of bytes in *number* of user files
- *number* of bytes available

- *number* of bytes per allocation unit
- *number* of allocation units on the disk
- *number* of available allocation units.

5.17 CHKDISK returns information on:

- total conventional memory
- amount of free conventional memory.

## Chapter 6

6.1 The difference between double and high density disks is the type of magnetic coating used in the disk surface and hence the resultant disk capacity.

6.2 It would be a 3.5" diskette: 160 tracks, 2 sides and 18 sectors per track.

6.3 The purpose of a boot record on the disk is to tell the computer how to load the operating system.

6.4 Floppy disks rotate at around one tenth the speed of hard disks and have much less capacity (somewhere between 1 per cent and 0.05 per cent of a typical hard disk!)

6.5 The data transfer rate is higher on a hard disk because of the higher rotational speed and the greater number of bits per linear distance of track.

6.6 FORMAT /S creates a bootable disk.

6.7 UNFORMAT will not work if the disk was reformatted with the /U switch or if the disk has been written to since reformatting.

6.8 The essential files required in the root directory of the boot drive are:

- IO.SYS
- MSDOS.SYS
- COMMAND.COM.

# BIBLIOGRAPHY


Curtin, D. 1993, *Application Software*, 3rd edition, Prentice Hall, Englewood Cliffs, New Jersey.

Microsoft Corporation 1992, *Microsoft MS-DOS 5.0 Getting Started and User's Reference*, Microsoft Corporation, Redmond, Washington.

Microsoft Corporation 1994, *Microsoft Windows for Workgroups 3.11*, Microsoft Corporation, Redmond, Washington.

Microsoft Corporation 1994, *Microsoft MS-DOS 6.2 Users Guide*, Microsoft Corporation, Redmond, Washington.

# INDEX


**Symbols**
.BAT 27

**A**
absolute path 48
access time 67
active drive 35
ALU 9
application 3
application program 18
ASCII 58
ATTRIB 74
AUTOEXEC.BAT 16, 35, 74
AUX 25

**B**
backup 68
bad sector map 72
basic input output system 14
binary digits 28
BIOS 14, 68, 74
bit 6, 28
block 66
boot 9, 15
  cold 15
  warm 16
boot disk 16, 88
boot process 16
boot record 69, 74
booting 74
bootstrap 15
byte 6, 28
bytes per sector 69

**C**
cache 8
CD-ROM 8, 9, 45
central processing unit 3, 9
change directory 45
CHDIR 45
child 32
CHKDSK 75, 76
CLOCK$ 25
coercivity 84
cold boot 15
command 56
  external 15
  internal 15
command line 32
  help 56, 59
  interface 18
  prompt 33
command processing 15
command syntax 37, 56, 58
COMMAND.COM 15, 16, 74, 87
COMn 25
compact disk 8
computer
  definition 2
  hardware 18
  organisation 3
  starting 15
computer-user interface 10
CON 25
CONFIG.SYS 16, 74
contiguous sectors 76
control unit 9
COPY 38
copying files 38
CPU 3, 9
creating directories 46
current directory 32
cylinder 67

**D**
data corruption 75, 90
date 70
default drive 35
DEFRAG 75, 76
DEL 38, 40
DELETE 38
deleting directories 47
deleting files 40
destination 58
device
  input 4
  output 5
DIR 38, 44, 70, 71, 86
directory 29, 30
  child 32
  contents 44
  current 32
  parent 32
  structure 32, 38
disk 28, 29
  3.5" 83
  5.25" 82
  access time 67
  capacity 67, 84
  formatting 66, 70, 72
  labelling 70
  management 75, 90
  organisation 85
  speed 67
  types 68
  unformatted 66
disk drive 14, 30
diskette 28, 82
displaying directory structure 47
displaying files 42
DOS 11
DOS prompt 33
dot matrix 5
double density 82
DRAM 8
drive 30, 56
  active 34
  default 34
dynamic RAM 8

**E**
examples page 60
extension 24, 26, 70
external commands 15

**F**
FAT 29, 69, 97
  starting entry 70
FDISK 72, 74
file 24
  attributes 70
  extension 24, 26
  loading 30
  management 14, 37
  manipulation commands 38
  name 24
  names 24
  opening 30
  reading 30
  saving 29
  storage 28
file allocation table 29, 69
filename 56, 70
  invalid 25
  valid 25
fixed disk 68
flexible disk 82
floppy disk 9, 82
floppy disk formats 84
FORMAT 72, 87
formatting 69, 70
fragmentation 75, 90
front-end processor 2

**G**
getting help 60, 61
guard bands 83
GUI 11

**H**
hard disk 9, 66
  booting 74
hardware 3, 18
head crash 67
high density 82

**I**
IDE 68, 72
  controller 68
  interface 68
input 3
input-output 7
input-output management 14
internal commands 15
IO.SYS 16, 74, 87

**J**
jump 62

**K**
keyboard 4, 14

**L**
LABEL 70, 74, 86
label 85
  reading 86
labelling 70
loading 30
LPTn 25

**M**
magnetic tape 9
mainframe 2
manipulating directories 45
MD 45, 46
memory 6
microcomputer 2, 3
microprocessor 9
Microsoft DOS 11
mini-computer 2
MKDIR 45, 46
MORE 43
mouse 4
MS-DOS 11, 18
  facilities 18
  features 18
MS-DOS Help 56, 59
MSDOS.SYS 16, 74, 87
multi-user 10

**N**
name 24
network 10
non-volatile memory 7
notes page 60
NUL 25

**O**
on-line help 56
opening 30
operating system 3, 6, 10, 14, 18
  structure 14
  version 19
options 56
output 3

**P**
parameter 56
parent 32
parent directory 48
partition 72
PATH 35, 36
path 33, 56
pathnames
  absolute 48
  relative 48
PC 3
PC keyboard 4
personal computer 3
physical disk 72
pipe operator 43
placeholder 56
platter 66, 67, 68
  fixed 68
  removable 68
primary storage 6
PRINT 38
printer 5
  impact 5
  inkjet 6
  laser 6
  non-impact 5
printing files 44
PRN 25
program
  execution 19
prompt 17

**Q**
QWERTY 4

**R**
RAM 7, 8, 16
RD 45, 47
read 14
read-only memory 7
read/write head 29, 66, 67, 82
read/write memory 7
recovering deleted files 40
relative path 48
REN 41
RENAME 38, 41
renaming files 41
retentivity 84
RMDIR 45, 47
ROM 7
root directory 31, 69, 70, 97

**S**
save 29
SCANDISK 75, 77
SCSI 68, 72
  interface 68
secondary storage 8
sector 28, 66, 82
separator 24
SIMS 8
single in-line module 8
single user 10
Small Computer Systems Interface 68
software 3
source 58
spindle motor 66, 82
storage 3, 6, 67
  primary 6
  secondary 6, 8
string 58
sub-directory 32, 33
supercomputer 2
surface degradation 75, 90
switch 56
syntax 37, 56
syntax page 60
system disk 16
system programs 16

**T**
time 70
track 66, 82
trackball 4, 5
TREE 45, 47
TYPE 38, 42

**U**
UNDELETE 38, 40, 73
UNFORMAT 72, 73, 87, 88
utilities 15

**V**
VDU 5, 14
voice coil 67
VOL 70, 71, 86
volatile memory 7

**W**
warm boot 16
wildcard
  asterisk (*) 50
  question mark (?) 50
write 14

**The Information Technology Series** is a self-paced, modular approach to computing. The series covers a broad range of fundamentals and theory of hardware and software, as well as various application packages. The series is integrated, with more advanced modules building and expanding on previous modules.

The series is ideal for TAFE, university or industry training programs. It will also be of great use to anyone interested in learning computer basics for business, office and home.

**Renzo E. D'Orazio**, the series editor, has over 20 years teaching and industrial experience in computing and business. He has written and published several texts and modules in program design, Pascal, COBOL, database systems and the UNIX operating system. He has also been involved in course development in a wide range of subject areas in Western Australia and in curriculum development at a national level.

**Introducing DOS** shows how to use a command line user interface operating system (DOS) to run software and manage fields stored on secondary storage on a personal computer.

**Mark Stevens** has a background in electronic engineering and has previously worked in the British defence industry. He has been involved in course development nationally and his interests include microprocessors, computer networks, peripherals, business computing and microcomputer applications.

Also in the series:

- *Introducing WordPerfect 6 for Windows*
- *Introducing Access 2*
- *PC User Fundamentals*
- *Introducing Windows*
- *Using Computer Networks*
- *Introducing Word 6 for Windows*
- *Introducing Lotus 123*
- *Introducing dBASE IV*
- *Computer Administration*
- *Programming Concepts 1*
- *Programming Concepts 2*
- *3GL Program Design*
- *Introducing Excel 5*
- *Computer System Basics*

ISBN 0-471-33648-3

JOHN WILEY & SONS

9 780471 336488

KB-567-494