CP/M 80 Programmer's Guide

Macmillan Computer Science Series

Consulting Editor

Professor F. H. Sumner, University of Manchester

S. T. Allworth, *Introduction to Real-time Software Design*
Ian O. Angell, *A Practical Introduction to Computer Graphics*
R. E. Berry and B. A. E. Meekings, *A Book on C*
G. M. Birtwistle, *Discrete Event Modelling on Simula*
T. B. Boffey, *Graph Theory in Operations Research*
Richard Bornat, *Understanding and Writing Compilers*
J. K. Buckle, *Software Configuration Management*
W. D. Burnham and A. R. Hall, *Prolog Programming and Applications*
J. C. Cluley, *Interfacing to Microprocessors*
Robert Cole, *Computer Communications*
Derek Coleman, *A Structured Programming Approach to Data*
Andrew J. T. Colin, *Fundamentals of Computer Science*
Andrew J. T. Colin, *Programming and Problem-solving in Algol 68*
S. M. Deen, *Fundamentals of Data Base Systems*
S. M. Deen, *Principles and Practice of Database Systems*
P. M. Dew and K. R. James, *Introduction to Numerical Computation in Pascal*
M. R. M. Dunsmuir and G. J. Davies, *Programming the UNIX System*
K. C. E. Gee, *Introduction to Local Area Computer Networks*
J. B. Gosling, *Design of Arithmetic Units for Digital Computers*
Roger Hutty, *Fortran for Students*
Roger Hutty, *Z80 Assembly Language Programming for Students*
Roland N. Ibbett, *The Architecture of High Performance Computers*
Patrick Jaulent, *The 68000 – Hardware and Software*
J. M. King and J. P. Pardoe, *Program Design Using JSP – A Practical Introduction*
H. Kopetz, *Software Reliability*
E. V. Krishnamurthy, *Introduction Theory of Computer Science*
V. P. Lane, *Security of Computer Based Information Systems*
Graham Lee, *From Hardware to Software – an introduction to computers*
A. M. Lister, *Fundamentals of Operating Systems, third edition*
G. P. McKeown and V. J. Rayward-Smith, *Mathematics for Computing*
Brian Meek, *Fortran, PL/I and the Algols*
Derrick Morris, *System Programming Based on the PDP11*
P. Oets, *Using MS-DOS and PC-DOS*
Christian Queinnec, *LISP*
John Race, *Case Studies in Systems Analysis*
W. P. Salman, O. Tisserand and B. Toulout, *FORTH*
L. E. Scales, *Introduction to Non-linear Optimization*
Peter S. Sell, *Expert Systems – A Practical Introduction*
Colin J. Theaker and Graham R. Brookes, *A Practical Course on Operating Systems*
M. J. Usher, *Information Theory for Information Technologists*
B. S. Walker, *Understanding Microprocessors*
Peter J. L. Wallis, *Portable Programming*
I. R. Wilson and A. M. Addyman, *A Practical Introduction to Pascal – with BS6192, second edition*

Other Macmillan titles of related interest

J. E. Bingham and G. W. P. Davies, *A Handbook of Systems Analysis, second edition*
J. E. Bingham and G. W. P. Davies, *Planning for Data Communications*

CP/M 80 Programmer's Guide

Barry Morrell

B.Sc.

Peter Whittle

B.Sc.

M
MACMILLAN

© Barry Morrell and Peter Whittle 1985

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provisions of the Copyright Act 1956 (as amended).

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

First published 1985

Published by
MACMILLAN EDUCATION LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

Typeset by TecSet Ltd, Sutton, Surrey

British Library Cataloguing in Publication Data
Morrell, Barry

CP/M 80 programmer's guide.

1. CP/M-80 (Computer operating system)

I. Title II. Whittle, Peter

001.64'25 QA76.6

ISBN 978-0-333-39558-5 ISBN 978-1-349-08123-3 (eBook)

DOI 10.1007/978-1-349-08123-3

Contents

<i>Preface</i>	<i>ix</i>
<i>Acknowledgements</i>	<i>x</i>
1 Introduction	1
What is an operating system?	1
What is CP/M 80?	2
What does CP/M 80 consist of?	3
How do you interface with CP/M 80?	4
How do your programs interface with CP/M 80?	4
Why use machine code?	6
Using high level languages under CP/M	6
Summary	6
2 Program Development	8
Design	8
Program documentation	9
Cutting code	9
Assembling, loading and running	10
Basic principles	10
Building libraries	13
Testing and debugging	14
Optimisation	14
Summary	15
3 Design	16
Information gathering	16
The main design activity	17
General points	17
Subroutine coupling	18
Program portability	19
The final stages of design	20
Summary	21
4 Using Simple BDOS Functions	22
The BDOS function call	22

Simple output using BDOS functions	24
Simple input using BDOS functions	26
A more efficient mode of input	28
Bypassing the BDOS	30
Summary	34
5 File Handling	36
Discs and files	36
Creating a simple file	38
Reading a simple file	41
Adding data to the end of a file	43
More about files	45
FCBs and their contents	45
Sectors and blocking	45
Types of file	47
Data areas	47
Passing filenames via the keyboard	49
Directory operations	51
Using random access with files	55
Random access of a sequentially-written file	56
Writing sectors in random order	58
Error handling with files	60
Summary	60
6 Disc Operations	62
Introduction	62
Protecting your discs and files	63
Getting hold of disc characteristics	64
Introduction to the disc data structures	64
The allocation maps	67
The Disc Parameter Blocks (DPBs)	68
Allocation block sizes	70
Summary	71
7 Debugging your Programs	73
General principles	73
How do bugs get into programs?	73
Keeping bugs out of your programs	74
Trapping the ones that got away!	74
Getting rid of the rest by testing	75
Debugging under CP/M	76
Loading a program (or ZSID) from disc	77
Setting breakpoints and running the code	77
Displaying the register set	78

Displaying an area of memory	78
'Patching' data	79
'Patching' a program	80
Disassembling a piece of code	80
Summary	81
8 Random Access Files	82
Creating sequential files	82
How random access files are allocated	84
Summary	86
9 Using BDOS Functions from High Level Languages	87
Accessing machine code from BASIC	88
Accessing machine code from Pascal	89
Summary	91
<i>Appendix A: Example Programs</i>	92
Reaction timer program	92
File size calculation program	92
Disc free space program	92
<i>Appendix B: CP/M Memory Map</i>	106
<i>Appendix C: ASCII Code Tables</i>	107
<i>Appendix D: BDOS Functions</i>	109
BDOS calling mechanism	109
Summary of BDOS functions	109
<i>Appendix E: BDOS Functions of CP/M 80 Family</i>	155
<i>Appendix F: The ZASM Macro Assembler</i>	158
<i>Index</i>	159

Preface

This book is for people who want to learn how to program under the CP/M 80 Operating System. It is *not* a guide to using CP/M. Before reading it you should have some knowledge of programming using Z80 machine code and programming in a high level language such as BASIC. You should also have read a book on CP/M operating such as *Using CP/M* by Peter Gosling (Macmillan, 1985).

The main part of the present book is meant to be read sequentially and, if you have little experience of machine code programming, this is the way that you should read it. If you are already an experienced programmer, you may want to 'dive' into the book at a particular point. In this case, you will find a list of keywords at the front of each chapter which will tell you what is covered within that chapter, and a more detailed summary at the end of most chapters.

The appendixes are meant to be used for reference purposes. They include descriptions of BDOS functions, laid out in alphabetical order with a separate page for each function. A list of functions grouped under appropriate headings is also given.

Examples in this book are written using Research Machines Ltd's Z80 macro assembler ZASM, mainly because it is a true Z80 assembler. However, with minor changes they can run under other assemblers such as Microsoft's M80. The main differences between ZASM and M80 are summarised in appendix F.

Acknowledgements

To reviewers and Research Machines Ltd for their help. In particular, to Research Machines for their permission to use some of the information on program portability in chapter 3.

CP/M and CP/NET are registered trademarks of Digital Research. Zilog and Z80 are trademarks of Zilog Inc.

ProPascal is a registered trademark of Prospero Software.

Microsoft BASIC is a registered trademark of Microsoft.

ZASM is a registered trademark of Research Machines Ltd.

Wordstar is a registered trademark of MicroPro International Corporation.