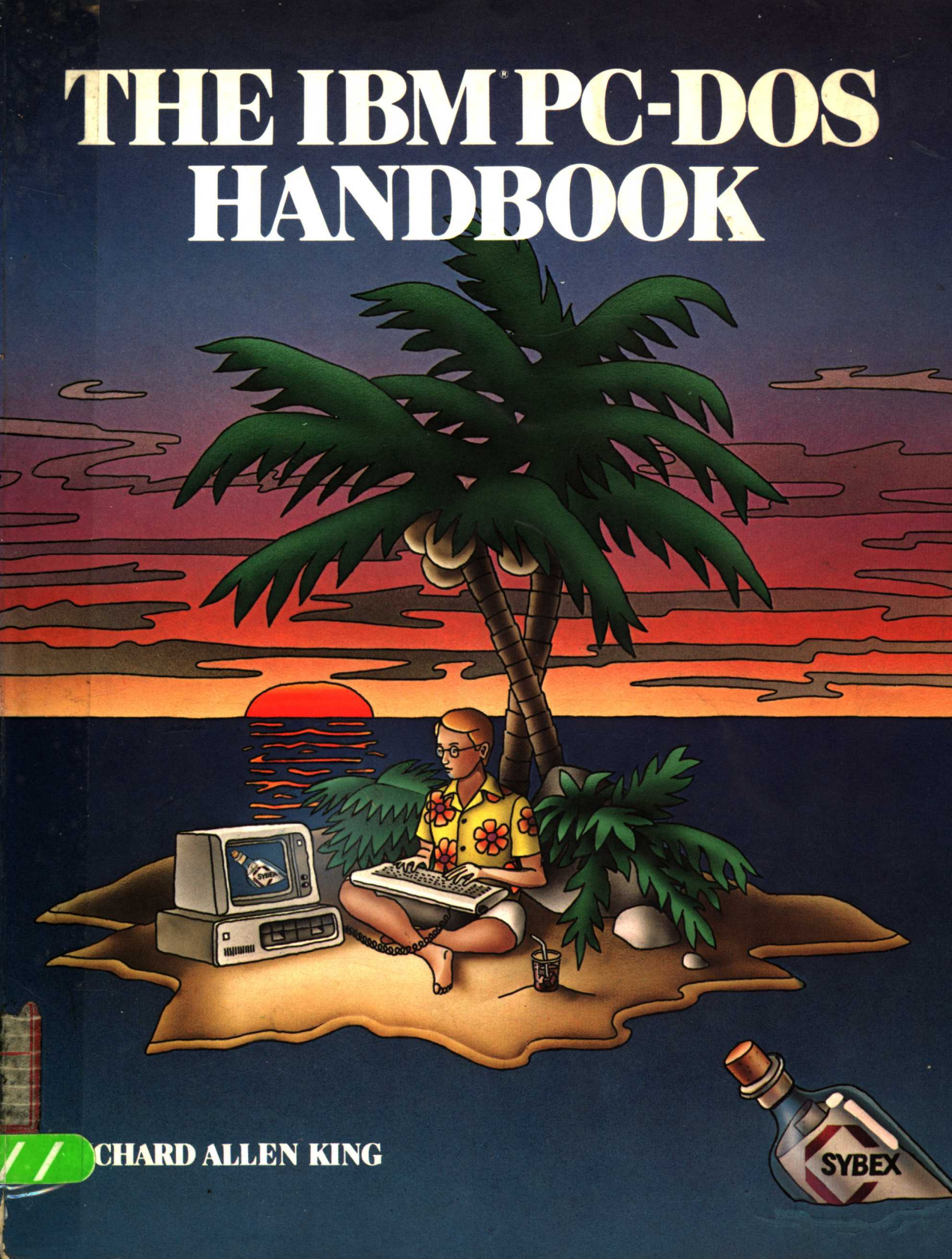
THE IBM® PC-DOS HANDBOOK



RICHARD ALLEN KING

The IBM® PC-DOS Handbook

Richard Allen King

Berkeley · Paris · Düsseldorf SYBEX®

Cover art by David Gauger
Art by Gaynelle B. Grover
Layout by van Genderen Studio

Altos is a registered trademark of Altos Computer Systems.
CP/M and CP/M-86 are registered trademarks of Digital Research, Inc.
IBM is a registered trademark of International Business Machines Corporation.
MS and XENIX are registered trademarks of Microsoft Corporation.
NorthStar is a trademark of North Star Computers, Inc.
Sup'R'Mod is a registered trademark of M & R Enterprises.
The Quantum System is a trademark of Quantum Software Systems, Inc.
UNIX is a trademark of Bell Laboratories.
VisiCalc is a registered trademark of VisiCorp, Inc.
WordStar is a registered trademark of MicroPro International Inc.
Victor is a registered trademark of Victor Technologies Inc.
6502 is a registered trademark of MOS Technology.
86-DOS is a registered trademark of Seattle Computer Products inc.

Sybex is not affiliated with any manufacturer.

Every effort has been made to supply complete and accurate information. However, Sybex assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties which would result.

©1983 SYBEX Inc. 2344 Sixth Street, Berkeley, CA 94710. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

Library of Congress Card Number: 83-61387
ISBN 0-89588-103-9
First Edition 1983
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Acknowledgements

I would like to thank Dr. Rudolph S. Langer, who suggested the subject of this book and from whom I learned most of my better programming practices (the others are mine); Ron Shafer, of AAO Computer Services Inc., for assistance in providing access to MS-DOS; Bob Good, for reviewing the manuscript and providing helpful suggestions; Salley Oberlin and Barbara Gordon, for their valuable editorial improvements, good humor, and courier services; and my wife Janet, whose support has been unflagging and generous.

Richard Allen King

June 1983

Introduction

This is a book written for people who know a little about computers. It assumes that you are using an IBM Personal Computer with the PC-DOS operating system, and that you need a rapid way of knowing more about some of the tools and features you have to work with.

If you intend to write programs that will run within PC-DOS (and MS-DOS), then the first half of the book, Chapters 1 through 8, is for you. It shows you the bare necessities of programming the 8086/8088. Then it shows you the functions your program can call within PC-DOS, along with the structures (files and programs) and devices (keyboard, screens, disks, and serial and parallel ports) on which the functions operate. Throughout the first eight chapters, where ROM BIOS (the basic input/output system installed within the IBM PC) provides better service than PC-DOS, the ROM BIOS function calls are described as well. Reading the first half of the book gives a programmer, both first-time and experienced, the background to proceed with confidence.

If you want to become adept at using PC-DOS at the command level, the second half of the book, Chapters 9 through 16, gives you quick access to the features and capabilities of this operating system. For instance, in the section on using the keyboard in Chapter 9, you will see how to use a single key (F3) to recall the last command, instead of having to retype it. The user-oriented half of the book also shows how to use batch operations, how to set up a filing system, the differences you need to be aware of between the monochrome and color/graphics capabilities, how to redirect output to a parallel printer, and a variety of other useful information.

A number of tables and maps exist that have been designed to give an "at a glance" view of important elements of the system. These tables appear within the text wherever necessary, but to save hunting around for the relevant illustration at a later date, any major map or table has been reproduced in the tables and maps chapter—Chapter 17. Much useful information has been transferred to the tables, so riffling through different manuals for that useful table you once noticed possibly has been reduced to looking in Chapter 17.

You are encouraged to skip the whole technical/programming (first) half

此为试读,需要完整PDF请访问: www.ertongbook.com

of the book with no qualms at all, if that suits your purpose. The second half, Chapters 9 through 16, covers a lot of what you need to know to make PC-DOS work for you. You can drive a car all over North America without knowing much at all about what is under the hood.

This book is written to save you a lot of time searching around in the IBM manuals. Key features described there in obscure places are brought to light here and placed in their logical context. Nonetheless, we assume that you have available the Disk Operating System manual for the IBM PC. If you have the Technical Reference manual, so much the better. It will be useful for the more complex programming tasks that you might someday undertake, although information needed from the Technical Reference manual is usually supplied here. This book is not supposed to replace the DOS manual; it is an extension and clarification of it. For instance, the basic DOS commands, CHKDSK through TYPE (ASSIGN through VOL in version 2.0), are described well in the DOS manuals, so they are not described in detail here. Instead, the PC-DOS Command Summary, with notes on each command as appropriate, is provided in Chapter 17.

HOW TO USE THIS BOOK

If you want to find out all you can about a subject, read the chapter with the relevant title and look up all appropriate index entries. This type of organization allows the author to avoid the repetition that would occur if it were necessary to exhaustively include everything on a subject in one central location.

Once you have read this introduction and Chapter 1, feel free to move to any chapter that interests you. The book has independent chapters on each subject. It need not be read from beginning to end, as if it were a tutorial.

Wherever appropriate, references to the relevant pages of the Disk Operating System and Technical Reference manuals have been provided. The page numbers refer to the second edition (May 1982) of the version 1.10 DOS manual, and the revised edition (July 1982) of the Technical Reference manual. If you have first editions of those manuals, try looking at slightly lower page numbers if what you seek is not on the page(s) referenced. If you have the later, reorganized edition of the Technical Reference manual, only the appendix page numbers will be useful.

CONVENTIONS USED IN THE BOOK

All numbers are expressed in decimal (base 10) unless they are followed by an H, as in 21H, in which case they are hexadecimal (base 16). Numbers

that are self-evidently hexadecimal (numbers that contain letters) are not necessarily followed by an H.

The use of K when referring to memory sizes and disk capacities indicates the number 1024. So 64K, a convenient way of referring to the space addressable by 16 bits, means 65,536.

The word *disk* refers to both diskettes and the fixed disk. Where information relevant only to the fixed disk is presented, the phrase *fixed disk* or *hard disk* is used.

The name of a file in PC-DOS terminology consists of a filename of one to eight characters, or a filename (of one to eight characters), followed immediately by a period and a filename extension of one to three characters. In this book *file name* (note the space) refers to the whole name of the file, which is either *filename* without an extension, or *filename.extension*. This may seem pedantic, but the distinction is useful later on.

Where you see [2.0], it means that the immediately preceding item is available only in PC-DOS version 2.0 (or later). Versions 1 refers to both versions 1.0 and 1.1. You will see comparisons made between versions 1 and version 2.0.

We are about done with the introductory section, save to tell about the history and origins of PC-DOS, then summarize each of the chapters.

THE HISTORY OF PC-DOS

In 1980 Seattle Computer Products needed an operating system to support their board-level 8086 microprocessor for S-100 systems. The ancestor of PC-DOS was born that summer, and they named it QDOS. As happens with most software, the PC-DOS we now know and love has come a long way from the original operating system, although it owes its basic structure to that dim past progenitor.

Tim Paterson is the programmer who wrote QDOS. Being a programmer of sensitivity and humor, he called it QDOS because that's what it was—a quick-and-dirty operating system. Quick and dirty are characteristics of the origins of a few fine (and many more not so fine) products. By the end of 1980 QDOS had become 86-DOS, not only because programmers' humor is not favored by those who sell, but also because 86-DOS was significantly better than QDOS and deserved another name.

Microsoft had earlier bought rights to the 86-DOS software. When IBM decided to venture into the microcomputer business, it chose Microsoft as one of the possible developers of an operating system for its line of personal computers. Within multiple shrouds of secrecy, a team of Microsoft's programmers, using 86-DOS as a base, developed the now-revealed PC-DOS on a prototype PC. By May 1981 that team included Tim Paterson. Just

before the release of the IBM PC, Microsoft closed its grip on 86-DOS by purchasing exclusive rights to it from Seattle Computer, renaming it MS-DOS at the same time.

Tim Paterson stayed with Microsoft to the completion of version 1.1, did a little planning for version 2.0, then returned to Seattle Computer in April 1982.

Now that you know the origins of PC-DOS, please do not leap to the phone to call Microsoft with your queries and suggestions. IBM bought the operating system from Microsoft, and has taken it over as a product of its own. It may be that IBM can alter and expand PC-DOS as it wishes, with no obligation to let Microsoft know of the changes. IBM is now solely responsible for all of PC-DOS. However, so far (up to the release of version 2.0) Microsoft says that as far as the applications software interface (function calls) is concerned, PC-DOS and MS-DOS are the same.

The valuable information behind all this history is that Tim Paterson, the creator of 86-DOS, chose his operating system interface conventions to match those used in CP/M. Independent software builders had by that time written a large variety of applications to run under CP/M. It was important to Seattle Computer Products' marketing of its 8086 card that the predominant suppliers of existing software written for CP/M would find it relatively easy to take advantage of the 8086's superior capabilities.

Perhaps the most rewarding aspect of Tim Paterson's work, for himself as well as for us, was that he was able to provide an operating system freed of the limitations inherent in CP/M, and that he got some of his operating-system-capabilities wish list taken care of. The result is a system that is a marked improvement over CP/M—competent instead of irritating in its handling of error conditions, friendly to the user, and now, with version 2, well on its way to becoming an all-purpose, sophisticated operating system.

SUMMARIES OF THE CHAPTERS

Chapter 1 is an introduction to operating systems in general and to PC-DOS in particular. The capabilities of PC-DOS are described from the point of view of the programmer and applications user. Relevant aspects of the 8086/8088 architecture are introduced first. This is followed by an overview of the utility routines that PC-DOS makes available to programs via function calls. The software components that make up PC-DOS are also discussed.

Chapter 2 discusses files in PC-DOS: their structures and capacities, the tools available for file creation and deletion, and data entry and manipulation. The tree-structured file access mechanisms of version 2.0 are

discussed, as well as the file control block file access strategy of versions 1. Chapter 2 also covers the use of file allocation tables for the allocation of disk space to files.

Chapter 3 gives a thorough description of the use of disk space in PC-DOS, including how the disk is organized into numbered sectors, what the capacities of the variously formatted disks are, a description of the disk access functions, and how the first few sectors of a disk are used for holding the PC-DOS software components.

Chapter 4 talks about the issues involved in getting single characters and strings of characters from the keyboard. It shows you how to make use of the keyboard access functions provided by PC-DOS. The key reassignment feature of version 2.0 is also described.

Chapter 5 introduces the two different display types (monochrome and color/graphics) available with the IBM PC. It describes how to use both the PC-DOS and the ROM BIOS functions to address them, and the differences between them. The screen control sequences provided in version 2.0 are also described.

Chapter 6 introduces the serial and parallel port handling capabilities; they exist as function calls within PC-DOS and ROM BIOS, and as part of the MODE command.

Chapter 7 is about dates and times in PC-DOS: their different representations, the function call used to access them, and how PC-DOS and ROM BIOS maintain the time and date.

Chapter 8 is about version 2.0 of PC-DOS. It discusses version 2.0 from a programmer's point of view. Version 2.0 function calls are described in each chapter as appropriate, but the new features of version 2.0 that do not have a home in other chapters of the book are shown here.

Chapter 9 begins the second half of the book with a discussion of PC-DOS at the user's level. Chapter 9 is a grab-bag of useful introductory information. Included is a brief comparison of PC-DOS with CP/M, a note on what parts of the DOS manual are most important to read, how PC-DOS makes use of the IBM PC's keyboard, how PC-DOS handles disk errors, and a note on the expanded capabilities of version 2.0.

Chapter 10 is a quick look at EDLIN, the PC-DOS editor with characteristics that endear it to so few.

Chapter 11 talks about files from a user's point of view. It discusses how to set up a filing system and how to choose file names. It also talks about file capacities and the tree-structured filing system available in version 2.0.

Chapter 12 is all about batch operations, i.e., how to use the PC's batch capability to save you much repetition and to have the PC perform sequences of commands without your intervention.

Chapter 13 gives the user's view of the two types of displays: monochrome and color/graphics. It discusses the differences between them,

color selection on the color/graphics display, and how well programs written for one type of display adaptor will run on the other.

Chapter 14 shows you how to use the serial port to redirect output from the parallel printer port to letter-quality printers attached to the serial port, and how to connect with information retrieval services.

Chapter 15 shows you how to use DEBUG: what it can do for you, how to patch a program using DEBUG, and how to use it to look around PC-DOS.

Chapter 16 tells you all about the new features introduced with version 2.0: pipes and filters, I/O redirection, hard disk support, print spooling, and much more.

Chapter 17 is the tables and maps chapter. In this chapter you will find most of the tables, figures, memory maps, and command summaries that you may need as you use PC-DOS.

Appendix A discusses the differences between MS-DOS and PC-DOS. Appendix B talks about the differences between PC-DOS versions 1.0, 1.05, and 1.1. Appendix C shows you how to create short programs using the debugger. This might save you buying the Macro Assembler.

Table of Contents

Acknowledgements xiv
Introduction xv

xvi *How to Use this Book*
xvi *Conventions Used in the Book*
xvii *The History of PC-DOS*
xviii *Summaries of the Chapters*

1 1

PC-DOS—A First View

1 *Introduction*
2 *Operating Systems*
2 *PC-DOS Fundamentals*
4 *PC-DOS and ROM BIOS*
5 *Features of the 8086/8088*
8 *Function Calls in PC-DOS*
15 *A Few More Notes*
16 *The Components of PC-DOS*
17 *Summary*

2 19

Files in PC-DOS

19 *Introduction*
20 *An Outline of Files in PC-DOS*
21 *An Overview of Versions 1 File Handling and Data Access*
21 *The Structure of a PC-DOS File—Versions 1*

此为试读,需要完整PDF请访问: www.ertongbook.com

22 *File Control Blocks*
25 *Accessing a File and Its Data in PC-DOS Versions 1*
27 *Working with a Single File*
33 *Finding, Creating, and Deleting Files—Versions 1*
36 *Versions 1 Utility Functions*
38 *Version 2.0 File Handling and Data Access*
41 *Version 2.0 File Handling Functions*
41 *Whole File Access—Version 2.0*
43 *Reading and Writing Data—Version 2.0*
45 *Trading File Handles*
46 *Finding Files in a Directory—Version 2.0*
47 *Making, Removing, and Changing Directories*
48 *Version 2.0 Utility Functions*
49 *Allocation of Disk Space to a PC-DOS File*
51 *Summary*

3 53
Disk Description and Usage

53 *Introduction*
53 *Diskette Sector Layout in PC-DOS*
55 *Disk Sector Numbering*
57 *Disk Maps, or What's Where on a PC-DOS Diskette*
60 *Diskette Capacities*
62 *Disk Access Mechanisms*
64 *Programming for Nonstandard Disk Drives*
65 *Summary*

4 67
PC-DOS and the Keyboard

67 *Introduction*
68 *Ctrl Break Processing*
68 *Keyboard Status Sampling*
68 *To Echo or Not to Echo*
69 *To Wait or Not to Wait*
70 *Buffered Input of Character Strings*
70 *Dumping Unwanted Keystrokes*

71 *Standard Interpretation of Nontext Keystrokes*
72 *Changing the Meaning of the Keys*
79 *Single-Character Input Functions*
80 *Summary of the Keyboard Access Functions*

5 83
Output to the Monochrome and Color/Graphics Monitors

83 *Introduction*
84 *The PC-DOS Function Calls That Affect the Screen*
85 *Character Attributes*
89 *Version 2.0's Extended Screen Control*
93 *Video I/O Using the ROM BIOS Function Calls*
100 *Switching Output between Monochrome and Color/Graphics*
102 *The Different Display, Different Software Question*
103 *Locations of the Characters and Their Attribute Bytes*
103 *Summary*

6 105
The Serial and Parallel Ports

105 *Introduction*
106 *PC-DOS Serial Port Function Calls*
106 *ROM BIOS Serial I/O Function Calls*
110 *Serial I/O with Version 2.0*
110 *The Hazards of Programming for RS-232 Communications*
111 *The Parallel Printer Interface*
113 *Summary*

7 115
Dates and Times in PC-DOS

115 *Introduction*
115 *Date Formats in PC-DOS*
117 *Time Formats in PC-DOS*
119 *ROM BIOS Time*
122 *Total Time Control*
122 *Summary*

8 125

Version 2.0 at the Systems Level

125 *Introduction*
125 *What's New with Version 2*
126 *New File Access Strategy*
126 *New Function Call Error Return Reporting*
126 *Some Loose Ends Cleared Up*
126 *The Break Function—Function 33H*
127 *Memory Management—Functions 48H, 49H, and 4AH*
128 *The Execute Function*
133 *Summary*

9 135

A User-Level Introduction to PC-DOS

135 *Introduction*
136 *A Brief Description of PC-DOS Capacities*
136 *The Difference between PC-DOS and CP/M*
137 *PC-DOS and the PC'S Keyboard*
140 *PC-DOS and Disk Errors*
140 *What's Where in the DOS Manual*
143 *The Expanded Capabilities of Version 2.0*
143 *Summary*

10 145

A Quick Look at EDLIN

145 *Introduction*
146 *Line Numbers in EDLIN*
146 *The EDLIN Commands*
148 *Creating a File Using EDLIN*
148 *The List Command*
149 *The Page Command [2.0]*
149 *The Insert Command*
150 *The Edit Command*
151 *The Search and Replace Commands*

152 *The Append and Write Commands*
153 *The Copy Command [2.0]*
153 *The Move Command [2.0]*
154 *The Transfer Command [2.0]*
154 *EDLIN Version 2.0 Features*
156 *Ending an Editing Session*
156 *EDLIN's Treatment of .$$$ and .BAK Files*

11 161
Your PC-DOS Files

161 *Introduction*
161 *Files in PC-DOS Versions 1.0 and 1.1*
172 *Files in PC-DOS Version 2.0*
178 *Summary of User-Level File Management in PC-DOS*

12 181
Batch Operations, or Bye Bye Repetition

181 *Introduction*
181 *Applications for Batch Files*
182 *How Batch Files Work*
183 *Creating a Batch File*
186 *Bigger, Better Batch Files with Version 2.0*
191 *A Note or Two about Batch Files*

13 193
The Monochrome and Color/Graphics Screens

193 *Introduction*
193 *The Monochrome Display*
194 *The Color/Graphics Capability*
196 *Color Selection with the Color/Graphics Adaptor*
200 *A Small Puzzle for Software Writers*
200 *Choosing a Monitor*
201 *Mixing and Matching*
201 *Summary*

14 203

Using the Serial Port

203 *Introduction*
203 *PC-DOS Communications Capabilities*
205 *Using Two Letter-Quality Printers with Serial Interfaces*
206 *Potential Problems When Using the Serial Port*
209 *Connection with Information Retrieval Services*
211 *Summary*

15 213

The Delights of DEBUG

213 *Introduction*
213 *Why "DEBUG"?*
214 *What DEBUG Can Do for Us*
215 *How DEBUG Works*
216 *How to Use DEBUG*
220 *Patching with DEBUG*
223 *Exploring with DEBUG*
225 *Summary*

16 227

A User-Level View of Version 2.0

227 *Introduction*
227 *Summary of New Features*
228 *Hard Disk Support*
228 *The Expanded Batch Facility*
229 *I/O Redirection*
230 *Pipes*
230 *Filters*
231 *Print Spooling with the Print Command*
232 *File Recovery*
233 *Graphics to the Printer*
234 *Modifications to Existing Commands*
234 *The 26 New Commands*
237 *The Configuration File*
239 *Summary*

17 241
Tables and Maps

241 *Introduction*
243 *Address Conversion Tables for Newcomers*
245 *Character Attribute Bytes*
247 *Color/Graphics Modes*
247 *Command Summaries*
252 *Diskette Information*
253 *File Attributes and Handles*
254 *Function Call Error Return Table [2.0]*
255 *Function Call Summaries*
259 *Interrupt Vectors*
262 *Memory Maps*
266 *PC-DOS Memory Overhead Summary*
267 *Reserved Names and Characters*
269 *Structure Maps*

Appendix A: 275
The Difference between MS-DOS and PC-DOS

275 *Introduction*
275 *The Universal Operating System*
276 *The Birth of PC-DOS*
278 *Conclusion*

Appendix B: 281
Differences between PC-DOS Versions 1.0, 1.05, and 1.1

281 *Introduction*
281 *Versions 1.0 and 1.05*
282 *Version 1.1*

Appendix C: 285
Creating Programs Using the Debugger

Index 290

此为试读,需要完整PDF请访问: www.ertongbook.com