<<Intel汇编语言程序设计>>

图书基本信息

书名：<<Intel汇编语言程序设计>>

13位ISBN编号：9787302197935

10位ISBN编号：7302197938

出版时间：2009-5

出版时间：清华大学出版社

作者：（美）埃尔温　著

页数：705

前言

PREfaCEAssembly Language for intel-Based Computers, Fifth Edition, teaches assembly language programs suage for intel-Based Computers, Fifth Edition, teaches assembly language program.,,.ining and architecture for intel lA-32 processors. It is an aDproDriate text for the following tvDes ofs and architecture for intel lA-32 processors. It is an appropriate text for the following types ofcollege courses:. AssemblV Language Programmingy Language Programming. Fundamentals of Computer Systems. Fundamentals of Computer ArchitectureStudents use intel or AMD processors and program with Microsoft Macro Assembler (MASM)8.0, running on any of the following MS-Windows platforms: Windows 95, 98, Millenium, NT, 2000,and XP.Although this book was originally designed as a programming textbook for college students, it has.if this book was originally designed as a programming textbook for college students, it hasevolved over the last 15 years into much more. Many uniVersities use the book for their introductoryJ J ycomputer architecture courses. As a testament to its popularity, the foufth edition was tfanslated into..KOrean, Chinese, French, Russian, and Polish.Emphasl'S of TOPiCS This edition includes topics that lead naturally into subsequent courses incomputer architecture, operating systems, and compiler writing:4,. Virtual machine concept. Elementary boolean operations; Doolean operations. Instruction execution cVcley de. Memory access and handshakingJ 5. Interrupts and polling. Pipelining and superscalar concepts. Hardware-based I/O. Floating-point binary representationOther topics relate specifically to intel lA-32 architecture:'. lA-32 protected memory and paging.. MemorV segmentation in real-address modeJ segmentation in real-address mode. 16-bit intemipt handling. MS-DOS and BIOS sVstem calls (interrupts)y item calls (interrupts). lA-32 Floating-Point Unit architecture and programming. lA-32 Instruction encodingCeftain examDles Dresented in the book lend themselves to courses that occur later in a comDUterpies presented in the book lend themselves to courses that occur later in a computer..,science curriculum:. Searching and sorting algorithms. High-level language structures.h-level language structures. Finite-state machines. Code optimization examples'.XIXU PREfACEimprOVementS in the Rfth Edition A number of improvements and new information have beenadded in this edition, listed in the following table by chapter number fStill a Programming Book This book is still focused on its original mission: to teach students howto write and debug programs at the machine level. It will never replace a complete book on computerarchitecture, but it does give students the first-hand experience of writing software in an environmentthat teaches them how a computCr works. Our premise is that students retain knowledge better whentheory is combined with experience. In an engineering course, students constrict prototypes; in acomputer architecture course, students should write machine-level programs. In both cases, they have amemorable experience that giVes them the confidence to work in any OS/machine-oriented environment.Real MOds and Protsctsd MOds This edition emphasizes 32-bit protected mode, but it still hasthree chapters devoted to real-mode programming. For example, there is an entire chapter on BIOSprogramming for the keyboard, video display (including graphics), and mouse. Another chapter covers MS-DOS programming using interrupts (system calls). Students can benefit from programmingdirectly to hardware and the BIOS.The examples in the first half of the book are nearly all presented as 32-bit text-oriented aPPlicationsruwhng in protected mode using the flat memory model. This approach is wonderfully simple because itavoids the complications of segment-offset addressing. Specially marked paragraphs and popup boxespoint out occasional differences between protCcted mode and real mode programming. Most differencesare abstracted by the book's parallel link libraries for real mode and protected mode programming.Link LibrarieS We supply two versions of the link library that students use for basic input-output,simulations, timing, and other useful stuff. The 32-bit version (lrvine32.lib) runs in protected mode,sending its output to the Win32 console. The 16-bit version (lrvine16.lib) runs in real-address mode.Full source code for the libraries is supplied on the book's Web site. The link libraries are availableonly for convenience, not to prevent stlldeflts from learning how to program input-output themselves.Stlldents are encouraged to create their own libraries.PREfACE fbiIncludsd Software and EX8mPI8s All the example programs were tested with Microsoft MacroAssembler Version 8.0. The 32-bit C++ applications in Chapter 12 were tested with Microsoft VisualC++ .NET. The real-address mode programs in Chapter 12 (linking to C++) were assembled withprograms in Chapter 12 (linking to C++) were assembled withBorland Turbo Assembler (TASM).W8b Sits informatl'On Updates and corrections to this book may be found at the book's Web site,http: //www. asmi rvi ne. corn, including additional programming projects for instructors to assignat the ends of chapters. If for some reason you cannot access this site, information about the book andpiers. If for some reason you cannot access this site, information about the book anda link to its current Web site can be found at www. prenhall. corn by searching for the book title or forthe author name "Kip lrvine."Overall GoalsThe following goals of this book are designed to broaden the student's interest and knowledge in top. 1ics related to assembly language:; language:. Intel lA-32 processor architecture and programming. Real-address mode and protected mode programming. AssemblV language directiVes, macros, operators, and program structurey language directiVes, macros, operators, and program structure. Programming methodology, showing how to use assembly language to create system-level soft.ramming methodology, showing how to use assembly language to create system-level software tools and application programs. Computer hardware manipulation. Interaction between assembly language programs, the operating system, and other aPPlication programsJ a o programs, the operating systCm, and other aPPlication programsOne of our goals is to help students approach programming problems with a machine-level mindset. It is impoftant to think of the CPU as an interactiVe tool, and to learn to monitor its operation asdirectly as Dossible. A debugger is a programmer's best friend, not only for catching errors, but as anJ as possible. A debugger is a programmer's best friend, not only for catching errors, but as aneducational tool that teaches about the CPU and operating system. We encourage students to lookbeneath the surface of high-level languages and to realize that most programming languages aredesigned to be portable and, therefore, independent of their host machines.In addition to the short examples, this book contains hundreds of ready-to-run programs that demonstrate instructions or ideas as they are Dresented in the text. Reference materials, such as guides toJ presented in the text. Reference materials, such as guides toMS-DOS interrupts and instruction mnemonics, are available at the end of the book.RequiTed BaCkground The reader should already be able to program confidently in at least oneother programming language, preferably Java, C, or C++. One chapter covers C++ interfacing, so it isvery helpful to have a compiler on hand. I have used this book in the classroom with majors in both.'computer science and management information systems, and it has been used elsewhere in engineer.lug courses.FeaturesCOmPI8ts Program LiStingS A companion CD-ROM contains all the source code from the examples in this book. Additional listings are available on the book's Web page. An extensive link library issupplied with the book, containing more than 30 procedures that simplify user input-output, numeric.,.', n 1, 11.'.processing, disk and file handling, and string handling. In the beginning stages of the course, studentscan use this library to enhance their programs. Latef, they can create their own procedures and addthem to the librarV'y.Programming LOgiC TWo chapters emphasize boolean logic and bit-level manipulation. A con. scions attempt is made to relate high-level programming logic to the low-level details of the machine.This approach helps students to create more efficient implementations and to better understand howcompilers generate object code...all PREfACEHardWare and OOeratl'na SyStsm ConceotS The first two chanters introduce basic hardwareperatl'ng Spetsm ConceptS The first two chapters introduce basic hardwareand data representation concepts, including binary numbers, CPU architecture, status flags, and memory.mapping. A survey of the computer's hardware and a historical perspectiVe of the intel processor familyhelps students to better understand their target computer system.StFUctUred Proaramml'na Approach BeZinning with Chapter 5, procedures and functional decomsramming APProach Beginning with Chapter 5, procedures and functional decomposition are emphasized. Students are giVen more complex programming exercises, requiring them tofocus on design before starting to write code.DI'Sk Storaae ConceDtS Students learn the fundamental principles behind the disk storage systemHe ConceptS Students learn the fundamental principles behind the disk storage systemon MS-Windows--based sVstems from hardware and software Doints of view.y items from hardware and software points of view.Creatl'na LI'nk LI'braries Students are free to add their own Drocedures to the book's link libraryd Link LI'brarieS Students are free to add their own procedures to the book's link libraryand create new libraries. TheV learn to use a toolbox approach to programming and to write code thatJ learn to use a toolbox approach to programming and to write code that.is useful in more than one program.MaCros and StFUctUres A chapter is devoted to creating structures, unions, and macros, which areesential in assemblV language and systems programming. Conditional macros with advanced operas language and systems programming. Conditional macros with advanced operators serve to make the macros more professional.professional.IRtsrfscina tO High-Level Lanauaaes A chanter is devoted to interfacinZ assembly lanZuaZe to Cs ic High-Level Languages A chapter is devoted to interfacing assembly language to Cand C++. This is an important job skill for students who are lacely to find jobs programming in high-levelJlanguages. They can ie~ to optimize their code and see examples of how C++ compilers optimize code.InstFUctional Al'dS All the program listings are available on disk and on the Web. Instructors are provided a test baal, answers to review questions, solutions to programming exercises, and a Microsoft.Powerpoint slide presentation for each chapter.. Summary of ChaDtersy of ChaptersChapters I to & contain a basic foundation of assembly language and should be covered in sequence.. After that, you have a fair amount of freedom. The following chapter dependency graph shows howlater chapters depend on knowledge gained from other chapters. Chapter 10 was split into two partsJfor this graph because no other chapter depends on one's knowing how to create macros:.laph because no other chapter depends on one's knowing how to create macros:@ UMughs 491. Basic Concepts: Applications of assembly language, basic concepts, machine language, and datarepresentation.2. lA-32 Processor Architecture: Basic microcomputer design, instruction execution cycle, lA-32.t.processor architecture, lA-32 memory management, components of a microcomputef, and the.input-output system.3. Assembly LanZuage Fundamentals: Introduction to assembly language, liaring and debugging,J Language Fundamentals: Introduction to assembly language, liaring and debugging,and defininZ constants and variables.5PREfACE XXiii4. Data Transfers, Addressing, and Arithmetic: Simple data transfer and arithmetic instructions,assemble-link-execute cVcle, operators, directiVes, expressions, JMP and LOOP instructions, andJ, operators, directiVes, expressions, JMP and LOOP instructions, andindirect addressing.5.5. Procedures: Linking to an external library, description of the book's link library, stack opera..nons, defining and using procedures, flowchins, and top-down structured design.6. Conditional Processing: Boolean and comparison instructions, conditional jumps and loops,high-level logic structures, and finite state machines.7. Integer Arithmetic: Shift and rotate instructions with useful applications, multiplication anddiVision, extended addition and subtraction, and ASCll and packed decimal arithmetic.8. Advanced Procedures: Stack parameters, local variables, advanced PROC and INVOKE direc..tives, and recursion.9. Strings and Arrays: String primitiVes, manipulating arrays of characters and integers, twodimensional arraVs, sorting, and searching.y o, hcf ting, and searching.10. Structllres and Macros: Stmctures, macros, conditional assembly directiVes, and defining repeat blocks.11. MS-Windows Programming: Protected mode memory management concepts, using the MicrosoftWindows API to display text and colors, and dynamic memory allocation.12. High-Level Language interface: Parameter passing conventions, inline assembly code, and linkingassemb1V language modules to C and C++ programs.y language modules to C and C++ programs.13. 16-Bit MS-DOS Programming: Calling MS-DOS interrupts for console and file input-output.14. Disk Fundamentals: Disk storage systems, sectors, clusters, directories, file allocation tables, handlingMS-DOS error codes, and driVe and directory m~ulation.15. BIOS-Level Programming: Keyboard input, video text, graphics, and mouse programming.16. ExDert MS-DOS Programming: Custom-designed segments, runtime program structure, andpert MS-DOS Programming: Custom-designed segments, runtime program structure, andInterrupt handling. Hardware control using I/O ports.17. Floating-Point Processing and instruction Encoding: Floating-point binary representation andfloatinZ-DOint arithmetic. Learning to program the lA-32 Floating-Point Unit. Understanding theo point arithmetic. Learning to program the lA-32 Floating-Point Unit. Understanding theencoding of lA-32 machine instructions.5Appendix A: MASM ReferenceAppendix B: The lA-32 Instruction SetAppendix C: BIOS and MS-DOS InterruptsAppendix D: Answers to Review QuestionsReference MaterialsW8b Site The author maintains an actiVe Web site at www.asmirvine.com.H8lP the Help file (in Windows Help Format) by Gerald Cahill of Antelope Valley College. Documents the book's link libraries, as well as Win32 data structures.Assembly Language Workbook An interactive workbook is included on the book's Web site, cover'. 1. Ilug such important topics as number conversions, addressing modes, register usage, Debug program.'minZ and floating-point binary numbers. The content pages are HTML documents, making it easyb, and floating-point binary numbers. The content pages are HTML documents, making it easyfor students and instructors to add their own customized content. This workbook is also available onthe book's Web site.Debugging Tools Tutorials on using Microsoft CodeView, Microsoft Visual Studio, and MicrosoftWindows Debugger (WinDbg).BIOS and MS-DOS Im6rruntS Appendix C contains a brief listing of the most-often-used INT 10hptS Appendix C contains a brief listing of the most-often-used INT 10h(video), INT 16h (keyboard), and INT Zlh (MS-DOS) functions.Instfucaon Set Appendix B lists most nonprivileged instructions for the lA-32 processor family.. D rXXIV PREfACEFor each instruction, we describe its effect, show its syntax, and show which flags are affected.Powerpoint Presentoaons A complete set of Microsoft Powerpoint presentations written by the author.AcknowledgmentsSpecial thanks are due to Tracy Dunkleberger, Executive Editor for Computer Science at PrenticeHall, who provided friendly, helpful guidance during the writing of the fifth edition. Karen Ettingerdid a terrific job as production editof, constantly keeping track of numerous minute details. Camille,oh as production editof, constantly keeping track of numerous minute details. CamilleTrentacoste was the book's managing editor.Fifth EditionI offer mV special thanks and gratitude to the following professors who boosted my morale, gave meJ peelal thanks and gratitude to the following professors who boosted my morale, gave megreat pedagogical tips, and tirelessly examined the entire book. They have been a huge influence onthe development of this book, in some cases across multiple editions:. Gerald Cahill, Antelope Valley College. James Brink, Pacific Lutheran University. William Barrett, San Jose State UniVersityMany thanks to Scott Blackledse and John Tavior, both professional programmers, who proofreadJ s Jlor, both professional programmers, who proofreadmost of the manuscript and flagged numerous errors. Several people reviewed individual chapters:. Jerry JoVce, Keene State CollegeJ J, Keene State College. Tianzheng Wu, Mount Mercy College. Ron Davis, Kennedy-King College. David Topham, Ohlone College. Harvev Nice. Depaul UniVersitVJ, Depaul UniVersityFOurth EditionThe following people were tremendously helpful in creating the fourth edition.o people were tremendously helpful in creating the fourth edition.. Gerald Cahill, Antelope Valley College. James Brink, Pacific Lutheran University. Maria Kolatis, County College of Moms. Tom JoVce, Chief Engineer at Premier Heart, LLC.. Jeff WOthke, Purdue Calumet University. Tim DowneV, Florida international Universityy, florida international UniversityThe followinZ individuals Drovided valuable Droofreading help in the fourth edition fo individuals provided valuable proofreading help in the fourth edition f. Andres Altamirano, Miami. Courtnev Amor, Los Angelesy Amor, Los Angeles. Scott Blackledge, Platform Solutions, Inc.. Ronald Davis, Kennedy-King College. Ata Elahi, Southern Connectictut State UniversitV. Jose Gonzalez, Miami. LeroV Highsmith, Southern Connecticut State UniVersity.. Sand lqbal, Faran institute of Technology,id lqbal, Faran institute of Technology. Charles Jones, Maryville College. Vincent KaVes, Mount St. Mary CollegeJ.. Eric Kobrin, Miami. Pablo Maurin, Miami. BarrV Meaker, Design Engineer, Boeing Corporationy bleaker, Design Engineer, Boeing Corporation. Ian Merkel, Miami. Sylvia Miner, MiamiPREfACE XXV. M. Nawaz, OPSTEC College of Computer Science. Kam Ng, Chinese University of Hong Kong. Hien Nguyen, Miami. Ernie Philipp, Northern Virginia Community College.. BoVd SteDhens, UGMO Research, LLCJ phens, UGMO Research, LLC. John Taylor, EnglandJ 5 england. ZacharV TaVlor, Columbia ColleZe, laylor, Columbia College. Virginia Welsh, Community College of Baltimore County.inia Welsh, Community College of Baltimore County. Robert WOrkman, Southern Connecticut State UniversityJ. Tianzheng Wu, Mount Mercy College& flu, Mount Mercy College. Matthew Zukoski, Lehigh University

<<Intel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。

作者简介

作者：吉普 (美国)埃尔温

<<Intel汇编语言程序设计>>

书籍目录

<<Intel汇编语言程序设计>>

编辑推荐

1. 《Intel汇编语言程序设计》组织结构灵活，教师可根据需要按不同的顺序、不同的深度进行授课。
2. 以32位保护模式编程为默认模型，使读者更容易理解和接受。
3. 为32位和16位程序设计给出了扩展的链接库，包括用于简化输入/输出、数字处理、磁盘与文件处理，以及字符串处理等的40多个函数。
4. 作者为每章提供了测试题库，方便教师准备考试。
5. 以自上而下的方式探究和解析程序的设计思想和过程。
6. 作者提供了内容丰富的教学课件。
7. 每小节后面都给出了复习题，有助于学生掌握所学知识。
8. 为教师提供了复习题和编程练习题答案。