



计 算 机 科 学 丛 书

伟大的计算原理

[美] 彼得 J. 丹宁 (Peter J. Denning) 著
克雷格 H. 马特尔 (Craig H. Martell)
罗英伟 高良才 张伟 熊瑞勤 译

Great Principles of Computing

GREAT PRINCIPLES
OF COMPUTING

PETER J. DENNING
CRAIG H. MARTELL



机械工业出版社
China Machine Press

机 科 学 丛 书

伟大的计算原理

[美] 彼得 J. 丹宁 (Peter J. Denning) 著
克雷格 H. 马特尔 (Craig H. Martell)
罗英伟 高良才 张伟 熊瑞勤 译

Great Principles of Computing

GREAT PRINCIPLES
OF COMPUTING

PETER J. DENNING
CRAIG H. MARTELL



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

伟大的计算原理 / (美) 彼得 J. 丹宁 (Peter J. Denning), (美) 克雷格 H. 马特尔 (Craig H. Martell) 著; 罗英伟等译. —北京: 机械工业出版社, 2017.4
(计算机科学丛书)

书名原文: Great Principles of Computing

ISBN 978-7-111-56726-4

I. 伟… II. ①彼… ②克… ③罗… III. 计算机科学 IV. TP3

中国版本图书馆 CIP 数据核字 (2017) 第 076369 号

本书版权登记号: 图字: 01-2016-3480

Peter J. Denning, Craig H. Martell: Great Principles of Computing (ISBN 978-0-262-52712-5).

Original English language edition copyright © 2015 by Massachusetts Institute of Technology.

Simplified Chinese Translation Copyright © 2017 by China Machine Press.

Simplified Chinese translation rights arranged with MIT Press through Bardon-Chinese Media Agency.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system, without permission, in writing, from the publisher.

All rights reserved.

本书中文简体字版由 MIT Press 通过 Bardon-Chinese Media Agency 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书系统总结了从算法到系统横跨整个计算机领域的 6 类计算原理 (计算、通信、协作、记忆、评估和设计), 旨在构建起一个框架帮助读者认识计算思维, 领会其核心思想——计算原理的相互影响以及问题有效解决的思维方式, 并将计算思维运用到计算机科学以外的其他领域。

本书适合作为高等学校非计算机专业计算思维课程以及计算机专业计算机科学导论课程的教学参考书, 也适合 IT 领域的程序员及专业人员阅读。

出版发行: 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: 和 静

责任校对: 殷 虹

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2017 年 5 月第 1 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 16.25

书 号: ISBN 978-7-111-56726-4

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，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；也正是这样的优势，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，还揭示了学术的源变，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和发展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到积极的推动作用，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公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系；从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 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、研究及珍藏。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相助，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，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，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。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所

采用。

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。随着计算机科学与技术专业学科建设的不断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将步入一个新的阶段，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。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：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

计算机及其相关技术正在广泛而深刻地改变着我们的生活。没有哪门科学像计算机科学这样，能够与其他科学领域产生出如此多的交叉。尤其是近年来云计算、大数据、移动应用、机器学习、人工智能等，更是引起了人们的广泛关注。计算机是如此的神奇，但是人们却很少能够真正了解计算机是如何运行并完成复杂的工作的。人们对于计算机的理解似乎总是隔着一层面纱，有时清晰，有时却又很模糊，甚至很多计算机专业人员也同样会有这种感觉。

本书并不是单纯地介绍计算机技术，而是重新审视和理解计算本质，或者说阐述“计算之道”。本书从一个不同的视角，把计算看作是一门遵从一组基本原理的科学，所有计算机相关的技术都可以从这组基本原理中找到依托。具体而言，本书提供了一种重要原理框架（great principle framework）来刻画计算科学中最具基础性的基本原理。它将计算原理分为6类：通信、计算、协作、记忆（存储）、评估和设计，而在阐述时又将其所覆盖的范围分成11个更容易管理的模块，其内容涵盖了计算的方方面面——包括计算机是如何工作的、如何选择算法、计算系统是如何组织的，以及如何进行正确而可靠的设计等。

正如作者所说，这本书是为所有想利用计算科学及相关技术来实现特定目标的人而设计的。在介绍关于计算的基本原理时，作者深思熟虑地综合描述了计算背后的基本概念，并试图以一种“任何对计算有一定了解的人都能理解”的方式呈现。因此，与其他深入剖析特定技术的书籍不同，本书努力为读者构建一个关于计算的全面视图，而更深入的知识则需要（或者值得）读者自己不断去发掘、吸收，并将它们连贯起来。

本书的作者之一 Peter J. Denning 教授曾是 ACM 的主席，是若干基本计算原理的发现者，也一直非常关注计算科学的教育，并且早在 20 世纪 90 年代就开始本书内容的研究、整理以及教育实践。本书的作者发现，随着核心技术的增加，对于初学者来说，原理框架比技术框架更容易理解。本书的原理框架并不是重新定义计算的核心知识，但它确实提供了一种看待该领域并降低其表面复杂性的新方式。从这个意义上来说，基于原理的“计算”教学，可以帮助初学者抑或专业人员更加容易地构建一个关于计算的全面视图，是一种非常值得去实践的教育理念。另一方面，本书关于基本计算原理的讨论也一定会启发更多的人对“计算之道”进行广泛而深入的思考。

本书由北京大学计算概论课程组的几位老师共同翻译，具体分工如下：前言，第 8、9 章及相关内容由罗英伟、张彬彬翻译，并由罗英伟负责全书翻译工作的组织和协调；序，第 3、11、12 章及相关内容由高良才翻译；第 1、2、5、10 章及相关内容由张伟翻译；第 4、6、7 章及相关内容由熊瑞勤翻译。译稿之中倘若有不当之处，敬请读者批评指正。

译者

2017 年 3 月于北京大学

Peter J. Denning 和 Craig H. Martell 提出并研究了一个具有里程碑意义的话题：识别并阐明一些原理，使计算机做“我们希望的”并努力减少其与计算机“实际做的”（即我们的命令）之间的差异。bug（程序错误）就是这种差异的典型例子。bug 通常是由于编程中疏忽大意导致机器做了“我们所不希望的”而出现的一种结果。但是程序错误并不是 bug 的唯一来源和表现形式，程序执行过程中的意外突发行为也会导致 bug 的产生，具有五花八门的软件和交互界面的计算机网络经常是这种意外突发行为的来源。我们有时候讲“网络效应”，其中，某些未曾设想过的特征会突然出现，并有可能强化为有力的趋势。在某个应用程序以一种难以预料的方式投入使用时，这种情况很可能发生，垃圾邮件和钓鱼邮件就是大规模网络中邮件应用的一个意外突发行为。

这样的效应使得理解、预测和分析大规模网络中大型软件系统所出现的复杂行为变得非常困难。即使系统中的每一个部件都按照其设计参数运行，整个系统仍可能因为组件间不可预测的交互而产生不确定的结果。

复杂突发行为的出现也可能仅仅因为计算机是有限的。数字化信息在表示上常常包括细小的错误或误差，然而小错误经过数十亿步的计算后很有可能积聚成大灾难。一个非常具体的例子就是有限精度的浮点数运算，正如 2005 年数值计算会议上 William Kahan 在他的论文中展示的那样¹，舍入误差以及对非常大或非常小的值的人工处理都可能导致灾难性的后果。

这些效应教会我们，让计算机代表我们来做事是一个不平凡的、高度智能化的尝试。因此，本书试图洞察一些基本计算原理，使得我们的方法和尝试能在最普遍意义上适应计算。

本书内容组织为 11 章，每一章的内容在与计算相关的活动中均起着重要作用。我认为这些内容在辅助“计算”上是调度和管理资源的焦点。计算即利用计算机和软件达到特定目标，这是一个故意模糊化的说法。在计算目标上，使一个电脑游戏工作和使一个分布式网络化的复杂金融交换系统工作是差不多的。而且，即使计算目标不同，在管理调度资源——信息表示、通信、计算单元、程序、内存、建模和分析等方面，一些具体的计算原理仍然可以帮助到设计师。本书的总体意图就是提出促使这些计算目标实现的基本原理，这一努力在广度和深度上均十分突出。

让计算如此有趣的一个因素是二进制表示的普适性。我们可以让比特位按照自己的想法代表任何事物，亦能以多种方式操作这些比特并从多种角度解释操作结果。就像把代数应用题转换为方程，根据直接的数学规则来获取初始方程的解一样，我们编写程序，并按照可以产生既定解释结果位的规则来操作比特位。大规模模拟、大数据以及复杂视觉渲染都具有这种属性，这些工作均可用来辅助我们理解并解释所操作的比特位。

我一直是“在高中、初中（甚至可能更早）阶段教授编程”的坚定支持者，原因之一在于编程对问题求解思维的系统训练。在编程这一活动中，学生需要分析问题，将问题分解为可管理的模块，想出程序需要怎样做才能解决问题（产生想要的结果），然后完成编程的整个过程：编写程序，利用已有的库，如果可能的话编译并运行程序，验证程序是否产生了想要的结果。最后一种训练（我们可称之为调试与验证的结合）是一种不仅仅适用于编程的技能。尽管我并不提倡每一个人都成为程序员，但学会良好编程中所用的技能是非常有价值的，因为这些技能对其他许多问题也广泛适用。

编程技能可以在处理复杂系统设计和分析时起作用，在这里我们接触到了 Denning 和 Martell 在“设计”这一章中强调的非常重要的一个领域。好的设计有许多有用的性质，这让我想起了一句名言——“整洁自有回报”，因为你可以在需要时找到整理好的东西。好的设计也自有回报，因为它有助于理解复杂性，并且具有通过演化、修正设计以达到新目标的能力。在互联网设计中，我们从先驱 ARPANET 那里汲取了教训，因为它无法在规模上扩展。然后我们设想了系统功能的分层并规范化层级间的接口，结果是在保持这些接口稳定的情况下，容许不同层在接口间实现与再实现的巨大灵活性。IP 协议是一个很好的例子，应用程序的设计者对于 IP 数据包如何运输（协议并未明确）一无所知，协议本身也不依赖于数据包载荷运输的是什么——载荷中比特位的含义是不清楚的。这种设计的一个结果是自 20 世纪 70 年代开始 IP 协议位于每一种新设计的通信系统的层级之上；另一个结果是新的应用程序不用改变网络就可放在互联网上，因为 IP 协议可将数据包传输给互联网上的软件。只有发送和接收的主机需要知道数据包运输的载荷比特位是什么意思，传输数据包的路由器也无需依赖数据包载荷的内容。

设计之于计算的作用怎样强调都不过分。不管是硬件、操作系统、应用程序、数据、文件目录结构或者语言的选择，归结到底都是思考设计以及设计的组合——系统怎样工作，虽然有时人们很少听说术语系统工程。我是一名系统设计师，乐于思考架构相关的问题：所有的部分如何组装到一起？每一部分应该是什么？设计如何适应新的需求？设计可维护吗？教会其他人如何设计困难吗？

对于好的设计，一个有趣的测试是看新接触某系统的人在不破坏之前设计功能的基础上能否让系统做一些新的事情。在某种程度上，这是对一个人理解程序或系统及其架构的一个十分有力的测试。你可能不需要了解系统的所有细节，但你需要了解足够多以确保你的改动不会产生预料之外的后果，这就是干净设计的含义，即设计能够合理地且安全地进行改动。我很高兴本书非常重视设计，并强调架构（不仅仅强调算法）对于设计的作用。

关于计算的基本原理还有许多可以说的，但那正是本书接下来的要点。记住这些原理将使计算系统的设计更加可控，继续阅读吧！

Vint Cerf

Woodhurst, VA

2014 年 4 月

就在 70 年前，除了少数专家之外，没有人听说过计算机。现在，计算机、软件和网络无处不在。在地球上的任何地方，它们都以更快的发展速度给我们的生活带来了各种各样的好处。

在这么短的几十年中，我们学会了设计和建造如此规模的系统，这真是一件令人吃惊的事。如今，通过支持大规模合作，计算技术使得知识工作能够自动化，同时也在不断扩大生产力。第二次机器革命正扑面而来¹。这是如何实现的？是什么样的伟大思想使这一切成为可能？

计算机给我们带来好处的同时也带来忧虑。计算机带来的自动化是否会使很多工人失业？计算机是否会成为终极监督工具而使我们失去隐私？计算机是否会发展出超越人类的智能？计算机能做的事情会有限制吗？

我们相信，理解计算的原理和法则可以帮助人们理解计算是如何完成如此之多的工作的，并消除他们的忧虑。为此我们写了这本书，书中介绍了关于计算的一些最重要的原理，并以任何对计算有一定了解的人都能理解的方式呈现。

计算机科学（computer science）不只是设计计算设备的工程领域，它是一门关于信息处理的科学。计算受科学原理和法则支配，这些原理和法则告诉我们计算机能做什么、不能做什么。信息的法则揭示了根据物理法则无法直接得出的新的可能性和限制。专家们赋予计算机许多计算科学（computing science）告诉我们不可能拥有的能力，同时，这些专家又低估了计算机真正的能力。

计算机科学与很多其他领域相互交叉。许多科学与工程领域都有计算（computational）分支，如计算物理、计算化学、生物信息学、数字化产品设计与制造、计算社交网络、计算心脏病学等²。各层次的教育者正努力在他们拥挤的课程表中加入计算相关的课程，以保证课程体系的先进性。但仍有很多中学由于缺少计算机科学方面的教师而不能开设计算机课程。在商业领域，诸如“大数据”“云计算”“网络安全”等热门词汇也散发出共同的信号，期望“计算原理”在数据管理、分布式计算、信息保护中发挥作用。

一直以来，人们把计算看作一个按照摩尔定律高速发展的技术领域³。而我们的观点有所不同，我们相信计算更应该被描述为一个科学领域，具有跨越所有计算技术以及人工

或自然的信息处理的基本原理。我们需要一种新的方法来刻画计算。就像望远镜之于天文学、显微镜之于生物学，计算机是计算的工具，而非计算的研究对象。

本书的重要原理框架（great principles framework）就是这样一种新的方法。它将计算原理分为 6 个类别：通信、计算、协作、记忆（存储）、评估和设计。计算原理是指用来指导或约束我们如何操纵物质和能量来进行计算的声明。计算原理可以是：（1）重现，包括描述可重复的因果关系的定律、过程及方法；（2）行为准则。局部性原理（locality principle）就是重现的一个例子：每一个计算在一定的时间间隔内，其对数据的访问都聚集在一个小的子集里。行为准则的一个例子就是网络程序员将协议软件划分为多个层次。所有这些原理的目的，都是希望通过增进理解和降低复杂性从而得到良好的设计。

每种计算技术都利用了这些类别的原理。这个框架是广泛和全面的，覆盖了计算的每个部分，包括算法、系统和设计。

从事计算工作的人员形成了许多计算领域（computing domain）——实践社区，如人工智能、网络安全、云计算、大数据、图形学以及科学计算（computational science）等。这些领域都专注于推进领域向前发展并与其他社区互动，它们既从计算原理中获益，又受其约束。没有这些计算领域的原理框架是不完整的。

由于这 6 个类别过于庞大，我们决定将其所覆盖的范围分成 11 个更容易管理的模块，就像你在目录中看到的那样。关于这一点，我们将在第 1 章中详细说明。

从机器到通用的数字化

计算的机器是早期计算领域的关注中心（从 20 世纪 40 年代到 20 世纪 60 年代）。计算被看作机器执行复杂演算、解方程、破译密码、分析数据及管理业务流程的行为。那时的先驱们将计算机科学定义为研究以计算机为中心的各种现象。

然而，这些年来，这一定义变得越来越没有意义。20 世纪 80 年代的科学计算运动认为，计算是除了传统的理论和实验之外的一种新的做科学研究的方法。他们使用“计算思维”（computational thinking）这个术语作为研究和问题求解的思维训练，而不是作为建造计算机的方法。十年之后，一些领域的科学家开始发现各自领域内的自然信息处理，其中包括生物学（DNA 翻译）、物理学（量子信息⁴）、认知科学（脑力过程）、视觉（图像识别）和经济学（信息流）。计算的重点从机器转变到信息处理，包括人工信息和自然信息。

现在，随着几乎所有事物的数字化，计算进入了人们的日常生活，包括求解问题的新方法，艺术、音乐、电影的新形式，社交网络，云计算，电子商务，以及新的学习方法

等。用计算作比喻成为日常语言中的必要组成部分，比如“我的软件有反应了”或“我的大脑崩溃了需要重启”这样的表达方式。

为了应对这些变化，各大学一直都在设计新的基于原理的方法来开展关于“计算”的教学。华盛顿大学是这方面的先驱之一，它开发了关于熟练掌握信息技术的一门课程和教材，目前正在高中和大学中广泛使用，以帮助学生学习和应用基本计算原理⁵。教育考试基金会（Educational Testing Foundation）与美国国家自然科学基金会（National Science Foundation）合作，开发了一门新的基于计算原理的先修课程⁶。现在很多人使用“计算思维”这个词，指的是在很多领域和日常生活中使用计算原理，而不仅局限于科学计算⁷。

随着计算领域的日趋成熟，它吸引了其他领域的众多追随者。我们知道有 16 本书是为感兴趣的非专业人员来解释计算的各个方面⁸。大部分书关注的只是单个部分的内容，如信息、编程、算法、自动化、隐私以及互联网原理等。本书则将这个领域作为一个整体来看待，给出所有各部分如何组合在一起的系统叙述。读者会发现在所有这些部分的背后是一套连贯的原理。

根据教授从其他专业转到计算机科学的研究生的经验，我们发现对于初学者来说，原理框架比技术框架更容易理解。当早期核心技术很少的时候，用技术思想的观点来描述该领域是一种好方法。1989 年，美国计算机协会（Association for Computing Machinery, ACM）列出了 9 大核心技术。而在 2005 年，ACM 列出了大约 14 种，到了 2013 年，则有约 18 种。本书的 6 类原理框架并不是重新定义计算的核心知识，但它确实提供了一种看待该领域并降低其表面复杂性的新方式。

起源和目标

我们经常被问及 6 类原理的起源。20 世纪 90 年代，本书作者之一 Peter J. Denning 在乔治梅森大学（George Mason University）开始这个项目。他从众多的同事那里收集了一个可能的原理陈述的列表。他发现了 7 个自然的群集，并将它们称为通信、计算、记忆（存储）、协作、评估、设计和自动化⁹。当组织本书的时候，我们意识到自动化并不是一个操纵物质和能量的类别，而是人工智能计算领域的重点。在本书中，我们从类别集合中删除了自动化，并将其包含在计算领域中。

这 6 个类别并不是把计算的知识空间划分成分离的片段。它们就像六角亭的窗户，每一扇窗户都以独特的方式呈现出内部空间，但同一件事物可以从多个窗户看到。例如，互联网有时以数据通信的方式、有时以协作的方式、有时以记忆（存储）的方式被看到。

这组类别有一个类别数目可控的框架，从而满足了我们的目标。虽然计算技术的列表还将继续增长，计算领域的集合也会扩大，但是类别的数目在较长时间内应该会保持稳定。

这本书是关于计算机科学的一个整体视角，注重最深入、最广泛的原理，即“宇宙普适的”原理（cosmic principle）¹⁰。本书将计算视作一种深层次的科学领域，其原理将影响包括商业和工业在内的其他每一个领域。

这本书是为所有想利用计算科学来达到其目标的人而设计的。受过科学教育的读者可以学到从算法到系统横跨整个领域的计算原理。而计算领域内的人，例如一个想要学习并行计算的程序员，可以找到这个巨大领域内不太熟悉的部分的概述。对于大学里学习诸如“计算机科学基础”课程的学生，本书可以帮助他们理解计算技术是如何影响他们的，例如网络和互联网如何使社交网络成为可能。初出茅庐的科学家、工程师和企业家可能在这本书中找到一个面向整个计算机科学的科普型方法。

致谢

Peter 要感谢他在计算原理的漫长旅程中遇到的许多人，这一旅程开始于他 11 岁时，那时他的父亲给了他一本关于机器原理的不同寻常的书——1911 年出版的《How It Works》¹¹。1960 年，他的高中数学老师和科学社团导师 Ralph Money，鼓励并引导他把精力投入计算机——引领未来的机器中去。1964 年，当他成为 MIT Mac 项目的学生时，他的导师 Jack Dennis、Robert Fano、Jerry Saltzer、Fernando Corbato 和 Allan Scherr 把他的兴趣扩展到所有计算背后的基本原理。1967 年他发表的第二篇论文是关于存储管理的工作集原理，其灵感主要来自于 Les Belady、Walter Kosinski、Brian Randell、Peter Neumann 和 Dick Karp 的帮助。1969 年，他带领一个工作小组设计操作系统原理的核心课程，他的队友 Jack Dennis、Butler Lampson、Nico Habermann、Dick Muntz 和 Dennis Tsichritzis 帮助确定了操作系统的原理，其中也包括 Bruce Arden、Bernie Galler、Saul Rosen 和 Sam Conte 的见解。在之后的几年里，Roger Needham 和 Maurice Wilkes 提供了关于操作系统原理的很多新的见解。1973 年，他与 Ed Coffman 合写了一本关于操作系统理论的书。

1975 年，Jeff Buzen 把他吸引到操作分析的新领域，这项研究关注计算机系统性能评估的基本原理。在那段时间里，Erol Gelenbe、Ken Sevcik、Dick Muntz、Leonard Kleinrock、Yon Bard、Martin Reiser 和 Mani Chandy 都促进了他对计算原理的理解。

1985 年，ACM 教育委员会（ACM Education Board）请他领导一个项目，以确定“计算”作为一门学科应具备的核心原理，用于设计 1991 年 ACM/IEEE 的课程推荐。他非常

感谢这个团队加深了他对计算原理的理解，这个团队的成员有：Douglas Comer、David Gries、Michael Mulder、Allen Tucker、Joe Turner 和 Paul Young。

20 世纪 90 年代中期，他开始在一个统一的框架下收集所有的计算原理。Jim Gray 嘱咐他去寻找“宇宙普适的”原理——那些有足够深度和广度的、在任何时间、在宇宙中任何部分都有效的原理。他设计了一门叫作“信息技术核心”的“旗舰”课程，并在乔治梅森大学的同事 Daniel Menascé、Mark Pullen、Bob Hazen 和 Jim Trefil 的帮助下于 1998 年启动了该课程。

2002 年，ACM 教育委员会邀请他成立一个重要原理工作组，以便就重要原理框架如何促进未来课程的设计给出建议。一个相当优秀的团队为此走到了一起，他们是：Robert Aiken、Gordon Bell、Fred Brooks、Fran Berman、Jeff Buzen、Boots Cassel、Vint Cerf、Fernando Corbato、Ed Feigenbaum、John Gorgone、Jim Gray、David Gries、David Harel、Juris Hartmanis、Lilian Israel、Anita Jones、Mitch Kapor、Alan Kay、Leonard Kleinrock、Richard LeBlanc、Peter Neumann、Paul Rosenbloom、Russ Schackelford、Mike Stonebraker、Andy Tanenbaum、Allen Tucker 和 Moshe Vardi。

关于计算中的科学问题，Rick Snodgrass（一个追求“Ergalics”[⊖]的志同道合的人）给了我们很多睿智的建议。Vint Cerf 和 Rob Beverly 就网络相关的章节也给出了很多有用的建议。

Peter 最感谢妻子 Dorothy Denning，感谢她对于清晰的逻辑流程和坚实基础的一贯坚持，感谢她始终如一地鼓励他抓紧时间写作。同时也要感谢女儿 Anne Denning Schultz 和 Diana Denning LaVolpe 对他的信任和支持。

Craig Martell 之所以被吸引来合作写这本书，是因为他总是想要参数化这个世界。在这方面，计算作为一个领域呈现出特别迷人的挑战，因为它是科学、数学和工程学的融合体。他时常痴迷于这些机器居然能工作！

Craig 要感谢 Mitch Marcus，他们在宾夕法尼亚大学共同讲授了“信息技术及其对社会的影响”这门课程。制作这门课程教学大纲的过程开启了他对本书的贡献。他还要感谢合作者 Peter Denning，是他让整个写作过程变得有趣且富有成效。最后，他要感谢 Pranav Anand、Mark Gondree、Joel Young、Rob Beverly 和 Mathias Kölsch，他们使工作毫不沉闷。

Craig 对于本书的贡献离不开妻子 Chaliya 的耐心与全方位的支持，还有女儿 Katie 崇敬的笑容。有了她们，他坚信自己是这世界上最幸运的男人。

⊖ Ergalics 是关于计算工具以及计算本身的科学，是一项由美国亚利桑那大学 Rick Snodgrass 提出并开展研究的项目，详情请见 <http://www.cs.arizona.edu/people/rts/ergalics/>。——译者注

万维网	184	一个崭新的机器时代来临	192
网络科学	187	我们的思维方式正在转变	193
致谢	188	设计的核心性	193
第 12 章 后记	189	各章概要	195
没有意识的机器	189	注释	200
智能机器	189	参考文献	213
架构和算法	191	索引	227
经验思维	192		

不可计算问题	92	电话交换机	148
总结	96	分时系统	149
第 7 章 存储	98	用模型来计算	150
存储系统	99	结论	152
存储器的基本使用模型	100	第 10 章 设计	154
命名	101	什么是设计	156
映射	105	软件系统的准则	158
虚拟存储	105	需求	158
共享	107	正确性	159
能力	108	容错性	159
认证	111	时效性	160
层级结构中的定位	112	适用性	160
为什么局部性是基础	116	设计原理、模式和示意	161
结论	117	原理	161
第 8 章 并行	119	模式	162
并行计算的早期方向	120	示意	163
并行系统的模型	123	软件系统的设计原理	163
协作的顺序进程	124	层级式聚合	164
功能系统	124	封装	165
事件驱动的系统	125	级别	166
MapReduce 系统	125	虚拟机	168
协作的顺序进程	125	对象	170
功能系统	131	客户端与服务器	171
结论	134	总结	172
第 9 章 排队	136	第 11 章 网络	173
排队论遇上计算机科学	137	弹性网络	174
用模型计算和预测	139	数据包交换	175
服务器、作业、网络和规则	140	互联网络协议	178
瓶颈	144	传输控制协议	179
平衡方程	146	客户端与服务器	180
ATM	147	域名系统	181
		网络软件的组织结构	183