



国外经典教材·计算机科学与技术

PEARSON
Prentice
Hall

A Balanced Introduction to Computer Science

新编计算机科学导论

- ◆ 以创新的视角，透彻介绍概念与方法
- ◆ 以新颖的选材，全面阐释知识与应用
- ◆ 编排体系灵活，有利于调整教学顺序
- ◆ 大量动手练习，侧重于提高应用技能

(美) David Reed 著
张玉芳 孙 思 译



清华大学出版社

国外经典教材·计算机科学与技术

新编计算机科学导论

(美) David Reed 著

张玉芳 孙 思 译

清华大学出版社

北 京

内 容 简 介

这是一本独具特色的计算机入门教材。传统的书籍要么侧重于广度(概述所有计算概念),要么侧重于深度(程序设计以及解决问题)。本书共18章,同时兼顾了广度和深度。广度章节包括了对初学者非常重要的计算概念以及问题,主题涉及计算机术语、因特网和万维网、算法和软件以及技术对社会的影响。深度章节则穿插于这些主题之间,借助于动手练习来介绍基础的程序设计概念和技巧。通过使用html和JavaScript,读者可在设计并实现交互式网页的同时,发展解决问题的技巧。

本书适用于“计算机科学导论”课程。同时也适合打算涉足计算机行业的人员自学,为以后的程序设计、网站开发和技术支持生涯奠定良好的基础。

Simplified Chinese edition copyright © 2005 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: **A Balanced Introduction to Computer Science, by David Reed,**
Copyright © 2005

EISBN: 0-13-046709-x

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2004-5630

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

新编计算机科学导论/(美)里德(Reed, D.)著;张玉芳,孙思译. —北京:清华大学出版社, 2005.3
(国外经典教材·计算机科学与技术)

书名原文: A Balanced Introduction to Computer Science

ISBN 7-302-10480-8

I. 新… II. ①里… ②张… ③孙… III. 计算机科学—教材 IV. TP3

中国版本图书馆 CIP 数据核字(2005)第 011261 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

客户服务: 010-62776969

文稿编辑: 文开棋

封面设计: 久久度文化

印 刷 者: 北京世界知识印刷厂

装 订 者: 三河市化甲屯小学装订二厂

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 21.75 字数: 515 千字

版 次: 2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷

书 号: ISBN 7-302-10480-8/TP · 7111

印 数: 1 ~ 4000

定 价: 37.00 元

出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，急需一批门类齐全、具有国际先进水平的计算机经典教材，以适应当前我国计算机科学的教学需要。通过使用国外先进的经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培育出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构，Pearson 在高等教育领域有着不凡的表现，其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系，这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为这套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

2004.03.20

国外经典教材·计算机科学与技术

编 审 委 员 会

主任委员：

孙家广 清华大学教授

副主任委员：

周立柱 清华大学教授

委员（按姓氏笔画排序）：

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

译者序

接触过很多关于计算机科学的介绍性书籍，当我看到这本书的名字时，心中不禁产生这样的疑问：何为广度？何为深度？所谓的“均衡”指的是什么？与其他计算机科学导论教材相比较，“均衡”又能带来哪些好处？

作为读者，也许对计算机科学一无所知，也许只了解其中的一个或几个方面，也许只希望对计算机科学有一个总体的认识，也许想深入学习计算机科学。怎样才能兼顾如此种种的“也许”？本书提供了一个理想的解决方案：在介绍计算机科学领域各个方面使读者全面了解计算机科学的同时，提供编程概念、语言和基本技巧等较深入的内容。

这是一本与众不同的计算机科学入门教材。它针对不同的读者，提供了相当大的自主空间。不同喜好和要求的读者，均能从中了解到自己迫切需要的知识。这就是均衡的好处，可以同时满足读者对广度和深度的要求，在纵、横两个方向深入概念和技能。

参加本书翻译工作的有张玉芳、孙思、张磊、杨炯明和任芳。张玉芳副教授负责和组织了全书的翻译，并翻译了序、前言、目录、第1章至第7章。孙思翻译了第6章至第9章。张磊翻译了第10章、第11章。杨炯明翻译了第12章至第14章。第15章至第17章由任芳翻译。译稿全文由孙思负责校对。

在此，要感谢清华大学出版社的策划和编辑人员，他们的不懈努力使本书得以顺利出版。

由于翻译时间较为仓促，翻译水平有限，翻译中的不当之处也在所难免。译文中的错误当然应当由译者负责。真诚地希望同行和读者朋友不吝赐教。

译者

2005 年春

前言

欢迎选用《新编计算机科学导论》。你可能出于很多原因阅读这本教材。也许是使用计算机的时间不多，希望进一步了解它们的工作原理，以及如何去控制它们。也许是认识到编程和计算机文化的市场潜力，想扩展自身的技能以利于将来就业。也许仅仅只是对万维网心存好奇，想知道被光环笼罩着的它究竟是怎么回事。无论如何，你将开始一种新的旅程，我希望它能像我希望的那样令人激动、充满挑战，最后让你硕果累累。

深度和广度的平衡

与绝大多数计算机入门教材不同，本书力求在计算广度和编程深度之间维持一种平衡。传统的入门级教材基本上都是集中介绍其中的一个方面。基于广度的教材着重强调对计算机和计算机科学的广泛理解。通过广泛介绍诸如计算机结构、图形学、网络以及当前社会的一些技术，让学生对该领域有一个广泛的认识，并且拓展学生的视野以利于将来正确理解和评价现实生活中技术所起的作用。基于深度的书籍更着重于强调计算机编程的作用。编程的学科不仅可以培养解决问题的能力，同时也是计算机科学众多领域的重点，因此能够意识到它们的重要性是非常重要的。

虽然每一种方法都有其自身的优点和价值，然而走向任何一个极端都具有潜在的不足。基于广度的介绍可能太肤浅，向学生陈述一种泛泛的观点，使得缺乏背景知识和实际经历的他们不能够充分地理解。虽然编程深度可以提供许多计算概念上的认识，但培养熟练的程序员和问题解决者还需要大量的实践经验(尤其是在学习复杂的编程语言,比如 C++ 或者 Java 时),而这并不是直接同所有的学生都有关系的。

本书所采用的方法在深度和广度间进行了平衡。书中包含了与计算机的概念和主题相关的章节，直接与初学者紧密相关，包括计算机术语、因特网和万维网、计算机的发展历史、计算机技术的组织结构和制造工艺、计算机技术对社会的影响。在穿插于广度主题之间的章节中，以实用的指南形式穿插介绍了基本的编程概念和技巧。学习编程语言 JavaScript，可以培养学生设计和实现交互式网页的技能。由于 JavaScript 简单、有自然的接口以及同网页的无缝集成，使初学者能够迅速开发出有趣而迷人的程序。此外，拥有 Web 浏览器的用户都能够免费获得 JavaScript，从而轻易将本书所学到的编程技巧运用到日常问题中。

为了突出深度和广度的平衡，本书并不旨在完整而详尽地介绍计算机，也不仅仅是作为 JavaScript 的一本参考书。广度方面的章节集中介绍与初学者相关的关键思想和概念，借助这些思想和概念，他们可以更好地理解计算机技术和计算机科学领域。同样地，编程方面的章节集中介绍 JavaScript 的特性。这些特性说明了基本的编程概念，同时也介绍了

有趣而迷人的应用实例。对于感兴趣的读者，还提供了章末的补充材料和练习。它提供了对计算机的广泛认识，同时也提供了足够的问题求解和编程深度，使学生充分认识到计算机科学的重要性。

本书的目的

本书及其附带的资源有三个主要的目的。首先，向学生揭示计算机科学领域的广度。计算机科学只不过是计算机的研究——它侧重于与计算机有关的所有方面，从算法(实现任务所需的一步一步的指令序列)的设计和分析，到计算机零部件的工程和制造，到软件系统的开发。通过查阅和使用网上的资源，学生们将学习以下主题：计算机技术的历史、现代计算机的内部结构、翻译和执行程序序列以及计算机的能力和限制。使用软件模拟器，他们可以构建计算机的虚拟部件，查看程序在低级机器上翻译和执行的流程信息。结合阅读和试验，这些概念栩栩如生地展现在学生面前，并让他们清楚地认识到计算机科学的“真面目”。

本书的第二个主要目的是教授编程原理。编程是在计算机上解决问题的过程，也就是设计指定任务的答案，然后以计算机可以理解和执行的语言形成这些答案。编程是计算机科学的中心任务，为该领域许多有趣的方面和挑战提供突破口。在学习编程的过程中，他们将学会分析问题，逻辑思考，形成自己的思想，从而解决问题。这是一门学科，因为必须学习系统的方法，但同时它又是一个创造性的过程，因为必须寻求新奇的方法来解决新的问题。因为在编程过程中形成的许多技巧都可以用于一般问题的求解，从本书所获的经验也可以用于继续学习其他学科。最后，采用交互式 Web 页面为背景写程序，这是为了强调编程与学生每天使用的应用程序之间的联系。

本书的第三个主要目的是解释说明计算机科学及其跨学科特性。在各种领域中的分析研究表明，数学和自然科学最为显著，现在越来越离不开计算机了。分析和调查这些领域的应用，比如生物、物理、心理学、甚至是经济等领域，学生们将学会把自己的编程技巧应用到广泛的问题中。此外，他们还将形成所有科学通用的经验技巧。

本书特点

本书对计算机科学和程序设计采用的均衡介绍，体现在本书的章节布局中。本书有两类章节，一类是叙述介绍计算机关键概念(即计算机科学的广度章节：第 1, 3, 6, 8, 10, 12, 14, 16 和 18 章)，另一类是用教程风格来培养求解问题和编程技巧(即计算机科学的深度章节：第 2, 4, 5, 7, 9, 11, 13, 15 和 17 章)。这些章节相互交叉既是特意安排的也是重要的。它为学生求解的任务提供了多种类型，因此更能满足具有不同学习风格的学生。阅读和课堂讨论是程序设计教程的缓冲，在开始下一个教程之前，允许他们有更多的时间来消化吸收编程的概念和技巧。最后，也许最重要的一点是，各个章节之间的交叉有

助于他们理解和学习上下文。例如，读者在第 2 章开发了自己的主页之后，就为在第 3 章理解什么是 Web 及其工作原理做好了充分的准备。

计算机科学广度章节的特点

- 它们侧重于与初学者最相关的主题。其目的并不是使用细节介绍来淹没学生，而是强调该主题的中心思想。
- 尽可能采用图表来阐明重点内容。
- 提供基于 Web 的可视化工具(请访问本书配套网站：<http://www.prenhall.com/reed>)以补充许多章节，有助于积极地学习。例如，第 14 章集成了一组模拟器，它允许学生研究计算机的内部工作原理。
- 每一章结束都有章末总结和复习题，它鼓励思考和综合该章的内容。

计算机科学深度章节的特点

- 这些章节以教程风格呈现，动手实践是学习编程(和一般的解题)的惟一途径。
- 书中的练习采用循序渐进的方式，首先让学生通过学习现有程序掌握编程概念(请访问本书配套网站：<http://www.prenhall.com/reed>)，然后使用新的工具和构造进行修改。最后，为感兴趣的和希望解决的问题创建新程序。
- 把常见错误和易混淆的地方识别出来，并在特色段落“需避免的常见错误”中专门讨论。
- 在特色段落“设计者秘诀”中专门提供了与求解问题有关的程序设计的建议。
- 每一章都包括一个章末小结，以简洁的项目列表提出了关键概念和编程工具。
- 为了深入学习，在每一章结束时提供了补充材料和练习。

在本书的末尾还提供了附录以供参考。附录 A 和 B 介绍了常用的 Web 浏览器和常用文本编辑器，这对于还不熟悉计算机和 Web 的学生来说或许很有用。附录 C 是 HTML 参考，以表格的形式收集了本书使用的所有 HTML 元素。附录 D 是 JavaScript 参考，收集了所有的 JavaScript 编程结构。附录 E 提供了 random.js 库的完整列表，它在第 5 章引入，并在随后的章节中一直使用。

最后，收集了 9 个做试验的课外作业，每一个都对应相关的深度章节，是对本书的补充。这些作业强调试验、分析和使用程序来解决各学科间的问题。

给学生的建议

本书并不假定你在计算机应用和编程方面具有任何经验。当然，熟悉计算机(比如字处理软件或电子邮件)是很好的，但并不是必需的。本书包含了所需的基本的计算机术语，提

供的附录是为了帮助只具有简单计算机技巧的新手，比如使用文本编辑器，保存文件到磁盘，浏览网页。本书的目的并不是要教会你想知道的计算机和编程方面的所有知识，而是为你提供一套实用的技巧和知识。

通过阅读、练习和试验，本书希望能为你提供计算机科学的广泛认识，同时从深度上将你培养成问题解决者和程序员。专门选择 JavaScript 作为培养编程技巧的媒介，是为了使问题更简单，与学生的联系也更紧密。JavaScript 是一种简单的脚本语言，用于控制万维网中的页面。使用 JavaScript，你可以通过单击鼠标来控制行为，或在页面上生成动态图像。同样地，学会了 JavaScript 就为你打开了 Web 之门，在这上面有许多激动人心的应用。JavaScript 与 Java 和 C++ 有类似的地方，这表明 JavaScript 编程经历是使用高级语言进行大规模编程的跳板。

不管你是选择继续学习计算机科学，还是只希望在日常生活中使用一些计算机技巧，本书关于计算机主题的均衡介绍对你是有价值的，这一点将得到证明。一如既往地享受和学习吧！

给教师的建议

本书设计的章节布局给教师提供了最大限度的灵活性。根据教师的喜好和指定课程的目的，材料的平衡和顺序可根据你的需要自行决定。如果你教的是传统的非专业课，那么深度和广度之间的均衡可能更为重要。如果学生具有较广的计算机知识，把这门课程作为计算机科学系列的一部分，那么可以跳过或者缩减一些广度方面的主题，而考虑进一步的编程深度。

例如，我在 Creighton 大学的非专业课中使用了本书。因为学习本课程的大多数学生并不打算成为计算机专家，本课程的重点是在重要的主题和基本的编程概念之间提供均衡的介绍。同样地，在深度和广度之间大约是 50/50 的比例。只包含了前面的 6 个编程章节(第 2, 4, 5, 7, 9 和 11 章)，每一章都足够传授编程技巧，也让学生做一些感兴趣的和迷人的事情。穿插于这些编程章节之间大多数是广度方面的章节，每一章需要花一到两节课(50 分钟)来讲授或讨论。并不是每学期都包含本书的所有广度章节，顺序也会变化。例如，最近一学期，我跳过第 8 章，移到第 12 章。此外，我在上课时间有 4 堂实验课，它们有助于将学生的编程和解决问题的技巧运用到各学科间的问题。

与之相比，在 Dickinson 学院，这些材料的使用方式却完全不同。这门课首次上连堂，实现了学院对实验室的科学利用，也被认为是计算科学的第一门专业课。像这样的情况，学生需要更多的时间和精力。每周有两个小时的实验课，让学生完成实验作业。编程章节每次花两到三节 50 分钟的课时，广度章节每次花一到两节 50 分钟的课时。

本书的优势之一是它提供给教师的灵活性。教师可以只选择深度或者广度章节，或者选择深度和广度间不同的平衡度来满足他或她的学生。例如，我猜想许多教师可能选择第 13 章(循环)，也许会取代课堂实验或某些广度章节。深度和广度章节的交叉贯穿于本书，这是专门设计的，它有助于学生学习和教学的灵活多变。当所包含的编程方面的深度章节

的相关顺序受到其内容的限制时，通常可以省略或去除一些计算机科学的广度章节。例如，一些教师可能喜欢在课堂上先讲授计算机的历史(第 6 章)，或者计算机的工作原理(第 14 章)和内部结构(第 16 章)。

一如既往地享受、教学和学习吧！

致 谢

本书的创作经历了一个思考、试验、实现、课堂测试、再思考、再试验这样一个反复的长期过程。在撰写本书的过程中，很多人做出了贡献，既有行动上的支持也有精神上的支持。做出显著贡献的有 Prentice Hall 的编辑 Petra Recter 和 Kate Hargett，以及制作编辑 Irwin Zucker。我永远感激策划编辑 Lauren Trees，帮助我形成自己的写作风格——本好书当然应归功于她。我还要感谢以下评审员，他们深思熟虑的评价为我提供了很大的帮助。

Debra Burhans, *Canisius College*
Martin Chetlen, *Moorpark College*
Donald Costello, *University of Nebraska*
Erica Eddy, *University of Wisconsin – Parkside*
Karen Ehrlich, *SUNY College at Fredonia*
Michael Hennessy, *University of Oregon*
Rowan Lindley, *Westchester Community College*
Ronald Marsh, *University of North Dakota*
David Middleton, *Arkansas Tech University*
Arnie Miles, *Georgetown University*
Vince Offenback, *North Seattle Community College*
Roger Priebe, *University of Texas – Austin*
Mary Ann May-Pumphrey *DeAnza College*
Charles Riedesel, *University of Nebraska – Lincoln*
Patrick Sebrechts, *California State University – San Marcos*
Jim Schmolze, *Tufts University*
Gene Sheppard, *Georgia Perimeter College*
David Valentine, *Slippery Rock University*
Mark Williams, *Lane Community College*

在 Dickinson 学院形成了本书的雏形，后来又受到 Nancy Baxter-Hastings 开展的试验课的鼓舞。Dickinson 学院的许多同事讲授了材料的雏形，并贡献了新的思想，尤其是 Grant Braught 和 Craig Miller。除了那些早期的同事以外，我还要感谢 Creighton 大学的同事，他们也对本书的课程和材料给予了支持。我要衷心感谢我的好朋友 Owen Astrachan，他是我早期执教的榜样，因为我在许多公共场合通常会感到局促不安。

还要感谢养育我的父母，他们让我体会到知识和学习的价值，并支持我的学业。在我撰写本书的过程中，我的妻子 Laura 也做出了巨大的牺牲，她经常和我工作到深夜，并听取我无尽的唠叨——征求她的意见：在课堂上哪些地方需要哪些地方不需要。她不仅乐意听取，而且提供了敏锐的见解，让我在家振作精神而不至于松懈。她的牺牲使本书的面世成为可能。最后，特别感谢我的孩子们 Charlie 和 Jack，因为他们给予了我无限的爱和耐心。现在本书已经完成，爸爸或许可以休息一下。清晨，他又将像啄木鸟一样辛勤工作。

Dave Reed
于 Creighton 大学

目 录

第1章 计算机基础	1	本章小结	85
1.1 什么是计算机	1	补充材料和练习	86
1.2 硬件	3		
1.3 软件	9	第6章 计算机的历史	90
1.4 因特网和万维网	11	6.1 第0代: 机械计算机(1642—1945)	90
1.5 展望未来	15	6.2 第1代: 电子管(1945—1954)	94
本章小结	15	6.3 第2代: 晶体管(1954—1963)	97
复习题	16	6.4 第3代: 集成电路(1963—1973)	98
		6.5 第4代: VLSL(1973—1985)	101
第2章 HTML 和网页	17	6.6 第5代: 并行处理和网络	
2.1 HTML 基础知识	17	(1985—???)	103
2.2 文档格式	19	6.7 展望未来	103
2.3 超文本和多媒体	24	本章小结	104
2.4 列表	27	复习题	105
2.5 表格	30		
2.6 使网页可访问	33	第7章 抽象和自定义函数	107
2.7 展望未来	34	7.1 自定义函数	107
本章小结	34	7.2 无返回语句的函数	111
补充材料和练习	35	7.3 多个输入	112
		7.4 参数和局部变量	113
第3章 因特网和万维网	38	7.5 random.js 程序库	116
3.1 因特网的历史	38	7.6 展望未来	119
3.2 因特网的工作原理	42	本章小结	119
3.3 万维网的历史	46	补充材料和练习	120
3.4 万维网的工作原理	49		
3.5 展望未来	51	第8章 算法和编程语言	125
本章小结	51	8.1 算法	125
复习题	52	8.2 设计和分析算法	127
		8.3 算法实例: 搜索链表	130
第4章 JavaScript 和动态网页	54	8.4 算法实例: 近似平方根	133
4.1 动态网页	54	8.5 算法和编程	134
4.2 变量和存储单元	59	8.6 展望未来	139
4.3 交互式网页	62	本章小结	140
4.4 展望未来	66	复习题	140
本章小结	67		
补充材料和练习	67	第9章 事件驱动程序设计	142
		9.1 通过按钮发起动作	142
第5章 JavaScript 数据和表达式	72	9.2 使用文本框的输入/输出	145
5.1 数据类型和操作符	72	9.3 通过多行文本区域输入/输出	151
5.2 数值提示	77	9.4 动态图像	153
5.3 JavaScript 预定义函数	80	9.5 展望未来	155
5.4 编程错误和调试	83	本章小结	155
5.5 展望未来	85	补充材料和练习	156

第 10 章 作为一门学科的计算机科学	161	第 15 章 JavaScript 字符串	249
10.1 计算机“科学”	161	15.1 字符串对象	249
10.2 计算机科学的主题	163	15.2 通用的字符串操作方法	252
10.3 计算机科学的分支	166	15.3 查找字符串	255
10.4 计算机科学道德标准	173	15.4 字符串操作和循环	257
10.5 展望未来	175	15.5 展望未来	259
本章小结	175	本章小结	259
复习题	176	补充材料和练习	260
第 11 章 条件执行	178	第 16 章 计算机内部——晶体管和集成电路	264
11.1 if 语句	178	16.1 电和开关	264
11.2 嵌套的 if 语句	183	16.2 从晶体管到门电路	268
11.3 例子: 掷骰子	185	16.3 从门电路到复杂电路	271
11.4 计数器	188	16.4 从电路到微型芯片	274
11.5 展望未来	190	16.5 展望未来	278
本章小结	191	本章小结	278
补充材料和练习	192	复习题	279
第 12 章 数据的表示	196	第 17 章 JavaScript 数组	281
12.1 模拟与数字	196	17.1 数组对象	281
12.2 二进制数	198	17.2 从字符串到数组	284
12.3 表示整数	199	17.3 数字数组	288
12.4 表示实数	200	17.4 例子: 用数组保存掷骰子的 统计数据	291
12.5 表示字符和字符串	202	17.5 展望未来	294
12.6 表示其他类型的数据	203	本章小结	294
12.7 计算机怎样区别不同的数据类型	207	补充材料和练习	295
12.8 展望未来	208	第 18 章 计算机和社会	302
本章小结	208	18.1 技术的正面影响	302
复习题	209	18.2 技术的潜在危险	309
第 13 章 条件重复	211	18.3 展望未来	313
13.1 while 循环	211	本章小结	313
13.2 计数器驱动的循环	217	复习题	314
13.3 变量和重复	220	附录 A 常用 Web 浏览器	316
13.4 例子: 产生一个 hailstone 序列	224	A.1 Microsoft Internet Explorer	316
13.5 展望未来	225	附录 B 常用文本编辑器	320
本章小结	225	B.1 Windows 的 NotePad	320
补充材料和练习	226	附录 C HTML 参考	323
第 14 章 计算机内部——冯·诺依曼 体系结构	230	附录 D JavaScript 参考	326
14.1 CPU 子单元和数据通道	230	附录 E random.js 函数库	332
14.2 CPU 和主存储器	233		
14.3 存储程序的计算机	237		
14.4 机器语言和汇编语言	244		
14.5 展望未来	246		
本章小结	246		
复习题	247		

第 1 章 计算机基础

本章将通过叙述、图解及复习题来介绍计算机科学和技术中的概念。

计算机相当快捷、准确却又笨拙；相对而言，人类相当迟缓、不准确却又才华横溢。两者的结合使它们的功能强大得超乎想像。

阿尔伯特·爱因斯坦

计算机之于计算，犹如乐器之于音乐。软件如同乐谱，它的演奏延伸了我们的空间，让我们的灵魂得以升华。达·芬奇曾将音乐称为不可见之物的神形。他精辟的论述同样适用于对软件的描述，甚至更胜于前者。

Alan Kay

《科学美国人》，1984 年

要想区分软件和硬件，最简单的办法是用脚去踢它。如果它碰伤了你的脚趾头，就表明它是硬件。

无名氏

计算机是一种较新的技术进步，至少就我们目前看待计算机的方式而言。第一批电子计算机是在 20 世纪 40 年代制造的，但供老百姓使用的台式计算机却是在 20 世纪 70 年代后期出现的。在短短的时间内，计算机迅速普及，从商业交易到个人通信，整个社会的方方面面都离不开计算机。然而，当许多人每天使用计算机工作和娱乐时，却很少有人了解计算机的工作原理。

本章将给出计算机技术的总体概念，着重介绍计算机的概念，计算机的组织结构及计算机的功能。这里不打算详细阐述每个主题——后续章节将再次讨论其中许多主题。但是，这个总体概念提供一个理解这些概念的框架，帮助你理解阅读此书过程中遇到的相关的计算概念，同时帮助你游刃有余地运用计算机技术。

1.1 什么是计算机

想到“计算机”这个词，脑海中浮现的画面很可能是一台放在办公桌上的机器，用于建立文档、发送电子邮件或上网。这种计算机通常称为**个人计算机(personal computer, PC)**，因为它被设计为一段时间内由一人使用；同时也称为**台式计算机(desktop computer)**，因为它体积小，可以放在桌面上。所以，一提到计算机，许多人仅仅想到台式机。然而，目前为满足不同的需要，有许多各式各样的计算机投入使用。一些计算机，我们称之为**超级计算机(super computer)**，它们的体积大得可以占据整个房间，每秒能执行数万亿次的计算。

超级计算机的主要使用者是涉及复杂任务(比如天气模式的建模或测试工程设计)的科研人员。较小较便宜的计算机包括能装进公文包的膝上计算机和能握于手中的掌上计算机(也称为个人数字助理或 PDA)。人们使用这些类型的计算机来完成文字处理和收发电子邮件等类似的工作。另外,大量的微型计算机被嵌入日常使用的设备中,并被用于控制这些设备,例如汽车、空调和微波炉。图 1.1 列出了 4 种不同类型的计算机,它们当中,有大得可占据一个房间的超级计算机,也有小得可握在手中的 PDA。

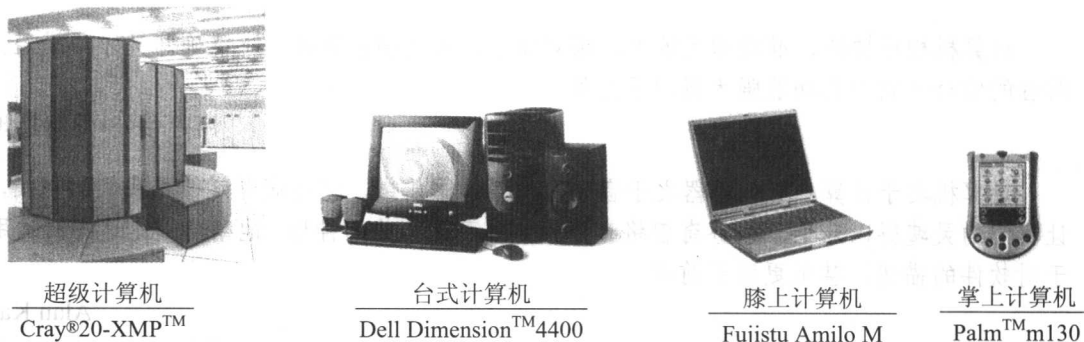


图 1.1 各种类型的计算机

一般说来,我们将把计算机定义为一种接收、存储和处理信息的设备。前面描述的计算机也许外观不同,而且功能也不同,但它们都符合这个一般化的定义。举个例子,一个超级计算机能接收与当前天气模式有关的数据,存储它们,并进行处理用以预测什么时候会发生暴风雨。一台个人计算机能识别使用者键入的字符,将它们存储于文件中,并处理它们以生成文档、电子数据表或电子邮件信息。同样,嵌入汽车防抱死制动系统(ABS)的计算机能感知到车轮的旋转,并能处理那些信息以计算出最佳的刹车模式,使车安全地停下来。

虽然计算机在日常生活中很普及,但极少有人真正理解它们的构造以及它们执行任务的方式。甚至购买一台个人计算机也极具挑战性,因为这涉及到技术性强的专业知识和晦涩难懂的计算机术语。例如,图 1.2 比较了一家大型计算机制造商在其广告中刊出的两种台式计算机系统的配置(2003 年 6 月)。系统 1 是低端计算机系统,它价格低但只提供有限的特性且使用较老的技术。系统 2 是高端计算机,通过融合最新技术来最大限度地发挥其性能,但成本较高。要理解如何取舍这两款机型,需要了解大量计算机知识——而且这还只是一家公司新推出的一种计算机系统可选项表格!

本章后面的内容将为图 1.2 中的关键术语给出定义,同时还将描述计算机的基本工作方式。本章还提供了“技术细节”将那些计算机概念同图 1.2 联系起来,对多数计算机概念进行补充说明。因此,在完成本章的学习后,你就能感受到这两个系统之间的区别。

对于图 1.2 的表格中的计算机——也可扩展到一般的计算机,理解它们的第一步就是区分开硬件和软件。硬件是指计算机的物理组件,例如监视器、键盘和硬盘驱动器。软件是指在计算机上执行(即运行)的程序,它们完成诸如文字处理或访问网络的任务。图 1.2 中,添加下划线的前 3 部分(标以 CPU、存储器和输入/输出)指系统的各种硬件组件。最后一排指软件组件。

	台式系统 1	台式系统 2
CPU	2.2 GHz 英特尔赛扬处理器	3.2 GHz 英特尔奔腾 4 处理器
存储器		
高速缓冲存储器	256 K 高速缓存	512 K 高速缓存
RAM	256 MB RAM	1 GB RAM
硬盘驱动器	40 G 硬盘驱动器	500 GB 硬盘驱动器
软盘驱动器	3.5"1.44 MB 磁盘驱动器	3.5"1.44MB 磁盘驱动器
CD-ROM/DVD	48x/24x/48x CD-RW/DVD 驱动器	48x/24x/48x CD-RW/DVD-RW 驱动器
输入/输出		
键盘	多功能键盘	多功能键盘
定点设备	光学滚轮鼠标	光学滚轮鼠标
显示屏	17"彩色显示器(15.9"可视)	18"液晶显示器
扬声器	扬声器	5 件套音响系统
调制解调器	56 K 调制解调器	56 K 调制解调器
网络适配器	10/100 以太网络适配器	10/100/1000 以太网络适配器
软件		
操作系统	Microsoft Windows XP 家庭版	Microsoft Windows XP 家庭版
应用程序	Internet Explorer Microsoft Works	Internet Explorer Microsoft Office XP Small Business Quicken® New User Edition Norton Anti-Virus™

图 1.2 两种不同的台式计算机系统(根据 2003 年 6 月的真实广告)

1.2 硬 件

鉴于当今计算机的多样性,我们难以描述出一个“典型的”计算机硬件系统。不过,某些特征是大多数台式机器所共有的,如图 1.3 所示。

虽然各计算机的特定组件和外观各不相同,但实际上当今所有计算机的基本结构都是相同的。这个结构就是著名的冯·诺依曼体系结构,以计算机科学先驱 John von Neumann(约翰·冯·诺依曼)的名字命名。在 20 世纪 50 年代早期,他就将自己的计算机组织方法进行了形式化定义。冯·诺依曼体系结构把一个计算机(即接收、存储和处理信息的设备)看作协同工作的三个基本部分。输入/输出设备允许用户通过键入命令然后看到结果的方式与计算机进行交互。存储器存储计算机要处理的信息以及程序,或者是用于指定完成特定任务所需步骤的指令。最后,中央处理器(CPU)执行程式化步骤以处理数据。这 3 部分通过总线连接起来,总线是以电信号形式传递信息的一组电线(阅读第 14 章,以了解更多关于计算机基础硬件的详细信息)。

采用冯·诺依曼体系结构的计算机称为存储程序的计算机,这意味着这些计算机可以从存储器中取出不同的程序,从而开始执行不同的任务。例如,为了完成诸如建立一个文本文档或进行一次计算这样的操作,CPU 要从存储器中读取相关联的程序指令并依次执行它们。如果该计算机用户想执行一个新任务,他只需要简单地将新程序载入存储器,然后指示 CPU 去执行这个程序。事实上,冯·诺依曼体系结构能使多个程序同时驻留在存储器中,并允许 CPU 调整这些任务,比如在 Web 浏览器和文本编辑器间来回切换。