

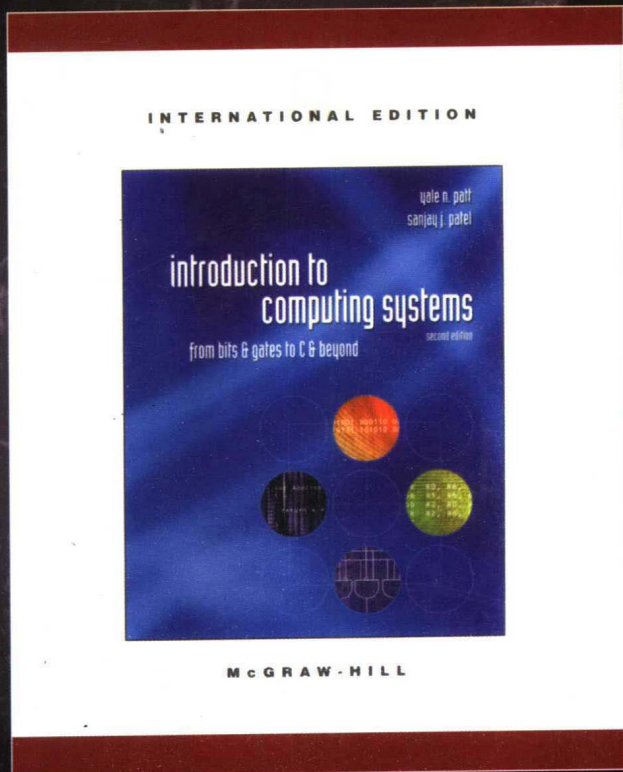


计算机科学丛书

原书第2版

计算机系统概论

(美) Yale N. Patt (得克萨斯大学奥斯汀分校) 著 梁阿磊 蒋兴昌 林凌 译
Sanjay J. Patel (伊利诺伊大学) 上海交通大学



Introduction to Computing Systems
From Bits and Gates to C and Beyond
Second Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

计算机系统概论

(美) Yale N. Patt (得克萨斯大学奥斯汀分校) 著 梁阿磊 蒋兴昌 林凌 译
Sanjay J. Patel (伊利诺伊大学) 著 上海交通大学



本书是计算机科学的经典基础教材。全书以自底向上方法帮助学生理解计算机系统的原理，前半部分阐述了计算机底层结构，后半部分讲解了高级语言编程及编程方法学，主要内容包括数据类型及其运算、数字逻辑、冯·诺伊曼模型、汇编语言、输入和输出、TRAP程序和子程序、C语言编程等内容。

本书可用作高等院校计算机及相关专业学生的入门教材，也可作为的计算机专业人士和高级程序员的参考用书。

Yale N. Patt and Sanjay J. Patel: Introduction to Computing Systems : From Bits and Gates to C and Beyond, Second Edition (ISBN: 0-07-246750-9).

Original language copyright © 2004 by The McGraw-Hill Companies, Inc.

All rights reserved.

Simplified Chinese translation edition published by China Machine Press.

本书中文简体字版由美国麦格劳-希尔教育出版公司授权机械工业出版社出版，未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2006-3131

图书在版编目（CIP）数据

计算机系统概论（原书第2版）/（美）派特（Patt, Y. N.），（美）派特尔（Patel, S. J.）著；梁阿磊，蒋兴昌，林凌译.—北京：机械工业出版社，2007.7
（计算机科学丛书）

书名原文：Introduction to Computing Systems: From Bits and Gates to C and Beyond
ISBN 978-7-111-21556-1

I. 计… II. ①派… ②派… ③梁… ④蒋… ⑤林… III. 计算机系统—教材 IV. TP30

中国版本图书馆CIP数据核字（2007）第082264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：王 璐

北京慧美印刷有限公司印刷 · 新华书店北京发行所发行

2007年7月第1版第1次印刷

184mm × 260mm · 27印张

定价：49.00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

Donald E. Knuth[⊖]: 有关Bottom-Up的论述

(摘自:《The Art of Computer Programming》(Fascicle 1: MMX))

为什么还需要机器语言?

许多读者可能都很困惑:“Knuth在替换MIX的时候,为什么还是选择了一个机器语言,而不是某个高级编程语言呢?现在还有谁在使用汇编语言啊?”

或许持这种观点的读者是有他们的道理的,他们当然可以直接跳过书中涉及机器语言的章节。但我仍将坚持我在20世纪60年代早期本书第1卷序言中曾表述的观点:

本书的主要目标之一是:揭示一个高层结构在现实机器上的实现机制,而不是仅仅介绍使用它的方法。其中所阐述的内容将包括:多个相互关联的程序之间的链接、树结构、随机数生成、高精度运算、radix变换、数据封装、组合查找、递归等内容,即一种自底层逐步向上的论述方法(简称自底向上, bottom-up)。

在我的书中,程序例子都非常简短,原因是:我希望能更多地突出关键思想,引发读者思考。我要对那些希望深入了解计算机的读者说:了解底层硬件的工作原理机制是必需的,否则写出来的程序必然是“莫名其妙的”。

正如书中所述,软件的运算过程及其所表现的输出结果都将表明:机器语言在有些场合是非常必要的。

一些基本算法(如排序和搜索)通常都是用机器语言编写的,因为,只有这样才能保证最佳效率。在它们的编写中,要求我们对Cache、RAM大小等硬件结构有足够深入的研究,如内存访问速度、流水、多发(multi-issue)、后援缓冲、Cache块大小等问题。并且,还要分析它们对程序运行性能的影响,从而在各种可能的实现方法之间作出权衡和比较。

⊖ Donald E. Knuth是著名的《计算机程序设计艺术》(*The Art of Computer Programming*)一书的作者,以下论述摘自该书第一卷“MMIX”的内容片段。——译者注

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、研究及度藏。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”已经出版了近百个品种，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，在“华章教育”的总规划之下出版三个系列的计算机教材：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”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师們服务，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。

这三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，我们的联系方式如下：

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
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

译者序

本书的翻译工作终于完成了，有机会翻译Patt教授的著作让我们感到非常幸运。尽管本书是为大学一年级学生编写的计算机基础教材，但作者在阐述这些基本概念时所站的高度和论述深度都是无与伦比的。我们几位译者都长期从事计算机专业的教学和开发，但看到书中那些我们“已熟知”的概念时，作者所采用的论述角度和方式却让我们有种顿悟的感觉。

Patt教授在他的主页中说到：他热爱计算机教学和科研，但相比科研，教学是他的“first love”。鉴于他在计算机发展历程中的贡献及在计算机科学教育方面的深刻理解和倾心投入，他被IEEE评为泰斗级人物（与《计算机程序设计艺术》的作者Donald Knuth齐名，全球只有他们两人享此殊荣）。Patt教授在计算机体系结构发展史中占有一席之地：他在芯片集成制造方面提出了WOS概念（1965）；在微结构方面设计了HPS结构（综合了超标量、动态调度、乱序执行、预测跳转等多项高级技术）（1984）；提出了著名的“两级自适应跳转预测”结构（1991）。1996年，他被IEEE/ACM授予“在微处理器的指令级并行（ILP）和超标量设计领域杰出贡献”奖。我们敬仰于他在过去60年间为计算机发展所做的杰出贡献，以及他生生不息的创新能力（在写这本书的时候，他正在研究“2009年的计算机应该是什么样的”这一课题）。

有关本书在教学中的使用方法可参考原书序言。作者在第1版前言中提出了六种教学模式，在第2版前言中又将其归纳为三种教学模式，即“一年级两个小学期全授课”、“一年级两学期先高级语言再回顾深入底层”、“二年级再回顾底层”三种模式。我们的总结是，本书分为两部分，前10章（第一部分）讲述的是计算机的组成原理，后9章（第二部分）讲述C语言。

在前半部分内容中，作者以开发的LC-3机器（仿真机器）为背景，讲述了计算机的基本组成和实现（如运算器、指令编码/译码、I/O）等。学生可以通过LC-3机器和汇编器学习汇编，甚至尝试自己的机器设计方法。

本书的后半部分讲述了C语言内容。但其讲述方法与通常的C语言课本完全不同，它在讲述C语言变量、指针、语句、数据结构等内容的时候，结合前10章的知识，阐释了这些高级语言行为怎样转换为机器语言并运行在硬件系统中。其中，涉及很多长期困惑程序员的问题，如函数调用时栈空间（或函数帧空间）的使用规则、给变量赋值常量的几种方法之间的比较、软件中断指令的执行过程等。此外，即使将后半部分内容看作是单纯的C语言教材也无妨，它没有通篇机械地论述C语言的语法规则，而是通过简单计算器、Hanoi塔、排序等问题的求解，轻松地讲述了C语言编程的要点，且非常自然地传授了C编译器的工作原理、递归编程技巧和注意事项、程序调试的原理和技术、数据结构的重要性等“高级”内容。也正如作者在文中提到的，在语言教学中，编程方法学和实现原理比编程语法更为重要，前者传授的是通过编程解决问题的能力，后者则可以通过练习和自学就达到熟练。

总之，我们认为本书的特点是：深入、透彻、全面、易懂。书中虽然没有涉及更多具体、实际的技术知识（如x86），但读完本书后，再看Intel的技术手册，你一定会体会到“一通百

VIII

通”的感觉。

最后，借“译者序”之页，对所有参与和帮助本书翻译的人们表示致谢：本书第1、4、5、6、7、15章及附录A由梁阿磊翻译，第2、3、8、18、19章和附录D及索引由蒋兴昌翻译，第9、10、11、12、13、14、16、17章由林凌翻译，附录B和C由李鸿翻译了初稿。全书由梁阿磊审校。感谢杨建华先生对本书部分章节做出的细致而专业的校对和翻译建议，感谢姜玲燕对部分章节所做的细致校对和修改建议，感谢邹恒明先生为本书所作的序言。您现在看到本书的中文版，也凝结了翻译者的努力。我们尽了最大所能，试图将本书翻译成一本好书。但鉴于能力有限，如有不足和错误之处，敬请谅解和指正。最后，感谢亲人和朋友对本书的热情期待和支持。

译者

2007年2月26日于上海交大

代 序

梁老师请我为本书写序之初我有些犹豫，原因有二：一是我没有给书写序的经验，二是恐怕自己水平有限，写的序不能充分表述或突出这样一本优秀教材的独特之处而影响该书的推广。不过，在经过慎重考虑后，我还是接受了这个挑战。凡事总有头一回报，没有经验不是推脱的理由，而且本人在密歇根大学攻读博士学位的三年间耳闻目睹了该书第一作者Yale Patt的过人学识与非凡个性，随后在上海交通大学任教的三年里又与本书的第一译者梁阿磊博士成为好友至交，写起序来总会比不认识作者和译者的人有些优势。至于水平有限，只能请读者包涵了。

Yale Patt曾任美国密歇根大学计算机体系结构实验室主任多年，被*IEEE Spectrum*称为美国计算机界的卓越泰斗 (luminary)，在美国乃至世界计算机体系结构领域有着广泛的影响力。这本《计算系统概论》不光是密歇根大学计算机专业的经典基础教材，也是美国多所知名大学，如得克萨斯大学、莱斯大学、明尼苏达大学、乔治亚理工学院、伊利诺伊大学和西北大学等校的计算机专业基础教材。

本书第一译者梁阿磊博士是上海交通大学教师，他多年在计算机教学上花费的心血博得了学生的广泛认同，并成为上海交通大学计算机系和软件学院最受学生欢迎的教师之一。梁博士在翻译本书过程中精益求精，既保持了原著的精彩阐述和独特风格，又注入了中文语言所特有的内涵，使译著精辟且顺畅，并成为上海交通大学计算机及相关专业学生的指定教材。

也许你听过这个有启示的故事：在20世纪五六十年代时，美国GE公司的一个大型发电机出了问题。所有的操作人员及工程技术人员面对一大堆的仪表和旋钮一筹莫展。于是他们请来了一位这方面的专家，这位专家看了一下，便拿出一把螺丝刀，将一个旋钮反时针转动了35度，发电机就正常运转了。后来，专家开出的账单为1000美元，GE觉得花两分钟转动一个旋钮就收取1000美元太多，便要求专家出具一份更详细的说明。两天后，专家寄来了新的账单：

将旋钮反时针方向转动35度： 0.75美元

知道应转动哪个旋钮及转动幅度：999.25美元

一本好的教科书不应只教导学生转动A旋钮35度，而是应教会学生为什么要转动A旋钮及为什么只能转动不多不少的35度。这个例子恰恰能印证本书的优点所在。

与多数人的感觉不同，本书不仅简单地讨论计算机组成原理或程序设计语言，而是站在计算机整体系统的高度将软硬件连贯起来进行阐述。本书强调的是理解，而不是死记硬背；强调的是软件与硬件结合，而不是软硬件的分别教授与学习。本书本着将软件硬件教学齐头并进的思路，从硬件的基本构件一直讲到软件的高级程序设计与构造，使学生在学习过程中能够将软硬件融会贯通、相互印证，从而提高学习的广度、深度和效果，并为后续的计算机专业课如操作系统、计算机系统设计与结构、算法设计与分析、高可靠软件工程理论等打下基础。

本书最大的特点是其提倡的层次转换概念，即从问题开始到计算机运算出结果可以分为七个层次。通过七个层次的转换，即可完成从问题到结果的转变。这七个层次及其转换是：问题到算法的转换、算法到程序设计语言的转换、程序到指令集结构 (ISA) 的转换、指令集结构到微观结构的转换、微观结构到电路的转换和电路到电路组件的转换。该书对每个层次

转换进行了深入的阐述，讲解了为什么需要这些转换以及没有这些转换所带来的困难。美国多个大学的教学实践表明，这种层次转换的概念极大地提高了学生对计算机软硬件的理解。

本书在结构上采取的策略是自底向上。在对计算机的哲学原理进行了简要综述后，从晶体管开始，对逻辑门、触发器、逻辑结构和内存实现分别进行了详细的讨论。在此基础上，再对冯·诺伊曼执行模式进行剖析、讲解如何构建一个简单的LC计算机。然后上升到软件层，对计算机汇编语言和高级程序设计语言C进行讲解，重点将程序语言的各种数据类型和构造与底层的硬件构造联系起来，从而使学生在深层次上理解程序语言里各种构造的来龙去脉，及计算机软硬件之间的有机的、千丝万缕的联系。

本书的使用方式有多种，其中适合中国国情的方式有以下四种：

1. 密歇根模式：作为计算机软硬件原理的入门教材，无需先修课程。该模式要求一个学期内覆盖本书全部内容，时间安排为本科第一学年的第一学期。该模式工作量大，挑战性高，非常适用于肯钻研、进取心强的学生。
2. 正常模式：作为软硬件专业入门教材，无需先修课程。该模式要求在一个学期覆盖大部分内容，但可以跳过10.3、10.4节、第16章、第18章和第19章。时间安排为本科第一年的第二学期。该模式较密歇根模式来说负担稍轻，但仍具有相当的挑战性。
3. 第二模式：在学生已经进行了专业入门教育如程序设计、数据结构和数字电路等课程的学习后使用本书作为第二课程的教材。这种模式在一个学期覆盖本书全部内容，而讲课时间则安排在本科二年级上学期。该模式适合于计算机专业的大多数学生。
4. 分治模式：将这本书的内容分为两个学期讲授。第一学期讲授第1~10章，第二学期讲授第11~19章。时间安排为本科一年级下学期和二年级上学期，这是使用本书的最优方式。

本书结构紧凑、风格鲜明、内容精辟、条理清晰、文笔流畅、语言优美，读起来引人入胜，爱不释手。本书还引入了大量的典故和比喻（当然，有些典故和比喻需要西方及美国文化背景才能更好地领悟），让人感觉到不是在读一本计算机教科书，而是在读一本幽默有趣的长篇小说，或者借用Donald Knuth的话，它是“一部蓝色的诗歌”。

本人强烈推荐本书给高校计算机专业的师生和所有对计算机软硬件有着浓厚兴趣的人士。读完本书后，你对计算机系统的理解将会达到一个崭新的境界。相信我，你不会失望的。

邹恒明

2007年3月2日于莘庄

第2版前言

本书自第1版发行至今已有三个年头，期间我们收到了大量学生和教师的反馈信息，几乎所有的反馈信息都是“正面的”，这让我们备感欣慰！能有机会出版本书的第2版，实在是让人高兴的一件事情。之所以高兴，是因为大多数人都赞同本书的编写方法，尤其是我们收到的反馈信息都来自学习第一线的读者——它们包含了学习者（学生）的感受和授业者（教师）对本书使用效果的评价，从他们的E-mail中能看出对本书的评价都很高。

但正如本书第1版前言中所提到的，本书仍然在追求不断地完善。另外，在接受赞誉的同时，我们也收到很多建议，这些建议指出怎样使本书写得更好，我们在此向所有关心本书的读者深表感谢。同时，由于第1版出版后我们两人都分别在课堂上讲了两遍以上，也发现了很多需要修改和改进的地方。因而我们决定在第2版中做出较大的修改，但同时也期望新版能延续第1版编写的初衷，并期待尽快听到读者的反馈。

修订内容

LC-3

本版最大改动之一就是，采用LC-3 (Little Computer 3) 结构替换了第1版的LC-2机器模型。实际上，新结构保持了LC-2的基本概念，即一个易于描述又有望在短期内掌握的丰富的ISA。指令仍然采用16位宽度，其中操作码为4位。正如我们的一个学生指出的，事实上，子程序返回指令 (RET) 就是LC-2的JMP指令的特例，因而在LC3中，我们取消了RET指令作为单独操作码的定义。在LC-3中，我们只定义了15种操作码，预留了一个用于未来应用（或许在第3版中会定义这个操作码）。

其次，我们收到很多反对PC-concatenate寻址模式的建议，特别是针对分支跳转指令。该寻址模式起源于20世纪60年代中期的PDP-8机器，问题表现在当前页中的一条指令访问下一页（或之前页）之时。这个问题很让人头痛，尤其是在接近页边界时出现的前向跳转指令。因而，很多人建议我们采用现代机器中常用的“PC+offset”方式。因此，我们将第1版中所有使用“PC偏移 (offset)”之处，都改成了“PC+SEXT (offset)”。

另外，新版还包括以下修改：

- 栈空间增长改为向0地址增长（符合现在的实际惯例）。
- LDR/STR指令中的偏移改成了有符号值（因而可以基于base地址做向前、向后偏移）。
- 操作码1101现在未做定义。
- JSR/JMP的操作码做了轻微改动。
- 最后，条件码扩展为16位处理器状态寄存器 (Processor Status Register, PSR，其中包含了特权模式和优先级字段)。

与第1版相同，在附录A中有LC-3结构的完整描述。

扩充内容

在第1版的基础上，新版几乎对所有章节都做了或多或少的修改，有些章节的改动大一些。

其中：

- 第1章中增加了两方面的内容（1）有关抽象方法的重要性及其本质性的描述；（2）硬件和软件之间相关性的讨论。
- 第3章增加了一节，即有关有限状态控制及顺序开关控制的实现方法。因为这些实现方法和技巧已成为计算机科学（CS）或计算机工程（CE）专业学生必须掌握的知识，这些知识的掌握将有助于对第4章冯·诺伊曼模型的深入理解。
- 第4章中增加了对LC-3微结构的概述，相关内容的详细描述参见附录C。
- 第5章，有读者反映内容太扼要，在此我们增加了一些素材，希望新增的这些图和注释会使相关概念更加清晰。
- 第8章和第10章中增加了有关中断驱动I/O的章节内容。
- 第11~14章介绍了C语言（与第1版一样），不同之处是，这些章节更加注重帮助编程初学者掌握语言的基本内容，而将那些特性内容移至该章结尾或附录D之中，使之脱离教材的主线。所有这些章节都增加了更多的例子。

另外，第2版改用“问题解决驱动”方式来讲述编程技术，即通过例子展示为什么新引入的C结构可以在C编程中解决当前的问题（这将有助于读者真正理解编程语言的设计精妙之处）。

- 第14章中还将介绍新的LC-3调用规范，新的规范更接近实际系统中所采用的规范。
- 第15章介绍了有关测试和调试的深入知识。
- 章节顺序的调整：根据来自教学中的经验，我们决定调换“递归”和“指针与数组”两章的顺序。从而使学生在学习“递归”内容（相对较难，目前推后至第17章）之前有更多的机会熟悉和掌握C语言的编程方法。

仿真器

针对新的LC-3结构，Brian Hartman对原先的Windows版仿真器（Simulator）做了修正。Ashley Wise则为UNIX平台编写了LC-3仿真器。在两个版本的仿真器中，都实现了“中断驱动I/O”功能。我们相信没有任何方法比“实际动手”更有助于知识的真正掌握。具备了中断驱动I/O功能的仿真器，使得学生现在可以实验这样一个场景，即通过键盘输入中断一个正在执行的程序，并调用对应的中断服务程序。

本书的使用方法

本书是计算科学相关专业新生的入门教材。我们认为“自底向上”方法是帮助学生理解技术计算原理的最好方法（参考后面第1版前言中的详细论述）。事实和经验表明，学生在掌握了计算机底层工作的原理机制之后，能更加从容地解决以后可能面临的新问题，包括高级编程语言方面的问题。而且，这些学生学习编程语言的方法是“理解式”而不是“记忆式”的，因为一切动作都是明明白白的。从我们的角度来看，本书的最佳使用方法是，开设一个学期的新生主干课程，或是两个学期的课程（节奏慢些）。如果教学的重点是单学期制的高级编程语言，则串行机和中断驱动I/O的内容可以跳过；而如果教学的重点是本书的前半部分，则第15、17、18和19章的内容可以跳过。

通常，这本书的参考使用方法包括：

- 一年级第一学期作为新生主干课程。

第一种方法是在一年级第一学期内分两个小学期完成，这可能是本书最好的使用方法。在第一个小学期，讲授本书的第1~10章，在第二个小学期，讲授第11~19章。这样的进度有些

快，但在两个小学期（即一学期）内是可以讲解完整本书的（注：一个学期（semester）分为两个小学期（quarter））。

- 一年级第二学期作为计算机的第二门课程。

第二种方法还可作为计算的第二门课程在一年级第二学期完成，学生在之前的第一学期已修完了第一门高级编程语言方面的课程，之后在这门课程中，则回头学习初级的数字逻辑、基本的计算机组成和汇编语言编程等底层知识。这种方式下，这一学期的大部分时间都用于学习第1~10章，只留下最后几周学习第11~19章的一些课题，以显示学生从第一门高级程序设计语言课程中学到的机理是如何在底层实现的。其中，通常会讲到函数、活动记录、递归、指针变量和其他基本数据结构等课题。

- 二年级计算机组成课程（Sophomore-Level）。

本书还可用于二年级要上的深入探究计算机实现的课程。在这期间，重点学习第1~10章，有时还需彻底学习附录C（LC-3之微结构和微编程实现）。需要指出的是，本书未讲述计算机体系结构中的一些重要概念，如高速缓存、流水线、虚拟存储器等。我们认同这些概念对于有志成为计算机科学家或计算机工程师的学生至关重要，但同时又感到这些内容更适合放到计算机体系结构与设计的更高级课程中来讲。本书的目的并不在此。

致谢

本书需要向太多的人表示致谢，感谢他们为本书所做的重要贡献。其中，尤其要感谢Brian Hartman和Matt Starolis。

Brian Hartman先生以他的积极热情和技术专长继续为本书新版做出重要贡献。1996年冬，他在密歇根大学作为一年级本科生选修了这门课。他随后在本科期间多次担任这门课的助教，并在硕士学习期间编写开发了Windows版LC-2仿真器（同时这也是他的硕士毕业课题）。最近，他又为LC-3开发了Windows版仿真器。现在，他已离开学校三年多了，但仍在坚持维护该系统。

Matt Starolis在两年前作为UT（得克萨斯大学）一年级学生选修了该课程，并于去年秋季担任该课程助教，在本书第2版的编写中担任了重要角色，为本书提出了许多意见并设计了书中的多幅插图。他还修订了LC-3仿真器的使用手册，以使之与新版仿真器一致。往往在任何事情需要有人去做的时候，他总是第一个站出来。他的热情为本书注入了活力。

目前本书有大约100多位采用者，我们定期地收到来自世界各地的教授们热情洋溢的E-mail。他们是Vijay Pai (Rice)、Richard Johnson (Western New Mexico)、Tore Larsen (Tromso)、Greg Byrd (NC State)、Walid Najjar (UC Riverside)、Sean Joyce (Heidelberg College)、James Boettler (South Carolina State)、Steven Zeltmann (Arkansas)、Mike McGregor (Alberta)、David Lilja (Minnesota)、Eric Thompson (Colorado Denver)、Brad Hutchings (Brigham Young)。

至于我们两人，自第1版出版以来，共使用本书教授了4届学生，并培养了一大批热情、积极的助教和学生。其中，助教包括Kathy Buckheit、Mustafa Erwa、Joseph Grzywacz、Chandresh Jain、Kevin Major、Onur Mutlu、Moinuddin Qureshi、Kapil Sachdeva、Russell Schreiber、Paroma Sen、Santhosh Srinath、Kameswar Subramaniam、David Thompson、Francis Tseng、Brian Ward、Kevin Woley，而Linda Bigelow、Matt Starolis、Lester Guillory等人一开始都是新生，两年后已成为最积极的助教。

Ashley Wise开发了Linux版本的LC-3仿真器，Ajay移植了LCC编译器来产生LC-3代码。

Gregory Muthler和Francesco Spadini对本书后半部分的原稿提出了积极的反馈意见。Brian Fahs为习题提供了解答。

Kathy Buckheit编写了LC-2仿真器的入门材料（因为她觉得这很有必要）。

得克萨斯大学（University of Texas）许多同仁都采用了本书，并与我们分享他们的见解，如ECE专业的Tony Ambler、Craig Chase、Mario Gonzalez、Earl Swartzlander和CS专业的Doug Burger、Chris Edmundson、Steve Keckler。在此表示感谢。

我们还要继续感谢认真负责的编辑们，他们是Betsy Jones和Michelle Flomenhoft。

自第1版以来，我们的书还受益于很多大学教师的审核，感谢他们：Robert Crisp (Arkansas)、Allen Tannenbaum (Georgia Tech)、Nickolas Jovanovic (Arkansas-Little Rock)、Dean Brock (North Carolina-Asheville)、Amar Raheja (Cal State-Pomona)、Dayton Clark (Brooklyn College)、William Yurcik (Illinois State)、Jose Delgado-Frias (Washington State)、Peter Drexel (Plymouth State)、Mahmoud Manzoul (Jackson State)、Dan Connors (Colorado)、Massoud Ghyam (Southern Cal)、John Gray (UMass-Dartmouth)、John Hamilton (Auburn)、Alan Rosenthal (Toronto)、Ron Taylor (Wright State)。

最后，还要感谢那些以独特方式做出贡献的人们（在此不再赘述他们的具体贡献，只是列出他们的名字）：Amanda、Bryan、Carissa Hwu、Mateo Valero、Rich Belgard、Janak Patel、Matthew Frank、Milena Milenkovic、Lila Rhoades、Bruce Shriver、Steve Lumetta、Brian Evans。另外，Sanjay（作者之一）还要感谢Ann Yeung的爱和支持。

结束语

再次重复第1版前言中的话：本书仍然在追求不断地完善，期待你们提出宝贵意见。我们的联系方式是：patt@ece.utexas.edu和sjp@crhc.uiuc.edu。

Yale N. Patt
Sanjay J. Patel
2003年3月

第1版前言

本书内容取自密歇根大学开设的EECS100课程——这是计算机科学（CS）、计算机工程（CE）和电子工程（EE）三个专业的第一门计算机类主修课程。该课程由Kevin Compton教授和本书第一作者于1995年秋季开设。

之所以开设EECS100，是因为多年来计算机科学与工程系的教师认为，他们的本科生对计算机基本概念缺乏深入的理解。例如，学生们无法清楚地解释指针变量，而递归概念对他们来说就像是在“变魔术”，难以理解。

所以，我们在1993年提出对传统教学思路的改革。传统的教学思路是从高级计算机语言开始入手，但是在这种教学方式下，学生仅仅是“记住”了其中的技术细节，并不能真正“理解”其原理、机制。本书的教学思路是“自底向上”（bottom-up）：从MOS晶体管开始，依次介绍逻辑门、锁存器、各种逻辑结构（如MUX、解码器、加法器、门控锁存器），然后通过存储器（或内存）的实现案例，将以上概念有机结合。随后，介绍冯·诺伊曼模型，并以简单的LC-2计算机为背景，介绍LC-2机器语言编程和汇编语言编程；再之后，我们继续上升到高级语言（如C语言），以及递归、指针、数组等概念；最后，引入一些基本的数据结构。

有一种“信息隐藏”（Information Hiding）学习方法，但我们不太赞同这种方法。事实上，“信息隐藏”确实是个很有效的学习方法[⊖]。但我们认为，只有在你真正搞明白它在做什么之后，信息隐藏方法才具有意义。换句话说，我们计划将每个概念的底层动作都暴露、表现出来，消除在上层感受到的神秘感。

需要指出的是，虽然本书采用的是“自底向上”方法，但并不意味着我们反对“自顶向下”（top-down）方法。相反，我们认为“自顶向下”在设计方面是非常正确的方法。在设计中采用的方法，与在学习中采用什么样的方法，是两件不同的事情。就是说，对前者我们主张采用“自顶向下”方法（前提是设计者已深入理解底层构造），而对后者（即学习和理解过程）我们则主张“自底向上”方法。

内容概要

本书内容包括两个部分：一是计算机底层结构（LC-2计算机），二是高级语言编程（C语言）。

LC-2

本书的前半部分偏重计算机底层机制。从底层基础知识开始，逐步上升到操作系统和高级语言程序的接口层，以便能够理解真实计算机的工作原理。

- 第2章：介绍基于位（bit）的算术和逻辑运算操作，以及相关的结构设计（即LC-2计算机的组成）。
- 第3章：存储器设计。从MOS晶体管开始，逐步介绍存储器的实现过程。该存储器很简

⊖ 所谓“信息隐藏”方法，即在学习新概念时，不要过分追究细节。——译者注

单，每字宽度为3-bit，共4个字大小（而不是64MB）。其中，每个部件的设计图都不超过一页大小（以方便阅读）。至此，构建存储器所要求的知识就全部介绍完了。

- 第4章和第5章：介绍冯·诺伊曼模型及LC-2计算机的组成结构。LC-2是一个16-bit结构的计算机，也是冯·诺伊曼模型的一个具体实现。它具备以下功能和特性：
 - 键盘和显示器等物理I/O设备。
 - 调用操作系统服务的TRAP机制。
 - 基于N、Z、P等条件码的条件跳转。
 - 子程序调用和返回机制。
 - 基本的操作指令集合（ADD、AND和NOT）。
 - 各种加载（load）和存储（store）寻址模式，如直接、间接、基址+偏移和立即数等有效地址计算方式。
- 第6章：介绍编程方法学和调试方面的知识。
- 第7章：介绍汇编语言编程，以及我们开发的LC-2仿真器和汇编器。事实上，我们开发了两个版本的仿真器，一个是在Windows平台上运行，另一个是在UNIX平台上运行。Windows版本的仿真器程序可以从网上下载或从本书光盘获取，偏好UNIX的同学，可从网上免费下载UNIX版仿真器，然后安装。

学生使用仿真器的目的是测试和调试用LC-2机器语言和LC-2汇编语言编写的程序。它支持在线调试（即内存检查、单步、断点设置等操作）。该仿真器仅适合于调试简单的LC-2机器语言或汇编语言程序，其主要目的是帮助学生理解和掌握第1~10章中的基本概念。

有关汇编语言，我们仅限于介绍，而不是熟练编程。汇编的编程技术是更高级课程（而不是一年级课程）的任务。之所以在第7章介绍汇编语言知识，是因为它在“自底向上”层次中独占一层。通过它，学生可以观察到从汇编语言到0/1序列的转换过程。并且，通过手工汇编（即手工转换），虽然花费了大量时间，但必将为学生打下坚实的概念基础，令他们受益终生。

所谓汇编语言，可以将它看做是机器指令的一种“友好”表示方式，它的好处在本书后半部分将体现出来。因为从第11章开始，我们将借用汇编语言或机器语言来解释C语言语句的底层含义。当然，汇编语句“ADD R1,R2,R3”显然比机器代码“0001001010000011”具有更好的可读性。

- 第8章：讨论物理设备的输入（键盘）和输出（显示器）。
- 第9章：讨论操作系统的陷入机制（TRAP），以及子程序调用和返回机制。学生将学习用LC-2代码编写的操作系统服务程序，以便生成由TRAP指令调用的物理I/O。
- 第10章：是本书上半部分的总结。通过“计算器程序”例子，讲述栈和数据转换的原理和机制。该程序由1个主程序和11个子程序组成。

C语言

本书下半部分内容是C语言编程，但不是对于C语言的一般性介绍，而是对其内部机制的深入洞察（学生之前已对C语言底层有所了解）。

作为一种高级语言，C语言最适合于我们的“自底向上”方法，因为C语言是各种高级语言中“最低级的”。通过它，能够清晰地表现软件和硬件的接口关系。本书的C语言学习内容，着重于控制结构语句、函数、数组等基本概念；在基本掌握编程概念之后，高级概念（如对象、抽象等）的学习就是轻而易举的事情了。

在学习过程中，每介绍一个C语言结构，我们都将给出其对应的LC-2汇编代码（即编译