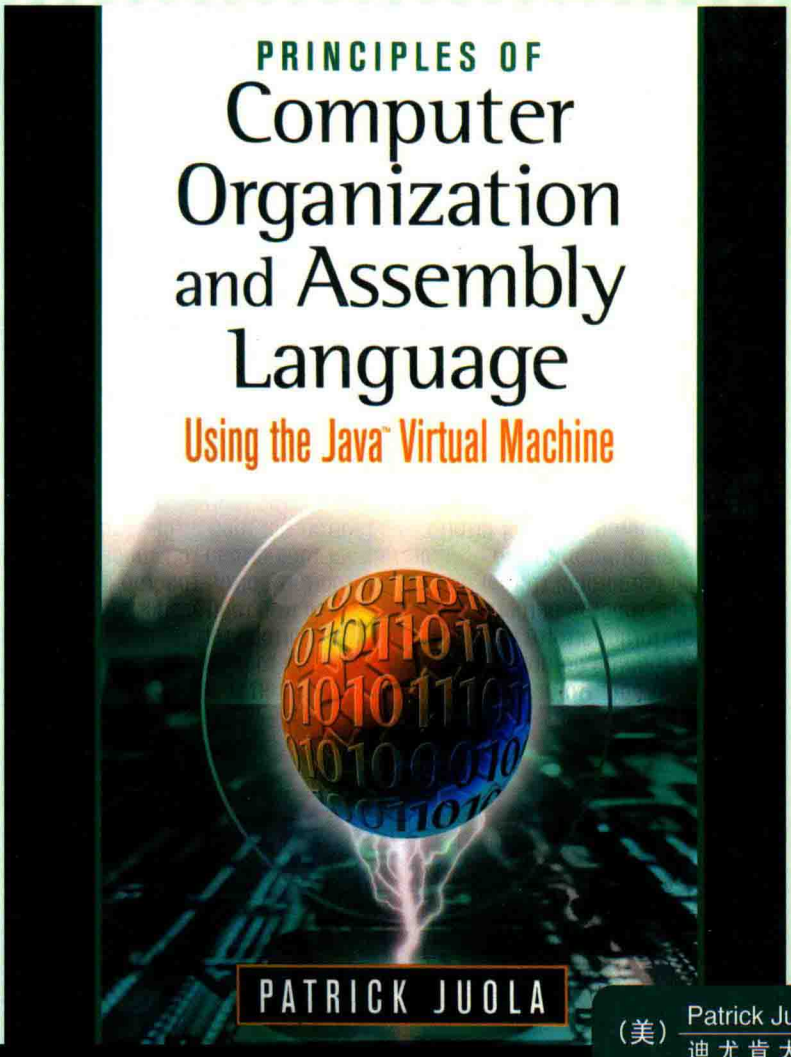
HZ BOOKS 华章教育
经 典 原 版 书 库
PRENTICE HALL

计算机组成及汇编语言原理

（英文版）



（美）Patrick Juola 迪尤肯大学 著

机械工业出版社
China Machine Press

经 典 原 版 书 库

计算机组成及汇编语言原理

（英文版）

Principles of Computer Organization and Assembly Language

Using the Java Virtual Machine

（美）Patrick Juola 迪尤肯大学 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2008-1783

图书在版编目（CIP）数据

计算机组成及汇编语言原理（英文版）/（美）卓拉（Juola, P.）著. —北京：机械工业出版社，2008.5

书名原文：Principles of Computer Organization and Assembly Language: Using the Java Virtual Machine

（经典原版书库）

ISBN 978-7-111-23917-8

Ⅰ.计… Ⅱ.卓… Ⅲ.①计算机体系结构-英文 ②汇编语言-英文 Ⅳ.TP303 TP313

中国版本图书馆CIP数据核字（2008）第049528号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：迟振春

北京京北制版厂印刷 · 新华书店北京发行所发行

2008年5月第1版第1次印刷

170mm × 242mm · 21.25印张

标准书号：ISBN 978-7-111-23917-8

定价：42.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近260个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”。为了保证这两套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、

北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这两套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

To My Nieces
Lyric Elizabeth, Jayce Rebekah, and Trinity Elizabeth

Preface

Statement of Aims

What

This is a book on the organization and architecture of the `Java Virtual Machine` (JVM), the software at the heart of the `Java language` and is found inside most computers, Web browsers, PDAs, and networked accessories. It also covers general principles of machine organization and architecture, with illustrations from other popular (and not-so-popular) computers.

It is *not* a book on Java, the programming language, although some knowledge of Java or a Java-like language (C, C++, Pascal, Algol, etc.) may be helpful. Instead, it is a book about how the Java language actually causes things to happen and computations to occur.

This book got its start as an experiment in modern technology. When I started teaching at my present university (1998), the organization and architecture course focused on the 8088 running MS-DOS—essentially a programming environment as old as the sophomores taking the class. (This temporal freezing is unfortunately fairly common; when I took the same class during my undergraduate days, the computer whose architecture I studied was only two years younger than I was.) The fundamental problem is that the modern Pentium 4 chip isn't a particularly good teaching architecture; it incorporates all the functionality of the twenty-year-old 8088, including its limitations, and then provides complex workarounds. Because of this complexity issue, it is difficult to explain the workings of the Pentium 4 without detailed reference to long outdated chip sets. Textbooks have instead focused on the simpler 8088 and then have described the computers students actually use later, as an extension and an afterthought. This is analogous to learning automotive mechanics on a Ford Model A and only later discussing such important concepts as catalytic converters, automatic transmissions, and key-based ignition systems. A course in architecture should not automatically be forced to be a course in the history of computing.

Instead, I wanted to teach a course using an easy-to-understand architecture that incorporated modern principles and could itself be useful for students. Since every computer that runs a Web browser incorporates a copy of the JVM as software, almost every machine today already has a compatible JVM available to it.

This book, then, covers the central aspects of computer organization and architecture: digital logic and systems, data representation, and machine organization/architecture. It also describes the assembly-level language of one particular architecture, the JVM, with other common architectures such as the Intel Pentium 4 and the PowerPC given as supporting examples but not as the object of focus. The book is designed specifically for a standard second-year course on the architecture and organization of computers, as recommended by the IEEE Computer Society and the Association for Computing Machinery.[1]

---

[1] "Computing Curricula 2001," December 15, 2001, Final Draft; see specifically their recommendation for course CS220.

How

The book consists of two parts. The first half (chapters 1–5) covers general principles of computer organization and architecture and the art/science of programming in assembly language, using the JVM as an illustrative example of those principles in action (How are numbers represented in a digital computer? What does the loader do? What is involved in format conversion?), as well as the necessary specifics of JVM assembly language programming, including a detailed discussion of opcodes (What exactly does the `i2c` opcode do, and how does it change the stack? What's the command to run the assembler?). The second half of the book (chapters 6–10) focuses on specific architectural details for a variety of different CPUs, including the Pentium, its archaic and historic cousin the 8088, the Power architecture, and the Atmel AVR as an example of a typical embedded systems controller chip.

For Whom

It is my hope and belief that this framework will permit this textbook to be used by a wide range of people and for a variety of courses. The book should successfully serve most of the software-centric community. For those primarily interested in assembly language as the basis for abstract study of computer science, the JVM provides a simple, easy-to-understand introduction to the fundamental operations of computing. As the basis for a compiler theory, programming languages, or operating systems class, the JVM is a convenient and portable platform and target architecture, more widely available than any single chip or operating system. And as the basis for further (platform-specific) study of individual machines, the JVM provides a useful explanatory teaching architecture that allows for a smooth, principled transition not only to today's Pentium, but also to other architectures that may replace, supplant, or support the Pentium in the future. For students, interested in learning how machines work, this textbook will provide information on a wide variety of platforms, enhancing their ability to use whatever machines and architectures they find in the work environment.

As noted above, the book is mainly intended for a single-semester course for second-year undergraduates. The first four chapters present core material central to the understanding of the principles of computer organization, architecture, and assembly language programming. They assume some knowledge of a high-level imperative language and familiarity with high school algebra (but not calculus). After that, professors (and students) have a certain amount of flexibility in choosing the topics, depending upon the environment and the issues. For Intel/Windows shops, the chapters on the 8088 and the Pentium are useful and relevant, while for schools with older Apples or a Motorola-based microprocessor lab, the chapter on the Power architecture is more relevant. The Atmel AVR chapter can lay the groundwork for laboratory work in an embedded systems or microcomputer laboratory, while the advanced JVM topics will be of interest to students planning on implementing JVM-based systems or on writing system software (compilers, interpreters, and so forth) based on the JVM architecture. A fast-paced class might even be able to cover all topics. The appendices are provided primarily for reference, since I believe that a good textbook should be useful even after the class is over.

Acknowledgments

Without the students at Duquesne University, and particularly my guinea pigs from the Computer Organization and Assembly Language classes, this textbook couldn't have happened. I am also grateful for the support provided by my department, college, and university, and particularly for the support funding from the Philip H. and Betty L. Wimmer Family Foundation. I would also like to thank my readers, especially Erik Lindsley of the University of Pittsburgh, for their helpful comments on early drafts.

Without a publisher, this book would never have seen daylight; I would therefore like to acknowledge my editors, Tracey Dunkelberger and Kate Hargett, and through them the Prentice Hall publishing group. I would like to express my appreciation to all of the reviewers: Mike Litman, Western Illinois University; Noe Lopez Benitez, Texas Tech University; Larry Morell, Arkansas Tech University; Peter Smith, California State University–Channel Islands; John Sigle, Louisiana State University–Shreveport; and Harry Tyrer, University of Missouri–Columbia. Similarly, without the software, this book wouldn't exist. Aside from the obvious debt of gratitude to the people at Sun who invented Java, I specifically would like to thank and acknowledge Jon Meyer, the author of `jasmin`, both for his software and for his helpful support.

Finally, I would like to thank my wife, Jodi, who drew preliminary sketches for most of the illustrations and, more importantly, has managed to put up with me throughout the book's long gestation and is still willing to live in the same house.

Contents