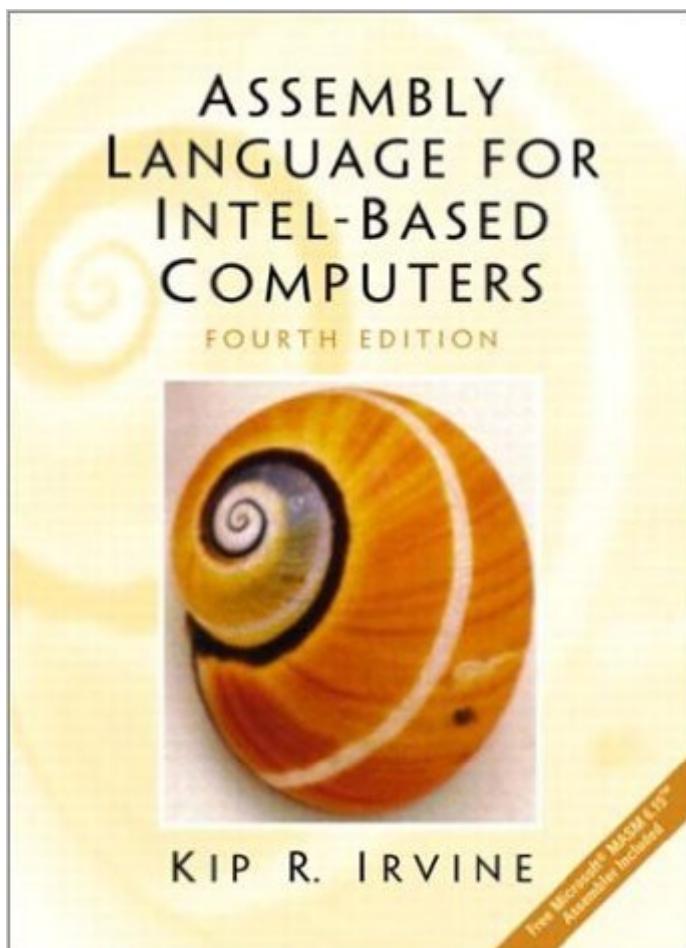


The book was found

Assembly Language For Intel-Based Computers (4th Edition)



Synopsis

For undergraduate courses in assembly language programming and introductory courses in computer systems and computer architecture. This complete and fully updated study of assembly language for the IBM-PC teaches students how to write and debug programs at the machine level. Based on the Intel processor family, the text simplifies and demystifies concepts that students need to grasp before they can go on to more advanced computer architecture or operating systems courses.

Book Information

Hardcover: 700 pages

Publisher: Prentice Hall; 4 edition (July 25, 2002)

Language: English

ISBN-10: 0130910139

ISBN-13: 978-0130910134

Product Dimensions: 6.8 x 1.4 x 9.2 inches

Shipping Weight: 8.8 ounces

Average Customer Review: 3.8 out of 5 starsÂ [See all reviewsÂ](#) (65 customer reviews)

Best Sellers Rank: #826,727 in Books (See Top 100 in Books) #52 inÂ Books > Computers & Technology > Programming > Languages & Tools > Assembly Language Programming #861 inÂ Books > Textbooks > Computer Science > Operating Systems #1981 inÂ Books > Computers & Technology > Operating Systems

Customer Reviews

This is a review for the Fourth Edition of Assembly Language for Intel-Based Computers by Kip Irvine. This book has just been released. I teach Assembly Language at a community college, and have used the third edition with much success. Matter of fact, I was so impressed by it that when Dr. Irvine asked if any of the current teachers who'd registered at his site wanted to review the fourth edition, which was still being written, I volunteered. The fourth edition is an organizational improvement on an already well-written book. The first nine chapters cover what I need to teach for an introduction to Assembly Language, required by Computer Science departments for a major in that field. Engineering programs can teach introductory hardware using it as well, with the emphasis on different chapters. The book starts with basic concepts of data representation and number-base conversions and moves on to the Intel IA-32 processor architecture. By the time we start to actually discuss Assembly Language in Chapter 3, the student understands why Assembly works the way it

does. That's no small feat. Unlike many Computer Science books purported to be for beginners, this one really is written for students near the beginning of their studies. It's aimed at those who have taken maybe one previous programming class. Dr. Irvine has a knack for reducing a very complex topic to its elements and explaining those elements so they're easily understood. It's possible to learn Assembly Language using this book alone, not taking a formal class, even though the text isn't really designed to be a Teach Yourself program. Assembly Language is a rather difficult topic, so the student would have to be very motivated. Instead of having a series of review questions at the ends of the chapters, this book has section reviews. This focuses the student on what was just read, reinforcing a smaller amount of data, so it sticks more easily. Students have access to answers to odd-numbered questions. For teachers who prefer to have students look in the textbook for the answers, the even-numbered questions cover material similar to what's asked by the odds. There are programming exercises at the ends of chapters, too. The author maintains a web site for the book. There are sections for students who use the book and parts just for teachers. Odd-numbered problem answers are on the students' site. Security is maintained at the teachers' site. When I logged in, I found sample tests, PowerPoint presentations, and FAQ's, to name but some of it. Rather than describe the contents of the book further, I'll give you the author's website. He has the complete table of contents posted, as well as a few sample chapters. Take a look for yourself. [\[URL\]](http://www.cs.iupui.edu/~cs320/AssemblyLang.html). I'm going to enjoy teaching from the Fourth Edition of Assembly Language for Intel-Based Computers. I feel like I have a well-organized book that covers what I have to teach, so I don't need to design my course around several textbooks. There's plenty of supplemental material on the website, which saves me lots of my most precious commodity - time.

This is a good starter book, but recognize that it focuses on DOS and does not "thoroughly cover assembly language for Intel-based computers" as the back jacket claims. No reasonably-sized book could do that. The book description states that it is "Designed for students and professionals interested in learning the basics of operating systems and architecture in the context of a microprocessor." Although the book covers some of the critical topics necessary for operating systems programming on the Intel Architecture, e.g. I/O, segmentation, interrupts, it fails to do so adequately, and it omits discussion of protection, paging, and privileged mode instructions, to name three among many. I grant that you could learn enough to understand the basics of real-mode DOS, but not any of the Windows flavors, OS/2, nor the varieties of Minix/Linux. That being said, it **is** a good book to learn Intel assembly from, and if you want to do OS programming then get the manuals from Intel's web site (downloadable in .pdf form). You may still find the DOS focus

distracting, but you'll have everything you need.... and don't think you'll find it all in one book. Intel's manuals alone are over 1500 pages.

This reviews the FORTH edition published July 25, 2002. This text addresses the needs of students who can write programs in a high-level language and wish to learn assembly language for the Intel family of microprocessors. The forth edition contains an abundance of new material. The author introduces 32-bit protect-mode programming using the Microsoft Windows platform and he delays coverage of 16-bit real-mode programming using the MS-DOS platform until chapter 13. Only MS-DOS programming was covered in the previous edition. Each section contains review questions and each chapter ends with a summary and several programming exercises. The author clearly describes each concept and uses excellent diagrams and code examples throughout the text.

Several new topics have been added to this edition such as graphics programming in both Windows and DOS applications. The early chapters present an overview of the Intel IA-32 architecture and cover the principles and techniques of programming Windows console applications in assembly language. The author has encapsulated the details of the Windows application interface in his 32-bit library, which the students use throughout the first 10 chapters. This allows the students to focus on learning and using the instruction set to create solutions to the programming problems. The details of his library routines are revealed in Chapter 11. Chapter 12 covers assembly language routines that can be called from a high-level language. Both 32-bit and 16-bit C/C++ compilers are used in the examples. The remaining chapters (13 through 17) cover 16-bit real-mode MS-DOS and BIOS programming in much the same way as was done in the third edition, but many more example programs have been added. A CD-ROM accompanies this text and contains the complete Microsoft Macro Assembler (MASM 6.15), Microsoft's 32-bit and 16-bit Linkers, all example programs from the text, the source code for the author's 32-bit and 16-bit link libraries, and many more useful files. Chapter 17, Advanced Topics, is contained on the CD-ROM and is not printed in the textbook. The author's web site ... contains an abundance of support material which he keeps current. Most useful are a list of corrections to the textbook, updates to the files contained on the CD-ROM, and the answers to the odd-numbered Review Questions. There are instructions for installing the assembler, configuring and using various integrated development environments, and links to other useful information such as the MASM 6 manual set and the author's e-mail address. Five chapters from the textbook are posted on his site for your review. Professors can access a password-protected area that contains answers to all of the review questions, solutions to the programming exercises, and a set of PowerPoint slides for use in the classroom. Assembly language is much easier to learn

when protect-mode is covered before real-mode. Early use of flat addressing allows coverage of segmented addressing to be delayed until the students have learned the complete instruction set. Flat addressing also makes it easy to locate run-time errors. The addresses contained in the Windows run-time error dialog box corresponds with the addresses in the linker's map file and the assembler's list file. The topics flow better when the simpler Windows interface is covered before the more complex interrupt service routines of MS-DOS and BIOS. I have successfully used the previous edition of this textbook for several semesters. Including techniques for protect-mode programming in the forth edition provides the additional information most often requested by students. This review is based on material I received from the publisher as a reviewer.

[Download to continue reading...](#)

Assembly Language for Intel-Based Computers (4th Edition) Assembly Language for Intel-Based Computers (3rd Edition) Introduction to 64 Bit Intel Assembly Language Programming for Linux: Second Edition Gun Digest Book of Firearms Assembly/Disassembly, Part 2: Revolvers (Gun Digest Book of Firearms Assembly/Disassembly: Part 1 Automatic Pistols) (Pt. 2) Kaizen Assembly: Designing, Constructing, and Managing a Lean Assembly Line Great Big World of Computers - History and Evolution : 5th Grade Science Series: Fifth Grade Book History Of Computers for Kids (Children's Computer Hardware Books) Digitalk PARTS Workbench for Win32 - 32-Bit Parts Assembly and Reuse Tool Set - Script Language Guide - Win32 Series Version 3.0 ARM Assembly Language Programming & Architecture (ARM books) (Volume 1) Z80 Assembly Language Programming Raspberry Pi Assembly Language RASPBIAN Beginners: Hands On Guide Raspberry Pi Assembly Language RISC OS Beginners (Hands On Guide) Assembly Language and Systems Programming for the M68000 Family Assembly Language Step-by-step: Programming with DOS and Linux (with CD-ROM) Basic IBM Mainframe Assembly Language Programming Raspberry Pi Assembly Language Beginners: Hands On Guide 80386/80486 Assembly Language Programming Principles of Computer Organization and Assembly Language 6502 Assembly Language Programming X86 Assembly Language and C Fundamentals Optimizing Powerpc Code: Programming the Powerpc Chip in Assembly Language

[Dmca](#)