

## <<汇编语言程序设计>>

### 图书基本信息

书名：<<汇编语言程序设计>>

13位ISBN编号：9787302260301

10位ISBN编号：7302260303

出版时间：2011-10

出版时间：清华大学出版社

作者：埃尔温

页数：715

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<汇编语言程序设计>>

### 内容概要

本书全面细致地讲述了汇编语言程序设计的各个方面。从微处理器体系结构、工作机制到指令集；从最基本的编译器链接器的使用到高级过程、结构和宏的使用；从用纯汇编编写程序到用vc++、bc++等最新编译器与汇编的混合接口编程；从16位实模式下bios、dos实模式文本及图形程序设计到32位保护模式的windows程序设计；从磁盘基础知识到intel指令编码、浮点运算等相关知识都做了深入而细致的讲解。

本书内容广博，但又讲解得非常深入，是汇编语言课程的很好教材。

## <<汇编语言程序设计>>

### 作者简介

Kip Irvine has written five computer programming textbooks, for Intel Assembly Language, C++, Visual Basic (beginning and advanced), and COBOL. His book Assembly Language for Intel-Based Computers has been translated into six languages. His first college degrees (B.M., M.M., and doctorate) were in Music Composition, at University of Hawaii and University of Miami. He began programming computers for music synthesis around 1982 and taught programmi.

## &lt;&lt;汇编语言程序设计&gt;&gt;

## 书籍目录

- preface xix
- 1 basic concepts
  - 1.1 welcome to assembly language
  - 1.2 virtual machine concept
  - 1.3 data representation
  - 1.4 boolean operations
  - 1.5 chapter summary
  - 1.6 exercises
- 2 x86 processor architecture
  - 2.1 general concepts
  - 2.2 x86 architecture details
  - 2.3 x86 memory management
  - 2.4 components of atypical x86 computer
  - 2.5 input-output system
  - 2.6 chapter summary
  - 2.7 chapter exercises
- 3 assembly language fundamentals
  - 3.1 basic elements of assembly language
  - 3.2 example: adding and subtracting integers
  - 3.3 assembling, linking, and running programs
  - 3.4 defining data
  - 3.5 symbolic constants
  - 3.6 real-address mode programming (optional)
  - 3.7 chapter summary
  - 3.8 programming exercises
- 4 data transfers, addressing, and arithmetic
  - 4.1 data transfer instructions
  - 4.2 addition and subtraction
  - 4.3 data-related operators and directives
  - 4.4 indirect addressing
  - 4.5 jmp and loop instructions
  - 4.6 chapter summary
  - 4.7 programming exercises
- 5 procedures
  - 5.1 introduction
  - 5.2 linking to an external library
  - 5.3 the book's link library
  - 5.4 stack operations
  - 5.5 defining and using procedures
  - 5.6 program design using procedures
  - 5.7 chapter summary
  - 5.8 programming exercises
- 6 conditional processing
  - 6.1 introduction
  - 6.2 boolean and comparison instructions

## &lt;&lt;汇编语言程序设计&gt;&gt;

- 6.3 conditional jumps
- 6.4 conditional loop instructions
- 6.5 conditional structures
- 6.6 application: finite-state machines
- 6.7 conditional control flow directives
- 6.8 chapter summary
- 6.9 programming exercises
- 7 integer arithmetic
  - 7.1 introduction
  - 7.2 shift and rotate instructions
  - 7.3 shift and rotate applications
  - 7.4 multiplication and division instructions
  - 7.5 extended addition and subtraction
  - 7.6 ascii and unpacked decimal arithmetic
  - 7.7 packed decimal arithmetic
  - 7.8 chapter summary
  - 7.9 programming exercises
- 8 advanced procedures
  - 8.1 introduction
  - 8.2 stack frames
  - 8.3 recursion
  - 8.4 invoke, addr, proc, and proto
  - 8.5 creating multimodule programs
  - 8.6 java bytecodes
  - 8.7 chapter summary
  - 8.8 programming exercises
- 9 strings and arrays
  - 9.1 introduction
  - 9.2 string primitive instructions
  - 9.3 selected string procedures
  - 9.4 two-dimensional arrays
  - 9.6 java bytecodes: string processing
  - 9.7 chapter summary
  - 9.8 programming exercises
- 10 structures and macros
  - 10.1 structures
  - 10.2 macros
  - 10.3 conditional-assembly directives
  - 10.4 defining repeat blocks
  - 10.5 chapter summary
  - 10.6 programming exercises
- 11 ms-windows programming
  - 11.1 win32 console programming
  - 11.2 writing a graphical windows application
  - 11.3 dynamic memory allocation
  - 11.4 x86 memory management
  - 11.5 chapter summary

## <<汇编语言程序设计>>

- 11.6 programming exercises
- 12 floating-point processing and instruction encoding
  - 12.1 floating-point binary representation
  - 12.2 floating-point unit
  - 12.3 x86 instruction encoding
  - 12.4 chapter summary
  - 12.5 programming exercises
- 13 high-level language interface
  - 13.1 introduction
  - 13.2 inline assembly code
  - 13.3 linking to c/c++ in protected mode
  - 13.4 linking to c/c++ in real-address mode
  - 13.5 chapter summary
  - 13.6 programming exercises
- 14 16-bit ms-dos programming
  - 14.1 ms-dos and the ibm-pc
  - 14.2 ms-dos function calls (int 21h)
  - 14.3 standard ms-dos file i/o services
  - 14.4 chapter summary
  - 14.5 programming exercises

chapters are available from the companion web site
- 15 disk fundamentals
- 16 bios-level programming
- 17 expert ms-dos programming
  - appendix a masm reference
  - appendix b the x86 instruction set
  - appendix c answers to review questions
  - appendices are available from the companion web site
  - appendix d bios and ms-dos interrupts
  - appendix e answers to review questions (chapters 15-17)
- index

章节摘录

版权页：插图：An effective way to explain how a computer's hardware and software are related is called the virtual machine concept. A well-known explanation of this model can be found in Andrew Tanenbaum's book, Structured Computer Organization. To explain this concept, let us begin with the most basic function of a computer, executing programs. A computer can usually execute programs written in its native machine language. Each instruction in this language is simple enough to be executed using a relatively small number of Programmers would have a difficult time writing programs in LO because it is enormously detailed and consists purely of numbers. If a new language, LI, could be constructed that was easier to use, programs could be written in LI. There are two ways to achieve this: Interpretation: As the LI program is running, each of its instructions could be decoded and executed by a program written in language LO.

## <<汇编语言程序设计>>

### 编辑推荐

《汇编语言程序设计(第6版)(影印版)》是大学计算机教育国外著名教材系列之一。



<<汇编语言程序设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>