HZ BOOKS
华章教育

计 算 机 科 学 丛 书

PEARSON

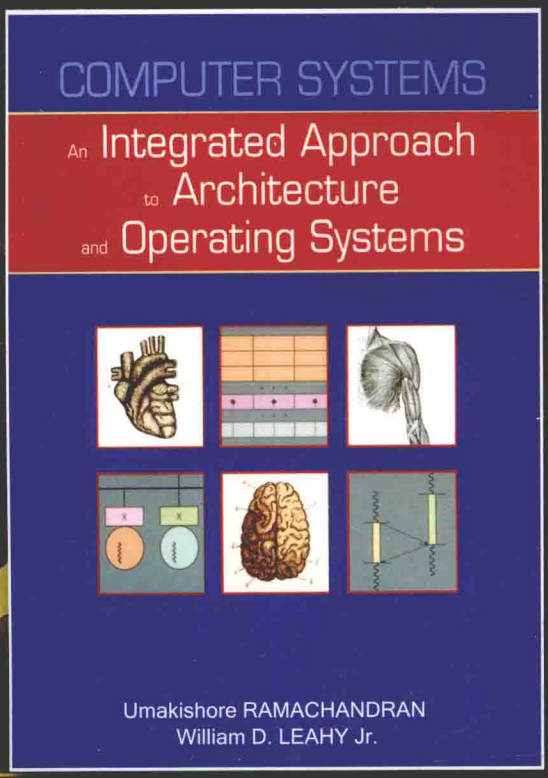
# 计算机系统

## 系统架构与操作系统的高度集成

[美] 阿麦肯尚尔·拉姆阿堪德兰（Umakishore Ramachandran） 小威廉 D. 莱希（William D. Leahy Jr.） 著
佐治亚理工学院
陈文光 等译
清华大学

# Computer Systems

## An Integrated Approach to Architecture and Operating Systems




机械工业出版社
China Machine Press

计 算 书

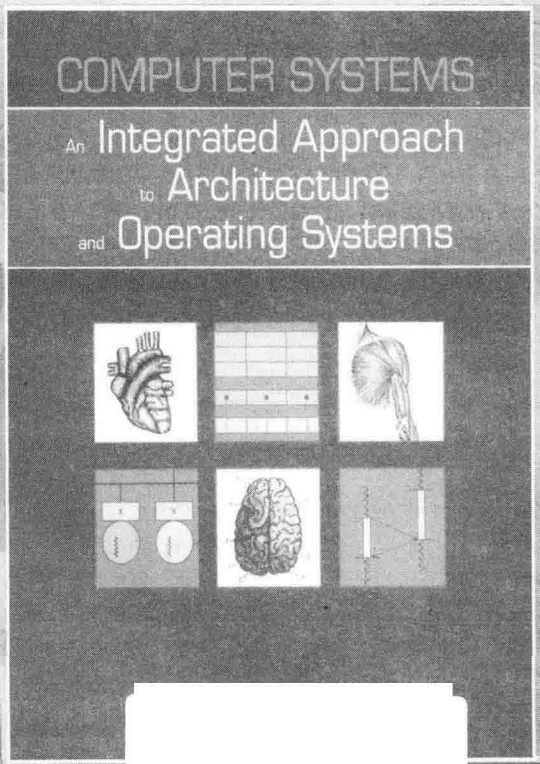
# 计算机系统

## 系统架构与操作系统的高度集成

[美] 阿麦肯尚尔·拉姆阿堪德兰（Umakishore Ramachandran） 小威廉 D. 莱希（William D. Leahy Jr.） 著
佐治亚理工学院
陈文光 等译
清华大学

Computer Systems

An Integrated Approach to Architecture and Operating Systems




机械工业出版社
China Machine Press

**图书在版编目（CIP）数据**

计算机系统：系统架构与操作系统的高度集成 /（美）拉姆阿堪德兰（Ramachandran, U.），（美）莱希（Leahy, W. D.）著；陈文光等译 . —北京：机械工业出版社，2015.7
（计算机科学丛书）
书名原文：Computer Systems: An Integrated Approach to Architecture and Operating Systems

ISBN 978-7-111-50636-2

Ⅰ. 计… Ⅱ. ①拉… ②莱… ③陈… Ⅲ. 计算机系统 Ⅳ. TP30

中国版本图书馆 CIP 数据核字（2015）第 128116 号

**本书版权登记号：图字：01-2010-5389**



本书采用集成方法，系统地讲解了计算机系统的软件和硬件知识。全书分为 5 个模块：处理器、内存系统、存储系统、并行系统和网络，分别讨论了处理器及其相关的软件问题、内存系统和分级存储体系、I/O 和文件系统、操作系统问题及支持并行编程的多处理器中相应体系结构特点、网络硬件的发展和处理各种网络行为的网络协议栈的特点等。

本书适合作为本科生“计算机系统”课程的教材，同时也适合相关专业研究人员阅读。

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：盛思源 关 敏 责任校对：董纪丽
印 刷：北京诚信伟业印刷有限公司 版 次：2015 年 7 月第 1 版第 1 次印刷
开 本：185mm × 260mm 1/16 印 张：34.25
书 号：ISBN 978-7-111-50636-2 定 价：99.00 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

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，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred 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，我们的联系方法如下：

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

# 译者序

Computer Systems: An Integrated Approach to Architecture and Operating Systems

美国未来学家阿尔温·托夫勒在1980年3月出版了他的经典著作《第三次浪潮》，该书在全世界引起了巨大的反响。在这本书中，他将人类社会发展分为农业阶段、工业阶段和以信息时代为主要特征的后工业化社会阶段。从历史视角来看，信息化具有与工业革命等同的重要性，将重塑我们的社会结构和日常生活。

计算机是信息化的核心，随着信息化与社会生活的深度融合，人们对计算机专业毕业生的要求也越来越高。他们不仅需要掌握计算机本身的知识，还需要了解相关行业的知识。这就给计算机专业教学带来了很大的挑战，如何在有限的课时里面，能够覆盖计算机专业的核心内容，提供给学生足够的基础使其能够在所选的方向上继续深造呢？

传统的计算机课程体系有一个重要问题，就是课程人为地割裂了解决问题时所需技能的综合性。例如汇编语言、计算机原理、计算机系统结构、操作系统和编译原理分别从不同角度介绍了计算机的硬件和软件系统，但人们在面临一个具体问题的时候，比如优化一个数据分析程序时需要的技能是综合性的，需要知道高级语言程序变成了什么样的汇编语言，这些汇编语言在操作系统的调度下如何加载和运行，运行时的指令如何在处理器的流水线里乱序执行，其访存是缓存命中还是缓存缺失，并发访问是如何相互隔离的，等等。

因此，国内外大学都在这方面展开了探索，即如何用一种综合的方法来介绍计算机系统的相关内容，这样不但可以减少课时，也能让相关知识的衔接更加平滑，整体的知识体系更加系统化。CMU的《深入理解计算机系统》是目前比较成功的探索，国内外很多大学都已采用该教材作为课程的基础。我们翻译的这本书则是另一个有益的尝试。本书中计算机系统结构和操作系统的内容基本平衡，而《深入理解计算机系统》则明显偏向操作系统，对计算机系统结构的相关内容介绍相对较少。例如，本书对I/O中断处理专门安排了硬件实验，要求学生用硬件设计语言设计CPU并支持中断处理，这类实验对学生理解整个计算机系统是如何运作的非常重要，但在《深入理解计算机系统》中没有这部分内容。

我们希望本书的翻译出版，能够为国内的计算机系统教育提供一种新的选择。对希望未来研究、设计新型计算机系统的学生来说，本书提供了更加完整的基础。

本书的翻译是由我和我的学生完成的，我本人翻译了前言和第1章，汤雄超翻译了第2～4章，杨弋翻译了第5～7章，张峰翻译了第8～10章，朱晓伟翻译了第11～12章，陈力维翻译了第13章和附录。本人对全书进行了审校，因此书中的错误都应该由本人负责。

感谢机械工业出版社华章公司将这本书引入国内，感谢温莉芳副总经理、朱劼编辑和关敏编辑在本书翻译过程中给予我们的极大耐心。

陈文光

2015年6月

## 为什么在计算机系统领域需要有一本新书

和高中生谈论计算机会让人感到兴奋。人们对“盒子（计算机机箱）里”有什么东西有一种神秘感，正是那个盒子里的东西使计算机能够完成诸如让用户玩有很棒图形的视频游戏、播放音乐（不管是 RAP 还是交响乐）、发送即时消息给用户的朋友等功能。本书的目的就是与读者一起开展一段揭示盒子里有什么秘密的旅程。作为即将开展的旅程的一瞥，让我们在一开始就表明，让这个盒子变得有趣的并不仅仅是硬件，还包括软件和硬件是如何结合起来完成各种功能的。因此，本书所采用的途径是把软件和硬件放在一起观察，看它们是如何相互帮助以及如何协同起来让计算机变得有趣而且有用的。我们把这个过程称作“打开盒子”——即揭开盒子里有什么这个秘密：我们查看盒子内部并理解如何设计关键的硬件单元（处理器、内存以及外设控制器），理解要管理计算机中的所有硬件资源，包括处理器、内存、I/O 和硬盘、多处理器以及网络所需的操作系统抽象。因此，这是一本**计算机系统**教学的入门课程教材，采用了一种新颖的**集成教学法**来介绍相关内容。

本书的目标是让学生在本科生涯（计算机科学或计算机工程专业）的早期就在相关主题方面接触足够宽泛的知识。本书的内容是为用软硬件集成的方式进行课程教学而写的，这种方式使得学生可以了解计算机体系结构和系统软件之间的关系。书中的材料可以作为 4 学分的半年学期课程教材，或者作为 5 学分的季度课程教材，或是作为每季度 3 学分的两季度的课程系列的教材。基于本书的课程可以为学生打下很好的基础，以进一步深入学习计算机体系结构、操作系统和网络的高级课程或研究生课程，在这些领域进一步深造。此外，这类课程可以尽早激发学生对计算机系统的兴趣，对学生在本科期间参加研究工作也有帮助。

本书的主要特点（除了处理器和内存系统之外）如下：

1）详细介绍了存储系统；

2）专门用一章介绍了网络问题；

3）专门用一章介绍了多线程和多进程编程。

## 教学风格

本书采用的教学风格是“发现”而非“教导”或“灌输”。此外，内容是以“自顶向下”的方式展现的，读者首先看到我们要解决的问题，然后看到解决方案。以内存管理部分（第 8 章）为例。我们首先提出问题“什么是内存管理”，一旦理解了内存管理的需求，我们再开始探讨内存管理所需的软件技术和相应的硬件支持。因此，本书是以一种讲故事的方式来进行概念展现，学生们看起来很喜欢这种方式。在适当的地方，我们在不同章节用一些例题来阐明观点。

我们在撰写本书的时候始终以学生为中心。书中包含大量例题，可以帮助学生固化刚刚讨论过的概念。从我们作为教师的经验来看，学生确实喜欢了解历史背景（那些对计算的演化起到重要影响的著名的计算机科学家和机构）和现状，以及我们是怎么一步一步发展过来

的。这些历史片段遍布在全书中。除此之外，在必要的时候，在若干章我们都包括了一节从历史角度进行的回顾。我们从学生那里学到并采用的另一个措施是在文中直接给出参考文献，而不是在文末才给出。读者可以看到贯穿本书的大量脚注。此外，我们在每章末尾专门有一节给出外部链接（教材和开创性的著作），包括参考文献和扩展阅读的建议，这些内容在正文中不一定都被引用了，但是有助于增强学生的知识基础。今天，随着因特网上的信息日益丰富，为附加的信息提供 URL 链接是一件很有诱惑力的事情。但是，我们拒绝了这一诱惑（除了那些权威信息源的可靠链接）。尽管如此，我们知道现在学生在去图书馆之前会先搜索因特网，当然他们也应该这么做。在这种情况下，我们给学生一个提示：在利用因特网作为信息源的时候要慎重。通常，使用 Google 搜索是获取某种信息的最快方法。但是，必须对这些信息进行筛选以保证其准确性。作为经验法则，使用因特网上的信息来满足好奇心或是回答与流言有关的问题。（DEC 是如何衰落的？为什么 Linux 成功了而 Unix BSD 却没有？ Burroughs 公司的历史是什么？计算机系统的真正先驱是哪些人？）对于技术问题（Pentium 4 处理器的流水线结构是什么？ VAX 11/780 的指令集体系结构是什么？）则要从已出版的书籍、相关会议和期刊论文（当然它们中的大多数也可以在线获取）中寻求答案。

佐治亚理工学院计算机学院从 1999 年秋季学期开始，每学期都开设这门软硬件集成的课程，本教材就是这门课程的副产品。在一开始，本书作者为课程开发了完整的讲义和幻灯片，并使用两本标准的教材（一本体系结构教材和一本操作系统教材）作为课程的背景参考资料来补充课程的材料。从 2005 年春季开始，我们将课件转换成了本教材的手稿，因为学生一直想要一本与课程内容和风格匹配的教材。本教材的在线版本从 2005 年春季开始在佐治亚理工学院用于本课程，使用集成的方法介绍计算机系统。本课程每年开设 3 次（包括夏季学期），每学期有 80 多名学生选课。因此，书稿在付印之前经过了连续 15 个学期的教学，从选修本课的学生那里接受了持续不断的反馈与改进意见。

在设计产生本书的课程时，以及在撰写本书的时候，我们从其他机构开设的系统入门课程以及一些优秀教材中学到了很多东西。例如，MIT [⊖]的计算机系统入门课程拥有很长的历史和传统，而且是真正独一无二的。从这门课程中总结的教材 [Saltzer，2009] 对激发学生深入学习计算机系统来说是极好的资源。在撰写本书的时候，我们坦承受到了 [Ward，1989] 和 [Kurose，2006] 的教学法的启发。

## 本书的结构和可能的阅读路径

本书的知识内容可以分为 5 个模块。下面的路线图建议了一些可能的阅读路径。这些路径假设关于体系结构和操作系统的内容一样多。

1）**处理器** 本书的第一个模块是关于处理器以及与处理器相关的软件问题的。我们从探索如何设计盒子中的大脑[⊖]（处理器）开始。有哪些软件问题？既然计算机的大部分部件主要是使用高级语言编程的，我们考虑了高级语言结构是如何影响处理器的指令集的（第 2 章）。一旦理解了指令集的设计，我们就开始关注实现处理的硬件技术。我们从实现一个简单的处理器开始（第 3 章），然后考虑实现一个使用流水线技术的性能优化的处理器（第 5 章）。处理器是计算机系统中的宝贵资源，因此必须在多个相互竞争的程序间复用，正如第 1 章中视频游戏的例子所揭示的一样（见 1.3 节）。操作系统的职责就是保证资源的有效使用。本模块以

⊖ http://mit.edu/6.033/www/。

⊖ 原书封面中的解剖图用来表示将计算类比为在人体内自然发生的网络分布式处理。

用于处理器调度的操作系统算法结束（第 6 章）。

我们预计第 2、3、5 和 6 章每章需要 3 小时的课堂讲授时间和 1 小时的练习题时间。

2）**内存系统** 第二个模块介绍了内存系统和内存层次。计算机程序包括代码和数据，并且都需要存放的空间。计算机的内存系统可能是决定性能最为关键的因素。如果内存系统不能以匹配处理器速度的方式提供执行程序所需的代码和数据，处理器速度（现在以 Ghz 为量度）就毫无意义。由于技术的进步，内存系统的大小一直在跨越式发展，但应用程序使用内存的胃口也在以同样的速度增长着，如果不是增长得更快的话。因此，内存也是宝贵资源，操作系统的作用就是保证用好资源。本模块的第一部分是关于有效管理内存的操作系统算法以及相应的体系结构支持的（第 7 章和第 8 章）；第二部分则介绍内存层次，可以帮助降低处理器在访问代码和数据时的延迟（第 9 章）。

我们预计第 7、8 和 9 章每章需要 3 小时的课堂讲授时间和 1 小时的练习题时间。

3）**存储系统** 第三个模块是关于 I/O（特别是稳定存储）和文件系统的。只有与计算机进行交互才能让计算机有用且有趣。首先，我们讨论能够把处理器的注意力从当前执行的程序中脱离出来的硬件机制（第 4 章）。这些机制既包括外部事件也包括处理器执行程序时遇到的内部异常。与硬件机制相关的软件问题是解决正常程序执行的“不连续”性，包括记录原有程序的执行位置以及程序的当前执行状态。然后，我们介绍处理器与 I/O 设备的接口机制以及相应的底层软件技术（第 10 章），并特别强调了磁盘子系统。随后，我们完整地介绍了在稳定的存储设备（如磁盘）上如何构建文件系统（第 11 章）。

我们预计第 4 章和第 10 章每章需要 3 小时的课堂讲授时间和 1 小时的练习题时间，第 11 章需要 6 小时的课堂讲授时间和 2 小时的练习题时间。

4）**并行系统** 计算机体系结构是一个快速变化的领域。芯片密度、处理器速度、内存容量等在过去 20 年中都呈现出指数增长速度，并在可预见的未来仍然保持这样的增长速度。并行处理已不再是超级计算机独有的深奥概念。随着在一个芯片上集成多个 CPU 的多核技术的到来，并行性已经变得很常见。因此，理解与并行性有关的软件和硬件技术对于回答“盒子里有什么”这样的问题十分必要。这个模块包括多处理器中支持并行编程的操作系统问题以及相应的体系结构功能（第 12 章）。

我们预计第 12 章需要 6 小时的课堂讲授时间和 2 小时的练习题时间。

5）**网络** 在我们生活的世界上，单独一个盒子几乎没有任何用处，除非它与外部世界相连。与你的朋友在网络上对战多人视频游戏（在第 1 章介绍）是一个很好的例子。但即使在日常生活中，我们也需要利用网络来收发电子邮件或浏览因特网等。网络与其他输入 / 输出设备的不同之处在于，现在你的盒子得以连接世界了！你需要一种语言让你的盒子与外部世界交谈，并处理网络的各种情况，例如暂时或永久的连接中断。这一模块讨论了网络硬件的进化，以及用来处理各种网络状况的网络协议栈（操作系统的一部分）的功能（第 13 章）。

我们预计第 13 章需要 6 小时的课堂讲授时间和 2 小时的练习题时间。

总而言之，第 2 章 ~ 第 10 章每章需要用 1 周时间授课；第 11 章 ~ 第 13 章每章需要 2 周时间授课，正好在 15 周的一个学期里讲完。五个模块中的软件和硬件问题在本书中是一起介绍的，上述建议的阅读路径也是按照这种处理方式进行的。

读者也可以选择在体系结构和操作系统主题之间重点关注某部分的内容，而不会损失连续性。以处理器模块为例，第 3 章和第 5 章都是关于处理器的硬件实现问题的。对于偏重操作系统的课程，可以考虑少讲授或者完全跳过介绍流水线处理器实现（从 5.7 节开始）的第 5

章，而不会损失课程的连续性。类似地，在偏重体系结构的课程里，可以跳过介绍处理器调度算法的第 6 章，而不会损失课程的连续性。

在内存模块中，第 8 章从操作系统角度涉及页式内存管理的细节。偏重体系结构的课程可以跳过这一章，而不会损失连续性。类似地，偏重操作系统的课程可以选择淡化第 9 章中对缓存的细节描述。

在存储模块中，面向体系结构的课程可以选择淡化第 11 章中文件系统的内容，而不必担心损失连续性。

在并行模块中（第 12 章），面向体系结构的课程可以跳过多线程的操作系统支持，以及一些高级主题，包括多处理器调度、死锁以及并发性的经典问题和解决方案；类似地，面向操作系统的课程可以选择跳过体系结构方面的高级主题，例如多处理器缓存一致性、并行机的分类以及互连网络等。考虑到并行性的重要性，在任何课程中，只要时间许可，应尽量覆盖这一章的全部内容。

在网络模块中（第 13 章），面向体系结构的课程可以跳过传输层和网络层的细节（分别是 13.6 节和 13.7 节）。面向操作系统的课程可以选择少讲一些协议栈的链路层（13.8 节）和网络硬件（13.9 节）的内容。

## 本教材在计算机科学课程体系中的位置

图 P-1 显示了计算机系统的抽象层次。我们可以尝试将图 P-1 中的不同层次的抽象与传统计算机科学课程体系中的课程相关联。诸如基础程序设计、面向对象程序设计、图形学以及 HCI（人机交互）的课程通常使用较高层次的抽象。通常计算机科学和计算机工程的课程体系中包含数字电路和逻辑电路设计的基础课程，然后是计算机组成原理，介绍计算机的硬件设计。在计算机组成原理课程之上（在图 P-1 的抽象层次级别之上），大部分课程使用烟囱方法：不同的课程分别覆盖计算机体系结构、操作系统和计算机网络的高级概念。


图 P-1　计算机系统中的抽象层次

今天，设计计算机系统已经是软硬件集成的过程，这使人们对烟囱模式提出了质疑，特别是对计算机科学本科的课程体系中学生发展专业技能的早期。

以本书为基础围绕上述主题的课程是一种独特的尝试，用集成的方法在计算机系统的入门课程中介绍中间层次的概念（覆盖了图 P-1 中的深色区域——系统软件及其与计算机体系结构的关系）。这门课程将为渴望学习计算机体系结构、操作系统和网络中的高级主题（图 P-2）的学生提供坚实的基础。

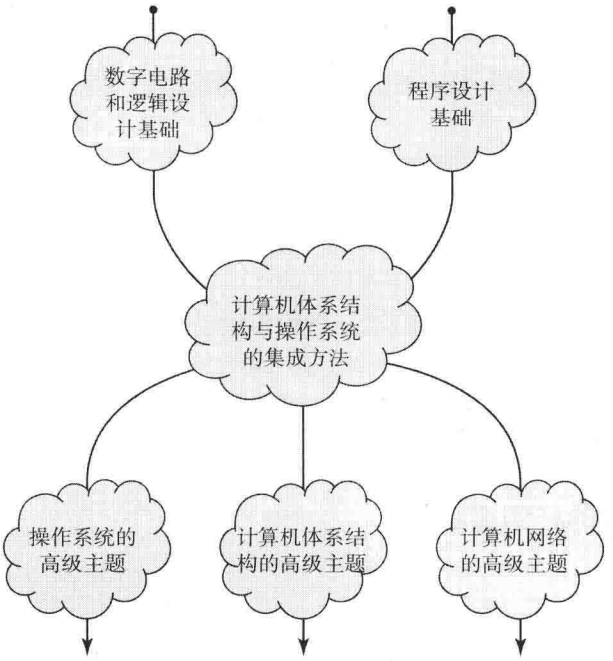



图 P-2　系统课程系列

使用本书内容的课程的先修课程很直接：逻辑设计基础和高级语言程序设计（最好是 C 语言）基础。换句话说，对在本书内容之上和之下的抽象层次需要有基本的理解（见图 P-1）。

在数字电路和逻辑设计基础以及程序设计基础方面都有非常优秀的教科书。类似地，在计算机体系结构、操作系统和计算机网络的高级主题方面，也有优秀的教科书。**唯独缺少的是对计算机系统进行简单、集成化的介绍，使其成为基础课程和高级主题之间桥梁的图书。本书的目标就是成为这样一座桥梁。**

计算机科学作为一门学科其边界已经扩展了。相应地，学习计算机科学的学生的兴趣也各不同。计算机科学的课程需要为学生在本科阶段的学习提供不同的选择。另一方面，课程也有责任保证，不论学生的选择是什么，都能学到计算机系统（广义）的核心知识。我们相信基于本书的课程可以满足这样一种系统核心知识的要求。如果正确地讲授本课程，可以给学生提供充足的机会，通过其他课程来深入学习计算机系统。例如，我们建议在大学二年级开设将本书作为教材的课程。在大三的时候，学生可能可以学习更加面向设计的课程——针对体系结构、操作系统或是网络——以他们在大二从本书中学到的基本概念为基础。最后，在大四的时候，学生可以选修在这些领域中更具概念性的高级主题课程。

本书在体系结构和操作系统的内容方面是大致平衡的。我们认为，计算机科学专业的学生在本科期间应该对这两方面同等重视，不管他们的职业目标是什么。当然，希望成为系统架构师的学生必须了解本书中介绍的软件和硬件之间的互动。即使是希望进行软件开发的学生，了解这些知识对于成为更好的程序员也是必需的。但是，这取决于每个老师对这两个主题强调的程度。好消息是，本教材允许教师选择他们认为必需的课程深度，以与他们所在学校的课程结构相适应。例如，如果教师选择减少体系结构方面的内容，可以很轻松地简单介绍处理器实现的有关章节（第 3 章和第 5 章），而不必担心内容的衔接问题。在讨论本书结构的时候，我们已经对五个模块给出了类似的建议。

## 讲授系统的集成课程的补充材料

我们充分理解教师在讲授需要介绍体系结构、操作系统和网络的计算机系统的集成课程时所面临的挑战。

为此，我们已经提供了一组在线资源。[⊖]我们已经讲授了 11 年本课程，每年 3 次，作为所有计算机专业学生的必修课，因此我们已经积累了相当多的在线资源。

1）我们有本课程所有内容的 PowerPoint 讲稿，使得准备课程和转换（从原有的烟囱模型）更加容易。

2）每个模块都有一个重要的实验部分。我们提供了这些已经迭代过多次的实验的详细描述，以及用于实验特定方面的软件模块（例如模拟器）。

3）除了每章后的练习题之外，我们针对本课程的不同模块还有附加的问题集、家庭作业以及本课程迄今为止的期中和期末考试题。

## 在补充材料中包含的样例实验

### 处理器设计

我们给学生提供一个完成了 90% 的处理器数据通路设计。通过完成数据通路可以帮助学生熟悉相关设计。然后他们要设计基于微码的控制逻辑（使用类似 LogicWorks 的逻辑设计工具），利用数据通路实现一个简单的指令集。这能帮助学生理解数据通路的工作原理并体会一些设计权衡。学生会得到真实电路设计的经验，并通过逻辑设计软件内置的模拟器对设计进行功能测试。

### 中断和输入 / 输出

学生在第一个实验的基础上增加电路以实现中断系统。然后他们（使用汇编语言）写一个中断处理程序。实验的电路设计部分再次通过 LogicWorks 软件系统实现并进行功能模拟。此外，我们还给学生提供了处理器模拟器，他们需要在其中增加中断支持，并与他们用汇编语言写的中断处理器程序一起工作。这个实验不仅可使中断系统的操作变得清晰，还展示了底层设备输入 / 输出的基本概念。

### 虚存子系统

学生在处理器模拟器上实现虚存子系统。在这个实验中，学生可通过实现和实验不同的页替换策略，获得开发操作系统中内存管理部分的经验。这个实验是用 C 语言实现的。

### 多线程操作系统

在我们提供的模拟器上，学生实现多线程操作系统的基本模块，包括 CPU 和 I/O 调度队列等。他们可实验不同的处理器调度策略。这个模块是用 C 语言和 pthread 实现的。学生可从实验中获得并行编程经验，并接触不同的 CPU 调度算法。

### 可靠传输层

学生在我们提供的一个模拟的网络层上实现一个简单的可靠传输层。在传输层必须处理

---

⊖ 关于本书教辅资源，用书教师可向培生教育出版集团北京代表处申请，电话：010-57355169/57355171，电子邮件：service.cn@pearson.com。——编辑注

的问题包括损坏的包、丢包以及乱序到达。这个实验也是用 C 语言和 pthread 实现的。

## 注意

在开始探索计算机系统内部的旅程之前，我们要提醒读者注意：在展示计算机系统设计的教科书中，习惯上会通过有数字的例子来说明和支持相关概念。历史可以揭示未来。如果说在技术发展中有什么东西不变的话，那就是变化。当你买了一辆新车，在车驶出展厅的那一刻，这辆车就变成了二手车。同样地，我们使用的任何有数字的例子中的数字，如处理器速度、内存容量或是外设的传输速率马上就会过时。真正不变的是原理，这也是本书的核心内容。一个让人欣慰的因素是，尽管绝对数字可能会随时间变化，从 MHz 到 GHz，从 MB 到 GB，相对数字随着技术的发展相对保持不变，这使得书中的数字示例也具有持久性。

## 致谢

我们极大地受惠于若干国内外同行，他们直接或间接促成了本书的出现。首先，我们要感谢 Yale Patt，从 2004 年夏天我们介绍了在佐治亚理工学院开设的这门课程后，他用具有无与伦比的说服力的方式告诉我们应该把课程的内容写成教材，因为大家迫切需要一本用集成方式介绍系统概念的图书。我们可以很诚实地说，如果没有他的鼓励，我们可能不会走上写书这条路。下面这些其他学校的同行也鼓励我们进行本书的写作，因此需要特别致谢：Jim Goodman（威斯康辛大学麦迪逊分校和新西兰奥克兰大学），Liviu Iftode（Rutger 大学），Phil McKinley（密歇根州立大学）以及 Anand Sivasubramaniam（宾州州立大学和 TCS）。我们要特别感谢 Jim Goodman，他仔细阅读了本书手稿的早期草稿，并提出了详细的反馈，使本书的叙述得到了极大的改进。除了这些人以外，我们还从其他学校的一些同行那里得到了很多对本书实验的积极支持。

写书的第一步是创建一份书稿供佐治亚理工学院的学生内部使用。对选择佐治亚理工学院 CS 2200 课程的学生，我们怎么感谢也不为过。从 2005 年春季学期开始，几代学生使用了本书的在线版本，并提出了反馈意见，对改进本书表达的清晰性、精炼例题、提供读者可能有兴趣的历史链接等起到了重要的作用。此外，有 3 名本科生帮助绘制了本书中的部分插图：Kristin Champion、John Madden 和 Vu Ha。

计算机学院的部分同事，包括 Nate Clark、Tom Conte、Constantine Dovrolis、Gabriel Loh、Ken Mackenzie 以及 Milos Prvulovic，对本书给予了建议和有洞察力的评论，帮助本书的叙述更加清晰。我们受惠于 Constantine Dovrolis 对本书网络一章早期版本的建议和反馈，使得我们不仅改进了内容，还改变了这一章的叙述顺序。Ken Mackenzie 的建议让我们在第 3 章的处理器设计中给出了一种简单的控制方法。Tom Conte 对流水线一章给出了详细的评论，帮助我们更清晰地表达内容。北卡州立大学的 Eric Rotenberg 为流水线一章的早期草稿提出了非常有意义的反馈。Junsuk Shin 写了本书附录中的简单客户端 - 服务器的套接字代码。我们向他们所有人表示特别的感谢。

我们要感谢佐治亚理工学院，以及计算机学院的远见卓识，鼓励我们在教学方面进行创新。实际上，从 1996 年开始对本科课程体系的改革使得我们开始批判性地思考应该如何教育本科生并了解在课程体系中缺少了什么，这最终导致我们开发了第一门集成方式的系统课程，包括体系结构、操作系统和网络等。

作为图书出版方面的新手，我们从成功的教科书作者那里学到了经验。我们需要特别感谢 Yale Patt（德州大学）、Jim Kurose（麻省大学）、Jim Foley（佐治亚理工学院）、Andy van Dam（布朗大学）、Sham Navathe（佐治亚理工学院）、Rich LeBlanc（佐治亚理工学院）和 Larry Synder（华盛顿大学）等。我们怎么感谢他们都不为过，他们分享了很多经验，包括出版社的选取、与编辑的合作、为可能的评阅人编制问题，以及如何有效地利用评阅意见修订书稿。

书稿经过了几轮的外部评审。大部分匿名评阅人深思熟虑而且有技巧地精准指出了改进书稿的方式。我们对匿名审稿人付出时间和精力帮助本书最后成型表示非常感谢。

特别感谢 Addison-Wesley 出版我们这本教科书。Matt Goldstein 是一个极好的编辑，他负责本书的评阅流程，并建议我们如何修改书稿。他具有一种既能督促我们工作，又不显得傲慢的独特风格。当我们没有按计划完成任务时他表现出了极大的耐心，并对本书背后的愿景给予了毫无保留的支持。我们要感谢 Marilyn Lloyd，Pearson 的高级产品经理，他负责我们的教科书产品。我们还要感谢 Pearson 的 Jeff Holcomb、Chelsea Bell 和 Dan Parker。作为管理产品流程日常事务的项目经理，Aptara 公司的 Dennis Freee 以及 Apatara 公司的职员，包括 Jawwad Ali Khan 和 Rajshri Walia，以及 Write With 公司的 Brian Baker，都值得特别提及。他们为本书尽快生产印刷做出了贡献。

最后，我们要感谢我们的家人，他们的爱、理解与支持使我们能够持续撰写本书。补充一点，Umakishore 的父亲是一位著名的小说家（笔名“Umachandran”），他著有多本小说，对他的回忆是写作本书的灵感。