

大学计算机教育国外著名教材系列 (影印版)



IBM PC ASSEMBLY LANGUAGE  
AND PROGRAMMING  
FIFTH EDITION

**IBM PC汇编语言  
程序设计 (第5版)**



Peter Abel 著



清华大学出版社

# IBM PC ASSEMBLY LANGUAGE AND PROGRAMMING FIFTH EDITION

## IBM PC汇编语言程序设计(第5版)

### 内 容 简 介

本书主要介绍用汇编语言进行程序设计, 先从硬件和软件着手, 然后介绍编程所需的各种指令, 并给出丰富的范例。全书分为七大部分: PC机硬件、软件基础, 汇编语言基础, 屏幕与键盘操作, 数据操作, 高级输入/输出, 特别主题, 以及关于BIOS数据区、中断与端口、运算符与指令、PC指令系统等的参考资料。

本书既可作为计算机及相关专业汇编语言课程的教材, 也可作为开发人员的参考用书。要掌握本书内容, 不仅要认真阅读本书的每个章节, 还应使用本书提供的范例并尝试运行这些范例, 并认真完成每章后面的练习。

For sale and distribution in the People's Republic of China  
exclusively(except Taiwan, Hong Kong SAR and Macao SAR).

仅限于中华人民共和国境内 (不包括中国香港、  
澳门特别行政区和中国台湾地区) 销售发行。

ISBN 7-302-12881-2



9 787302 128816 >

定价:35.00元



<http://www.pearsoned.com>

大学计算机教育国外著名教材系列（影印版）

**IBM PC Assembly Language and Programming**

Fifth Edition

**IBM PC 汇编语言程序设计**

（第 5 版）

Peter Abel

清华大学出版社  
北京

English reprint edition copyright © 2006 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: IBM PC Assembly Language and Programming, Fifth Edition, by Peter Abel, Copyright © 2001

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall, Inc.

This edition is authorized for sale and distribution only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong, Macao SAR and Taiwan).

本书影印版由 Pearson Education (培生教育出版集团) 授权给清华大学出版社出版发行。

**For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macao SAR).**

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

北京市版权局著作权合同登记号 图字 01-2006-2136

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

IBM PC 汇编语言程序设计: 第5版 = IBM PC Assembly Language and Programming / (美) 阿贝尔 (Abel, P.) 著. 一影印本. 一北京: 清华大学出版社, 2006.5

(大学计算机教育国外著名教材系列)

ISBN 7-302-12881-2

I. I… II. 阿… III. 汇编语言—程序设计—高等学校—教材—英文 IV. TP313

中国版本图书馆 CIP 数据核字 (2006) 第 036247 号

出版者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-6277 0175

印刷者: 北京市世界知识印刷厂

装订者: 三河市化甲屯小学装订二厂

发行者: 新华书店总店北京发行所

开本: 148×210 印张: 17.75

版次: 2006年5月第5版 2006年5月第1次印刷

书号: ISBN 7-302-12881-2/TP·8189

印数: 1~3000

定价: 35.00 元

地址: 北京清华大学学研大厦

邮编: 100084

客户服务: 010-6277 6969

## 出版说明

进入 21 世纪,世界各国的经济、科技以及综合国力的竞争将更加激烈。竞争的中心无疑是对人才的竞争。谁拥有大量高素质的人才,谁就能在竞争中取得优势。高等教育,作为培养高素质人才的事业,必然受到高度重视。目前我国高等教育的教材更新较慢,为了加快教材的更新频率,教育部正在大力促进我国高校采用国外原版教材。

清华大学出版社从 1996 年开始,与国外著名出版公司合作,影印出版了“大学计算机教育丛书(影印版)”等一系列引进图书,受到国内读者的欢迎和支持。跨入 21 世纪,我们本着为我国高等教育教材建设服务的初衷,在已有的基础上,进一步扩大选题内容,改变图书开本尺寸,一如既往地请有关专家挑选适用于我国高校本科及研究生计算机教育的国外经典教材或著名教材,组成本套“大学计算机教育国外著名教材系列(影印版)”,以飨读者。深切期盼读者及时将使用本系列教材的效果和意见反馈给我们。更希望国内专家、教授积极向我们推荐国外计算机教育的优秀教材,以利我们把“大学计算机教育国外著名教材系列(影印版)”做得更好,更适合高校师生的需要。

清华大学出版社

# PREFACE

The heart of a personal computer is a microprocessor, which handles the computer's requirements for arithmetic, logic, and control. The microprocessor had its origin in the 1960s, when research designers devised the integrated circuit (IC) by combining various electronic components into a single component on a silicon "chip." In the early 1970s Intel introduced the 8008 chip, which ushered in the first generation of microprocessors.

By 1974 the 8008 had evolved into the 8080, a popular second-generation microprocessor with general-purpose use. In 1978 Intel produced the third-generation 8086 processor, which represented a significant advance in its design. The 8088, a variation of the 8086, provided a slightly simpler design and compatibility with then-current input/output devices. The 8088 was selected by IBM in 1981 for its forthcoming personal computer. Enhanced versions of the 8086 include the 80286, 80386, 80486, Pentium, and other more advanced Pentium and Celeron models, each of which provides additional processing power.

Each family of processors has its own unique set of instructions that are used to direct its operations, such as accept input from a keyboard, display data on a screen, and perform arithmetic. This set of instructions (the machine language) is too complex and obscure for use in the development of programs. Software suppliers provide an assembly language for the processor family that represents the various instructions in more understandable symbolic code.

## LEVELS OF PROGRAMMING

The levels of programming languages are the following:

- *Machine language* consists of the individual instructions that the processor executes one at a time; these are embedded in the operating system and in the low-level ready-only portions of the machine architecture.

- *Low-level assembly language* is designed for a specific family of processors; the symbolic instructions directly relate to machine language instructions one-for-one and are assembled into machine language.
- *High-level languages* such as C, C++, and Visual BASIC were designed to eliminate the technicalities of a particular computer; statements compiled in a high-level language typically generate many low-level instructions.

## ADVANTAGES OF ASSEMBLY LANGUAGE

A knowledge and use of assembly language includes these advantages:

- Shows how programs interface with the operating system, the processor, and BIOS.
- Shows how data is represented and stored in memory and on external devices.
- Clarifies how the processor accesses and executes instructions and how instructions access and process data.
- Clarifies how a program accesses external devices.

As well, reasons for using assembly language include the following:

- A program written in assembly language requires considerably less memory and execution time than one written in a high-level language.
- Assembly language gives a programmer the ability to perform highly technical tasks that would be difficult, if not impossible, in a high-level language.
- Although most software specialists develop new applications in high-level languages, which are easier to write and maintain, a common practice is to recode in assembly language those sections that are time-critical.
- Resident programs (that reside in memory while other programs execute) and interrupt service routines (that handle input and output) are almost always developed in assembly language.

The following material is required for learning PC assembly language:

- Access to an IBM personal computer (any model) or equivalent compatible.
- A copy of the Windows 95/98 or DOS operating system and familiarity with its use. It is much easier to learn the intricacies of assembly language while working within a relatively simple operating system like DOS rather than within the Windows environment. Within DOS, you can freely experiment and can later step up to the Windows environment.
- A copy of an assembler translator program. Common suppliers include Microsoft, Borland, and SLR Systems.

The following are *not* required for learning assembly language:

- Prior knowledge of a programming language, although such knowledge may help you grasp some programming concepts more readily.
- Prior knowledge of electronics or circuitry. This book provides all the information about the PC's architecture that you require for programming in assembly language.

## FOCUS OF THIS BOOK

To assist readers in learning assembly language programming, this book first covers the simpler aspects of the hardware and the language and then introduces instructions as they are needed. As well, the text emphasizes clarity in program examples. Thus the examples use those instructions and approaches that are the easiest to understand, even though a professional programmer would often solve similar problems with more sophisticated—but less clear—code.

The programs also omit macro instructions (explained in Chapter 21); although professional programmers use macros extensively, their appearance in a book of this nature would interfere with learning the principles of the language. Once you have learned these principles, you can then adopt the techniques of the professional.

## THE APPROACH TO TAKE

This book can act as both a tutorial and a reference. To make the most effective use of your investment in a PC and software, work through each chapter carefully and reread any material that is not immediately clear. Use the program examples and get them to execute (or “run”) on your computer. Also, be sure to work through the exercises at the end of each chapter.

The first eight chapters furnish the foundation material for assembly language. After studying these chapters, you can proceed with Chapters 9, 11, 12, 14, 15, 16, 20, 21, or 22. Chapters 24 through 26 are intended as references. Chapters related to each other are:

- 8 through 10 (screen and keyboard operations)
- 12 and 13 (arithmetic operations)
- 16 through 19 (disk processing)
- 22 and 23 (subprograms and overlays)

On completing this book, you will be able to:

- Understand the hardware of the personal computer.
- Understand machine-language code and hexadecimal format.
- Understand the steps involved in assembling, linking, and executing a program.
- Write programs in assembly language to handle the keyboard and screen, perform arithmetic, convert between ASCII and binary formats, perform table searches and sorts, and handle disk input and output.
- Trace machine execution as an aid in program debugging.
- Write your own macro instructions to facilitate faster coding.
- Link separately assembled programs into one executable program.

Learning assembly language and getting your programs to work is an exciting and challenging experience. For the time and effort invested, the rewards are sure to be great.



## NOTES ON THE FIFTH EDITION

This fifth edition reflects a considerable number of enhancements to the previous edition, including the following:

- More features of the Intel Pentium processors
- More program examples and exercises
- Earlier introduction to interrupt operations
- Considerable reorganization and revision of explanations throughout the text
- More material on protected mode, passing parameters, the use of the stack, addressing modes, video systems and INT 10H functions, array handling, subprograms, and ports
- Revised and additional questions at the end of each chapter.

Users of the fourth edition should note that the contents of Chapter 7 (.COM Programs) has been combined in this edition with Chapter 5. Also, Chapter 21 has been dispersed in this way: the material on mouse handling to its own chapter (15) and the rest of the chapter (ports, string I/O, and sound) combined with BIOS and program interrupts in Chapter 24.

Note to the Student/Reader: Check out the web site for answers to selected questions, programs from the book for downloading, and questions for trying out. The address is [www.prenhall.com/abel](http://www.prenhall.com/abel)

Note to the Instructor: An instructors' CD with software, additional problems, and solutions is available for adopters of the book.

## Acknowledgments

The author is grateful for the assistance and cooperation of all those who contributed suggestions for, reviews of, and corrections to earlier editions.

# CONTENTS

<b>PREFACE</b>	<b>xi</b>
<b>Part A Fundamentals of PC Hardware and Software</b>	<b>1</b>
<b>1 BASIC FEATURES OF PC HARDWARE</b>	<b>1</b>
Introduction	1
Bits and Bytes	1
The Binary Number System	3
Hexadecimal Representation	6
ASCII Code	7
PC Components	8
Internal Memory	10
Segments and Addressing	12
Registers	13
Hardware Interrupts	18
Key Points	19
Review Questions and Exercises	19
<b>2 INSTRUCTION ADDRESSING AND EXECUTION</b>	<b>22</b>
Introduction	22
Features of an Operating System	22

The BIOS Boot Process	23
The System Program Loader	24
The Stack	25
Instruction Execution and Addressing	27
Instruction Operands	29
Protected Mode	29
Key Points	30
Review Questions and Exercises	30
<b>3 EXAMINING COMPUTER MEMORY AND EXECUTING INSTRUCTIONS</b>	<b>32</b>
Introduction	32
Using the DEBUG Program	32
Viewing Memory Locations	34
Machine Language Example I: Using Immediate Data	36
Machine Language Example II: Using Defined Data	40
An Assembly Language Program	44
Using the INT Instruction	45
Using the PTR Operator	47
Key Points	48
Review Questions and Exercises	49
<b>Part B Fundamentals of Assembly Language</b>	<b>51</b>
<b>4 REQUIREMENTS FOR CODING IN ASSEMBLY LANGUAGE</b>	<b>51</b>
Introduction	51
Assembly Language Features	52
Conventional Segment Directives	58
Simplified Segment Directives	62
Initializing for Protected Mode	64
Defining Types of Data	65
Equate Directives	70
Key Points	72
Review Questions and Exercises	73
<b>5 ASSEMBLING, LINKING, AND EXECUTING PROGRAMS</b>	<b>74</b>
Introduction	74
Preparing a Program for Assembling and Execution	74
Two-Pass Assembler	81
Linking an Object Program	81
Executing a Program	83
The Cross-Reference Listing	84
Error Diagnostics	84
The Assembler Location Counter	85
Writing .COM Programs	86

Key Points 89  
 Review Questions and Exercises 90

**6 SYMBOLIC INSTRUCTIONS AND ADDRESSING 92**

Introduction 92  
 The Symbolic Instruction Set—An Overview 92  
 Data Transfer Instructions 95  
 Basic Arithmetic Instructions 97  
 Repetitive Move Operations 98  
 The INT Instruction 100  
 Addressing Modes 100  
 The Segment Override Prefix 104  
 Near and Far Addresses 105  
 Aligning Data Addresses 105  
 Key Points 106  
 Review Questions and Exercises 106

**7 PROGRAM LOGIC AND CONTROL 109**

Introduction 109  
 Short, Near, and Far Addresses 110  
 The JMP Instruction 111  
 The LOOP Instruction 113  
 The Flags Register 114  
 The CMP Instruction 115  
 Conditional Jump Instructions 116  
 Calling Procedures 119  
 The Effect of Program Execution on the Stack 121  
 Boolean Operations 125  
 Shifting Bits 128  
 Rotating Bits 131  
 Organizing a Program 133  
 Key Points 134  
 Review Questions and Exercises 135

**Part C Video and Keyboard Operations 137**

**8 INTRODUCTION TO VIDEO AND KEYBOARD PROCESSING 137**

Introduction 137  
 Screen Features 138  
 Setting the Cursor 138  
 Clearing the Screen 139  
 INT 21H Function 09H for Screen Display 139  
 INT 21H Function 0AH for Keyboard Input 141  
 INT 21H Function 02H for Screen Display 147

File Handles	148	
INT 21H Function 40H for Screen Display	148	
INT 21H Function 3FH for Keyboard Input	149	
Key Points	151	
Review Questions and Exercises	151	
<b>9 VIDEO SYSTEMS</b>		<b>153</b>
Introduction	153	
Components of a Video System	154	
Video Modes	156	
Attributes	156	
BIOS INT 10H Operations	158	
Using Graphics Mode	175	
Direct Video Display	179	
ASCII Characters for Boxes and Menus	182	
Key Points	183	
Review Questions and Exercises	184	
<b>10 KEYBOARD OPERATIONS</b>		<b>186</b>
Introduction	186	
BIOS Keyboard Data Areas	187	
INT 21H for Keyboard Input	188	
INT 16H for Keyboard Input	189	
Extended Function Keys and Scan Codes	192	
BIOS INT 09H and the Keyboard Buffer	197	
Key Points	199	
Review Questions and Exercises	201	
<b>Part D Data Manipulation</b>		<b>203</b>
<b>11 PROCESSING STRING DATA</b>		<b>203</b>
Introduction	203	
Features of String Operations	204	
MOVS: Move String Instruction	205	
LODS: Load String Instruction	207	
STOS: Store String Instruction	207	
Program: Using LODS and STOS to Edit Data	208	
CMPS: Compare String Instruction	209	
SCAS: Scan String Instruction	213	
Alternative Coding for String Instructions	214	
Replicating a Pattern	215	
Key Points	215	
Review Questions and Exercises	216	

<b>12</b>	<b>ARITHMETIC I: PROCESSING BINARY DATA</b>	<b>218</b>
	Introduction 218	
	Processing Unsigned and Signed Binary Data 218	
	Addition and Subtraction of Binary Data 220	
	Multiplying Binary Data 225	
	Dividing Binary Data 232	
	The Numeric Data Processor 237	
	Key Points 239	
	Review Questions and Exercises 239	
<b>13</b>	<b>ARITHMETIC II: PROCESSING ASCII AND BCD DATA</b>	<b>241</b>
	Introduction 241	
	Data in Decimal Format 242	
	Processing ASCII Data 242	
	Processing Packed BCD Data 247	
	Converting ASCII Data to Binary Format 250	
	Converting Binary Data to ASCII Format 251	
	Shifting and Rounding a Product 252	
	Key Points 257	
	Review Questions and Exercises 258	
<b>14</b>	<b>DEFINING AND PROCESSING TABLES</b>	<b>259</b>
	Introduction 259	
	Defining Tables 259	
	Direct Addressing of Table Entries 261	
	Searching a Table 264	
	The XLAT (Translate) Instruction 268	
	Sorting Table Entries 271	
	Address Tables 274	
	Two-Dimensional Arrays 275	
	Key Points 278	
	Review Questions and Exercises 278	
<b>PART E</b>	<b>Advanced Input/Output</b>	<b>281</b>
<b>15</b>	<b>FACILITIES FOR USING THE MOUSE</b>	<b>281</b>
	Introduction 281	
	Basic Mouse Operations 283	
	Program: Displaying the Mouse Location 284	
	More Advanced Mouse Operations 287	
	Program: Using the Mouse with a Menu 290	
	Key Points 293	
	Review Questions and Exercises 294	

<b>16</b>	<b>DISK STORAGE I: ORGANIZATION</b>	<b>295</b>
	Introduction	295
	Characteristics of a Disk Storage Device	295
	The Disk System Area and Data Area	299
	The Boot Record	300
	The Directory	301
	The File Allocation Table	302
	Processing Files on Disk	307
	Key Points	308
	Review Questions and Exercises	308
<b>17</b>	<b>DISK STORAGE II: WRITING AND READING FILES</b>	<b>310</b>
	Introduction	310
	ASCII Strings	310
	File Handles	311
	Error Return Codes	311
	File Pointers	311
	Creating Disk Files	312
	Reading Disk Files	314
	Random Processing	318
	Key Points	327
	Review Questions and Exercises	328
<b>18</b>	<b>DISK STORAGE III: INT 21H FUNCTIONS FOR SUPPORTING DISKS AND FILES</b>	<b>330</b>
	Introduction	330
	Operations Handling Disk Drives	331
	Operations Handling the Directory and the FAT	341
	Operations Handling Disk Files	343
	Key Points	349
	Review Questions and Exercises	352
<b>19</b>	<b>DISK STORAGE IV: INT 13H DISK FUNCTIONS</b>	<b>353</b>
	Introduction	353
	BIOS Status Byte	354
	Basic INT 13H Disk Operations	354
	Other INT 13H Disk Operations	358
	Key Points	362
	Review Questions and Exercises	363
<b>20</b>	<b>FACILITIES FOR PRINTING</b>	<b>364</b>
	Introduction	364
	Common Printer Control Characters	364
	INT 21H Function 40H: Print Characters	365

Special Printer Control Characters 369  
 BIOS INT 17H Functions for Printing 373  
 Key Points 374  
 Review Questions and Exercises 375

**Part F Special Topics 377**

**21 DEFINING AND USING MACROS 377**

Introduction 377  
 Simple Macro Definitions 378  
 Using Parameters in Macros 379  
 Using Comments in Macros 380  
 Nested Macros 382  
 Macro Directives 383  
 Key Points 391  
 Review Questions and Exercises 392

**22 LINKING TO SUBPROGRAMS 393**

Introduction 393  
 The SEGMENT Directive 394  
 Intra-segment Calls 395  
 Inter-segment Calls 396  
 The EXTRN and PUBLIC Attributes 397  
 Using EXTRN and PUBLIC for an Entry Point 398  
 Defining the Code Segment as PUBLIC 400  
 Using Simplified Segment Directives 402  
 Passing Parameters to a Subprogram 404  
 The ENTER and LEAVE Instructions 406  
 Linking a C/C++ Program with an Assembly Language Program 408  
 Key Points 412  
 Review Questions and Exercises 412

**23 PROGRAM LOADING AND OVERLAYS 414**

Introduction 414  
 The Program Segment Prefix 414  
 The High-Memory Area 418  
 Memory Allocation Strategy 418  
 The Program Loader 419  
 Allocating and Freeing Memory 424  
 Loading or Executing a Program Function 425  
 Program Overlays 428  
 Resident Programs 432  
 Key Points 435  
 Review Questions and Exercises 436



<b>Part G Reference Chapters</b>	<b>437</b>
<b>24 BIOS DATA AREAS, INTERRUPTS, AND PORTS</b>	<b>437</b>
Introduction	437
The Boot Process	437
The BIOS Data Area	438
Interrupt Services	442
BIOS Interrupts	443
INT 21H Services	446
Ports	451
String Input/Output	453
Generating Sound	454
Key Points	455
Review Questions and Exercises	456
<b>25 OPERATORS AND DIRECTIVES</b>	<b>458</b>
Introduction	458
Type Specifiers	458
Operators	459
Directives	463
<b>26 THE PC INSTRUCTION SET</b>	<b>479</b>
Introduction	479
Register Notation	480
The Addressing Mode Byte	480
The Instruction Set	483
 <b>APPENDICES</b>	
<b>A CONVERSION BETWEEN HEXADECIMAL AND DECIMAL NUMBERS</b>	<b>508</b>
<b>B ASCII CHARACTER CODES</b>	<b>511</b>
<b>C THE DEBUG PROGRAM</b>	<b>513</b>
<b>D RESERVED WORDS</b>	<b>520</b>
<b>E ASSEMBLING AND LINKING PROGRAMS</b>	<b>522</b>
<b>F KEYBOARD SCAN CODES AND ASCII CODES</b>	<b>528</b>
 <b>INDEX</b>	<b>533</b>