IBM® PC ASSEMBLY LANGUAGE AND PROGRAMMING

Fourth Edition

IBM® PC 汇编语言与程序设计

第 4 版

Peter Abel

Professor Emeritus

British Columbia

Institute of Technology

清华大学出版社

Prentice-Hall International, Inc.

(京)新登字 158 号

IBM PC assembly language and programming 4th ed./Peter Abel

北京市版权局著作权合同登记号:01-98-0391

图书在版编目(CIP)数据

IBM PC 汇编语言与程序设计:第 4 版:英文/埃布尔(Abel, P.)著. - 影印版. - 北京:清华大学出版社,1998.2
(大学计算机教育丛书)
ISBN 7-302-02830-3

Ⅰ.I… Ⅱ.埃… Ⅲ.程序语言-程序设计-英文 Ⅳ.TP311.11

中国版本图书馆 CIP 数据核字(98)第 07060 号

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)
http://www.tup.tsinghua.edu.cn
印刷者:清华大学印刷厂
发行者:新华书店总店北京发行所
开 本:850×1168 1/32 印张:19.625
版 次:1998 年 5 月第 1 版 2001 年 9 月第 4 次印刷
书 号:ISBN 7-302-02830-3/TP・1487
印 数:13001～15000
定 价:30.00 元

出版者的话

今天，我们的大学生、研究生和教学、科研工作者，面临的是一个国际化的信息时代。他们将需要随时查阅大量的外文资料；会有更多的机会参加国际性学术交流活动；接待外国学者；走上国际会议的讲坛。作为科技工作者，他们不仅应有与国外同行进行口头和书面交流的能力，更为重要的是，他们必须具备极强的查阅外文资料获取信息的能力。有鉴于此，在国家教委所颁布的“大学英语教学大纲”中有一条规定：专业阅读应作为必修课程开设。同时，在大纲中还规定了这门课程的学时和教学要求。有些高校除开设“专业阅读”课之外，还在某些专业课拟进行英语授课。但教、学双方都苦于没有一定数量的合适的英文原版教材作为教学参考书。为满足这方面的需要，我们陆续精选了一批国外计算机科学方面最新版本的著名教材，进行影印出版。我社获得国外著名出版公司和原著作者的授权将国际先进水平的教材引入我国高等学校，为师生们提供了教学用书，相信会对高校教材改革产生积极的影响。

我们欢迎高校师生将使用影印版教材的效果、意见反馈给我们，更欢迎国内专家、教授积极向我社推荐国外优秀计算机教育教材，以利我们将《大学计算机教育丛书(影印版)》做得更好，更适合高校师生的需要。

清华大学出版社

《大学计算机教育丛书(影印版)》项目组

1999.6

Contents

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." The manufacturers set this tiny chip into a device resembling a centipede and connected it into a functioning system. 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 provided some compatibility with the 8080 and represented a significant advance on its design. Next, Intel developed the 8088, a variation of the 8086 that 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. An enhanced version of the 8088 is the 80188, and enhanced versions of the 8086 are the 80186, 80286, 80386, 80486, Pentium (or 586), PentiumPro (or 6x86) each of which provides additional operations and 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 is known as the system's machine language, which (as you will soon discover) is too complex and obscure for developing programs. Software

suppliers provide an assembly language for the processor family that represents the various instructions in a more understandable symbolic code.

High-level languages such as C and BASIC were designed to eliminate the technicalities of a particular computer, whereas a low-level assembly language is designed for a specific family of processors.

The use of assembly language provides a number of advantages:

- A program written in assembly language requires considerably less memory and execution time than a program 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.
- A knowledge of assembly language provides an understanding of machine architecture that no high-level language can ever provide.
- 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 routines that have caused processing bottlenecks.
- 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 DOS operating system (preferably a recent version) and familiarity with its use.
- A copy of an assembler translator program (preferably a recent version). 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.

OPERATING SYSTEMS

The major purposes of an operating system are (1) to allow users to instruct a computer regarding actions it is to take (such as executing a particular program) and (2) to provide means of storing (cataloging) information on disk and of accessing it.

The basic operating system for the PC and its compatibles is MS-DOS from Microsoft, known as PC-DOS on the IBM PC. Each version of DOS has provided additional features that have extended the capability of the PC. It is much easier to learn the intricacies of assembly language while within a relatively simple operating system like DOS

rather than attempt it from within the OS/2 or Windows environment. Within DOS, you can freely experiment and can later step up to more advanced systems.

FOCUS OF THIS BOOK

The primary aim of this book is to assist readers in learning assembly programming. To this end, the 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 22); 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 clever techniques of the professional.

THE APPROACH TO TAKE

This book can act as both a tutorial and a permanent 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. Key the program examples into your computer, convert them into executable "modules," and get them to execute (or "run"). Also, be sure to work through the exercises at the end of each chapter.

The first nine chapters furnish the foundation material for the book and for assembly language. After studying these chapters, you can proceed with Chapters 12, 13, 15, 16, 20, 21, or 22. Chapters 25, 26, and 27 are intended as references. Chapters related to each other are:

- 9 through 11 (on screen and keyboard operations)
- 13 and 14 (on arithmetic operations)
- 16 through 19 (on disk processing)
- 23 and 24 (on subprograms and memory management)

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 programs.
- 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 FOURTH EDITION

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

- More features of the Intel 80486 and later processors
- More program examples and exercises
- Earlier introduction to interrupt operations
- Inclusion of material on more recent assembler versions
- Considerable reorganization and revision of explanations throughout the text
- Revised and additional questions at the end of each chapter.

Users of the third edition should note that the contents of Chapters 25 and 26 have been combined in this edition. Also, the conditional jump instructions described in Chapter 27 ("The PC Instruction Set") are now organized and summarized in a table.

ACKNOWLEDGMENTS

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