HZ BOOKS
华章教育

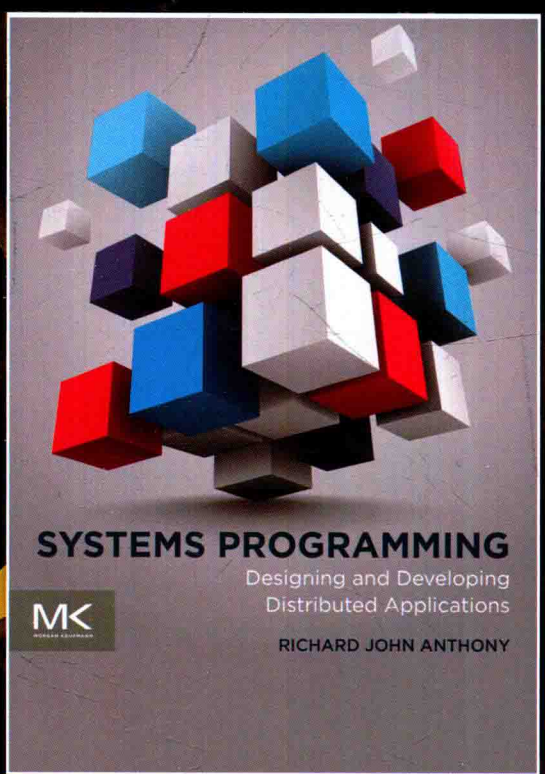
计 算 机 科 学 丛 书

ELSEVIER
爱思唯尔

系统编程
分布式应用的设计与开发

[英] 理查德·约翰·安东尼（Richard John Anthony） 著
张常有 封筠 等译

Systems Programming
Designing and Developing Distributed Applications



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

系统编程

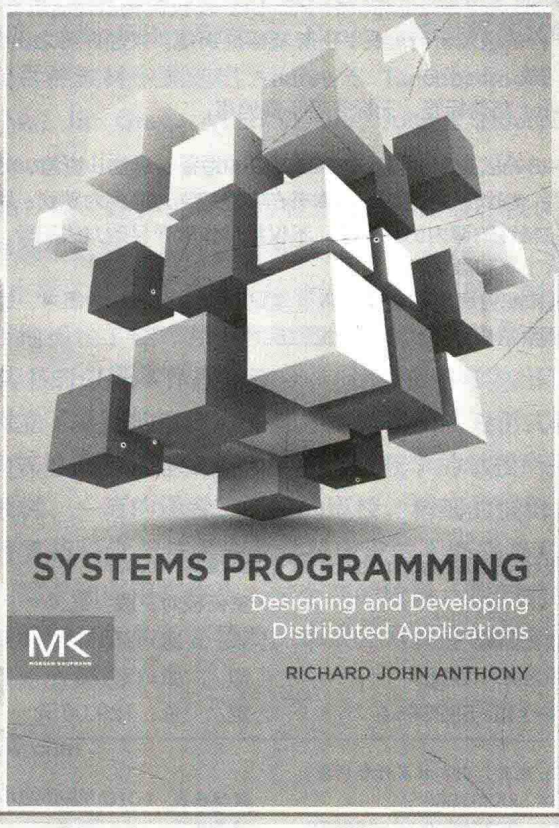
分布式应用的设计与开发

[英] 理查德·约翰·安东尼（Richard John Anthony）著

张常有 封筠 等译

Systems Programming

Designing and Developing Distributed Applications



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

系统编程：分布式应用的设计与开发 /（英）理查德·约翰·安东尼（Richard John Anthony）著；张常有等译 . —北京：机械工业出版社，2017.9
（计算机科学丛书）
书名原文：Systems Programming: Designing and Developing Distributed Applications

ISBN 978-7-111-58256-4

I. 系… II. ① 理… ② 张… III. 分布式操作系统 IV. TP316.4

中国版本图书馆 CIP 数据核字（2017）第 254465 号

本书版权登记号：图字 01-2015-6381

本书用系统思维讲解分布式应用的设计与开发，以“进程、通信、资源、体系结构”四个视角为核心，跨越不同学科的界限，强调系统透明性。本书在实践教学方面尤为独到：既有贯穿各章的大型游戏案例，又有探究不同系统特性的课内仿真实验；不仅提供步骤详尽的方法指导，而且免费提供专为本书开发的 Workbench 仿真工具和源代码。

本书自成体系的风格和配置灵活的实验工具可满足不同层次的教学需求，适合作为面向实践的分布式系统课程的教材，也适合从事分布式应用开发的技术人员自学。

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：朱秀英　　责任校对：殷 虹
印　刷：中国电影出版社印刷厂　　版　次：2017 年 11 月第 1 版第 1 次印刷
开　本：185mm×260mm 1/16　　印　张：27.75
书　号：ISBN 978-7-111-58256-4　　定　价：129.00 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，Brian 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

华章教育

华章科技图书出版中心

译 者 序

Systems Programming: Designing and Developing Distributed Applications

本书的作者 Richard John Anthony 就职于格林威治大学（University of Greenwich），通过本书，他分享了 13 年的分布式系统教学和开发经验，以及多年来开发的配套模拟实验工具，还有用 C++、C#、Java 三种语言编写的实例程序的源代码。

随着互联网的加速发展，分布式系统已经延伸到人们工作、生活的深处。本书融会了分布式系统设计与开发方面完整的知识体系，以四个核心视角（进程、通信、资源、体系结构）为基础，再汇总到分布式系统，最后用两个编程实例总结。本书特别强调理论与实践的紧密结合，在实践活动中借助作者提供的 Workbench 教学工具，可深入理解分布系统的理论知识和运行行为。透明性是分布式系统的重要目标之一，这一目标贯穿了全书各章节。

张常有（中国科学院软件研究所）和封筠（石家庄铁道大学）是本书翻译工作的主要负责人。本书的前言由张常有负责，党云龙参加；第 1 章由张常有负责，许家栋参加；第 2 章由段淑凤负责，党云龙、胡晶晶、王婷、刘博参加；第 3 章由张常有负责，李震宇、居文军参加；第 4 章由封筠负责，段淑凤、王峰、王兵茹参加；第 5 章由封筠负责，王兵茹参加；第 6 章由张常有负责，党云龙参加；第 7 章由张常有负责，籍晨晖、赵朝栋参加；索引由张常有负责。此外，张常有还负责全书的统稿和审校。段淑凤和党云龙为翻译过程的组织付出了努力，感谢他们的贡献。

在翻译过程中，我们尽量反映原作者的思想和风格，同时也融入了自己的经验和理解。希望它的出版能够帮助读者在分布式系统设计与开发方面奠定理论基础，顺利开始实践之旅。

本书非常适合用作分布式系统设计与开发方面的教材，也适合用作相关课程的参考资料。同时，对于正在学习和练习编程的学习者，这也是一本很好的参考资料，因为书中用 C++、C#、Java 三种不同的语言开发了实例和源代码，它们都是分布式程序的范例，可作为项目开发的起点。

由于时间仓促，加之水平有限，书中翻译错误和不准确之处在所难免，敬请广大读者指正。

译者

changyou@iscas.ac.cn

于中国科学院软件园区

本书全面讲解分布式应用程序的设计和开发，主要强调多组件系统的通信特性，以及系统设计与底层操作系统、网络、协议等行为的相互影响方式。

当前，商业乃至全社会对分布式系统和应用日益依赖。对于有能力设计优质解决方案的训练有素的工程师而言，其需求也在同步增长。这需要高超的设计技能和优秀的实现技术。同样，工程师们更愿意让应用程序以全局方式使用系统资源，并受控于宿主系统的整体配置和行为表现。

本书采用综合的方法，讲解了多门传统的计算机科学学科，包括操作系统、网络、分布式系统、编程等，并将需要的背景和理论以多种运行案例形式置入应用程序和系统环境中。这本书是多维的，它具有问题剖析的风格，并通过分布式应用程序用例的开发，实现了理论基础与实战之间的平衡。

通过穿插的实践活动，读者在阅读中真正体验了书本内容、实验操作和模拟执行。在这些实践环节中，系统的动态特性以生动的方式呈现，可以传达更多信息，让系统的复杂特性变得容易理解。大多数配套实验和模拟是用户可配置的，以支持“假设”探究，留给读者深入理解的机会。实践性编程的挑战涵盖系统的诸多方面，包括构建完整的分布式应用程序。本书提供了文档齐全的示例源代码以及清晰的任务指南，通过扩展示例代码来添加功能并构建系统，使这些挑战变得易于教学。

初衷和目标

分布式应用程序的设计与开发是计算机科学领域中的交叉课题。从根本上说，它基于的概念和机制抽取自几门传统的核心学科方向，包括计算机网络、操作系统、分布式系统（理论而非开发）以及软件工程。目前，绝大多数高质量教材只关注单一学科方向，有传统意义上划定的清晰范围边界。它们多数在风格和方法上以理论为主。

编写本书初期，我已讲授了多年分布式应用这门面向实践的课程，很清楚没有一本教材能以实践为中心，全面讲解分布式应用程序的设计和开发这一主题。实际上，我当时想要的是一本指导书，既用作我自己课程的主教材，也能为其他喜欢的人所使用。我也曾想有一本能被我的学生读懂的书。学生是多样性群体，他们的学习经验不同，自信心也有强有弱。我曾想，用一本书来鼓励那些在软件工程方面刚刚起步的人，同时，也能满足那些期望更高挑战的、较有经验的学习者的需要。我的课程强调理论与实践相结合，教学工作开展 13 年来，效果良好，并深受学生喜爱。有时，在讨论课程配套教材缺乏可用性时，学生们建议我自己直接在这门课程的基础上编写一本。

本书内容填补了一项清晰定义的空白。它是一本综合性教程，以“自成体系”方式讲述了多种底层概念，以便读者能在领会跨系统全局的同时，理解关键基础理论，并能在支撑实践活动中进行探索。这全部源自“自成体系”。就这一点而论，本书区别于其他主流教材。传统教材倾向于专注单一的传统主题领域，更侧重理论性基础教学。

本书专门用于讲授以“理论与实践相结合”为重点的分布式应用程序设计课程。本书主

要聚焦于应用程序开发，以及为确保高质量教学效果而必需的支撑知识。这种组织方式使得本书能自然地连接计算机科学的相关领域。它没有尝试像传统方式一样组织教材（例如，仅关注网络或者操作系统），也没有试图囊括该领域尽可能宽广的内容（传统教材往往如此）。相反，它提供了横跨这些学科的非常重要的集成。本书的主要设计专注于易于教学，基于示例程序来阐述分布式系统和应用的关键特性，同时基于案例研究、互动教学工具和实践活动来展开细节讨论。主要目的是便于读者理解基于套接字应用的实际示例程序，进而起步开发他们自己的应用程序，并以此作为与书本阅读同步的指导性环节。

本书的理论方面和大部分实践方面具备跨语言的可移植性，但其实现方面有语义上的语言依赖性。为尽可能易于学习，部分示例代码采用了 3 种流行的编程语言：C++、Java 和 C#。

附加资源代码库内容丰富，包括各种课内实例的示例程序代码和章末编程任务的示例方案，以及全部三个案例研究的完整源代码。

附加资源还包括作者搭建的教学工具 Workbench 套装的特定版本。这个工具可用于课内活动，也可用于不同主题的独立研究或导师指导下的科学探索，或者用于给课堂教学或实验项目注入活力。Workbench 的灵感源自对以逼真且易理解的方式表现系统动态特性方面的需求，曾经尝试用一系列静态图表讲授调度（一个包含更多动态特性的例子）的教师，一定能体会到静态方法的局限性。静态方法在表达动态行为所能展现的真正含义方面存在困难。Workbench 专门为克服系统动态性和复杂性教学方面的局限性而设计，支持用户自行配置具体实验和模拟，涵盖了网络、分布式系统和操作系统领域的很多不同内容。每章都包含相应的实践活动，引导读者在实践学习和基础理论概念之间建立联系。

本书非常强调针对核心理论的有指导的实践探索，使其既适合用作自学教程，又适合用作课程的辅助教材。

预期读者

本书的预期读者对象比较广泛，包括：

- 讲授分布式系统的教师，需要一本“自成体系”的教材，包含实践活动、编程练习和案例研究，能用作生动有趣的课堂读物。
- 正在学习应用程序开发的学生，需要一本书汇聚分布式系统、操作系统、网络等多个不同方面，配以众多清晰的实践案例和丰富的样例代码库。
- 对于分布式应用程序设计与开发或套接字编程有经验的程序员，或许需要一本快速入门资料，以启动一个分布式应用项目。
- 正在学习 C++、Java 或 C# 语言的初级程序员，期望提高挑战性，编写套接字网络应用程序。
- 熟悉本书支持的语言之一的套接字程序员，需要仿照样例程序资源，以便在两门语言之间展开交叉训练。
- 学习计算机科学其他方向的学生，希望在分布式系统方面找到自学导读形式的、注重实践的基础读物。

组织结构

本书的核心部分包括四章（第 2～5 章，即四个视角），涵盖背景概念、技术需求、当前挑战以及构建分布式应用程序的必备技术和支持机制。本书选定这四个特定视角，以便材料

分门别类，从设计和操作角度看都很有意义。该方法使读者能够以结构化的思维方式仔细理解系统的底层概念和系统机制，跨越了传统教学学科的界限。

随后的第 6 章定位于高级分布式系统本身。该章的重要工作是将前述核心章节中讨论的思想、概念、机制等融进总体系统环境，并关注公共服务，以确保系统在功能和非功能需求方面的高质量。

各章都强调实践，穿插着实验和实践探索活动，以及一个贯穿各核心章节的案例研究，从而整合和交叉连接各章。为拓展介绍分布式应用程序的体系结构、组成、行为、运行环境等，最后一章附加两个容易操作的案例研究，并提供清晰的文档和完整代码。

绪论（第 1 章）开始本书，也启动了本书采用的综合系统方法。它针对分布式计算的兴起及其在现代环境中的重要性，提供了简明的历史回顾。它还简短地介绍了一些将在后面章节中深入讲述的关键主题。但对于读者来讲，在起初建立基本的认识是有必要的。这些基本主题包括：分布式系统的大体特性、分布式系统的主要优势、构建分布式应用程序必须面临的主要挑战、分布式系统的质量和性能的度量标准以及透明性的主要形式等。其中还介绍了三个案例研究、本书配套网站上的教辅材料、课内实践活动以及互动教学工具 Workbench 套件。

进程视角（第 2 章）考察进程的管理方式及其对低层通信的影响，涉及进程调度与阻塞、消息的缓冲与传送、端口和套接字的使用以及进程与端口的绑定机制等。绑定机制使操作系统能代表本地进程，在计算机层面管理通信。还讨论了多进程环境概念、线程以及操作系统资源，如定时器。

通信视角（第 3 章）考察网络和通信协议的运行方式，及其功能性和行为对应用程序的设计和表现的影响。本视角关注的主题包括：通信机制和不同的通信模式，比如单播、多播和广播，且选定的方式能影响应用程序的行为、性能和扩展性。描述了传输层协议 TCP 和 UDP 的功能和特点，并对比了其性能、延迟和开销方面。从开发者的视角，检测了通信的底层细节，包括套接字 API 原语的作用和操作。同时，还检测了远程过程调用和远程方法调用等更高级的通信机制。

资源视角（第 4 章）阐述计算机系统资源的本质及其对分布式应用程序通信的帮助作用。感兴趣的物理资源包括处理能力、网络通信带宽和内存。讨论聚焦于有限资源的高效使用方面的需求等，它直接影响到应用程序和系统自身的性能和可扩展性。还讨论了内存在消息的组装、发送、接收缓冲器方面的使用情形。

体系结构视角（第 5 章）阐述分布式系统及应用程序的结构。主要聚焦于把应用程序逻辑划分为若干功能组件的各种模型，以及这些功能组件互连互访的方式。还讨论了系统组件映射到系统底层资源的方式，以及映射关系引出的附加功能需求，例如，对灵活的运行时配置的需求。分类讨论了不同架构模型对关键非功能性质量度量的影响，如可扩展性、健壮性、效率、透明性。

分布式系统（第 6 章）紧随四个核心视角章节，形成了前面各章节的大背景。四个视角分别阐述了某一方面的支撑理论的特性、概念和机制，而这一章定位于较高层面，重点关注分布式系统本身，及其关键特征和功能性需求，以及与设计和开发相关的特殊挑战等。因此，该章从更宽泛的系统角度审视核心章节的内容，讨论因分布式特性本身而导致的问题，以及解决这些问题的技术。“透明性”是追求分布式应用程序质量的关键，因此“透明性”成为贯穿各章的主题，既与各章涵盖的不同专题相关，同时也成为案例研究讨论中的主要关

注方面。为进一步强调其重要性，这一章全面讲述了“透明性”，使其成为一个独立的主题，定义了“透明性”的10种重要形式，并根据各自的重要性以及对系统和应用的影响方式进行探讨，解释了实现不同形式的“透明性”需要采用的技术。此外，还介绍了公共服务、中间件以及异构环境下支持互操作能力的技术。

案例研究：融会贯通（第7章）以两个详尽的分布式应用程序案例研究的形式，把前面章节的内容联系起来。以这两个案例为载体，阐明了许多不同的问题、挑战、技术以及机制。目标是提供一个综合的全貌，用于审查应用程序贯穿的全生命周期。这一章采用了“问题－求解”的方法，该方法基于提供的可行应用程序、程序源代码以及详尽文档。这些应用程序建立并加强了理论与实践之间的联系，阅读过程中有必要参考前面的章节。

课内活动

一系列实践活动贯穿于第2～6章，用于借助实验强化关键概念。各活动都基于特定概念或机制植入系统或应用环境中。

如何使用本书

本书采用这样的设计风格是为了灵活地支持读者需求的广泛性和多样性。建议用途如下。

用作一门课程的主教材。这是编写本书的主要动机。本书以一门成功的“设计和构建分布式应用系统”课程为基础，全面讲述了分布式应用程序的核心主题，辅以详细的实践活动、可用示例、案例研究、章末问题以及带解决方案的编程挑战等。本书还包含范围广泛的基础支撑技术资料，包括概念、问题、机制以及需要理解的系统特性相关策略。

鉴于采用了交叉综合方法并提供了扩展的资源库，本书准备了理想的课程大纲。教师可以根据他们希望达到的技术深度和学生的知识水平，围绕本书灵活地编排课程。本书的资源网站上提供了以C++、Java和C#三种语言编写的示例程序的源代码。

基于Workbench的活动可用于课堂演示或指导性练习，或者用于指定的自学过程。这些环节也可以用于模拟调度、线程优先级或死锁等活动，用图解方式解释相关的动态行为（众所周知，有些内容很难仅使用静态图表的手法教会学生）。生动的模拟可以产生令人兴奋的教学效果。比如，通过暂停模拟，请学生猜想接下来会发生什么，或者通过多次使用不同条件运行模拟实验，展示特定配置怎样影响系统行为。

网络版Workbench实验可以用来演示在进程间建立通信和通过网络发送消息的过程。它可用于探索套接字原语的使用，诸如阻塞和缓冲机制的性能特性，以及TCP和UDP传输协议的行为。例如，在课堂上，可使用动态的真实多进程，探索TCP中“绑定－侦听－连接－接受”序列对建立连接的必要性。

Workbench活动还可用于建立多种多样的学生实验室练习或家庭作业，例如，用于评估通信效率、通信死锁可能发生的情景，或者可能使用详尽的统计结果，从效率、公平性、吞吐率方面对比多个调度算法。

分布式系统版Workbench可帮助我们搭建分布式实验，如一个运行在实验室中全部计算机上的选举算法实验。学生们能够测试一些场景，如终止当前主进程并预测下一步表现。我们可以启动多个主进程实例，观察算法如何应对这种情形。配备的日志程序记录进程状态变化的运行时序列，用于后期分析。另一个例子是探索“客户端－服务器”游戏应用程序，

并要求学生根据观察到的游戏过程，尝试通过逆向工程方法推测出发生的消息序列。这类挑战非常有利于鼓励深度探索。

全部用例均可运行，且可改编成学习和评估练习，或者作为开发更大型项目的起点。

章末编程练习可用作测评活动或补习活动。既可以直接使用，也可以强化挑战，视班上学生的知识水平而定。

用作自学辅导书。本书是一本独立学习的理想辅导书。书中提供了大量辅助实践的技术文档，便于读者自学。阅读过程中对照示例解决方案（针对编程挑战）、解释预期输出结果（针对课内活动）和章末问题答案，能够检查自己的学习进展。Workbench 提供的实验和模拟工具支持用户自行配置参数，便于自学者保持自己的学习进度，同时探索自己想要的深度。开发 Workbench 的主要原因之一是我发现大家的学习速度各不相同。对于相同内容，有的读者轻松领会，而有的读者却可能起步蹒跚。因此，Workbench 支持个性化和渐进式的学习。

通过提供以三种流行的语言编写的示例应用程序，本书还可以用作套接字 API 和传输层协议（TCP 和 UDP）的“罗塞塔石碑”——一旦读懂其中一门语言的实现，就能够贯通使用该示例的其他语言版本。这对特定场景很有帮助，例如，正在开发的系统中有一部分组件用一种语言开发（比如服务器端用 C++），而另一部分组件用另一种语言开发（如客户端用 C#）。在第 7 章中最后的案例研究中，专门强调了异构特性和互操作特性。从制定自学目标和学习效果的角度看，这些用例是一种强有力的学习资源，能帮助读者逐个掌握这些知识点，并且完全按照适合自己的时间进度，兼顾了读者在这方面的起步知识准备。

用作辅助教材。本书的主题交叉集成了传统的主题领域，包括网络、操作系统、程序设计、分布式系统理论。鉴于其清晰的解释和对实践能力的重点强调，本书是一本理想的辅助教材，适用于网络、应用程序开发、操作系统等课程，附加的多种活动和编程练习更添加了趣味性。与强调传统知识结构和面向理论的教材相比，它风格迥异，适合与其他专门的教材共同使用，为学生补充不同的信息来源。同时，还有助于在一定程度上填补主要教学内容与其他课程之间的交叉形成的缺口（本书很擅长）。

基于 Workbench 的活动都适合单独拿出来使用。例如，读者也许只选用操作系统版 Workbench 活动的一个子集（或者其他子集），为部分难学的课程内容添加些生动性，或者只是想用现场实验或模拟替换一长串的幻灯片。

用作带实践示例的参考教材。本书讲述了分布式系统领域宽泛的主题，涉及计算机网络、程序设计和操作系统等。本书有别于主流同类教材的是宽泛且大量的实践示例和源码资源。因此，它可用作一本变革性的参考指导书。对于大部分主题，读者都能找到相关的带指导的实践环节，或者带解决方案的编程挑战，或者一个或多个用例形式的上下文情景。

教辅资源[⊖]

本书配套网站上提供了教辅材料，网址为 booksite.elsevier.com/9780128007297。

材料在网站上的组织方式对应于书中相关章节，类型如下。

示例程序代码。本书为课内活动、相关举例、用例程序、编程练习的解决方案提供了示例代码，可按以下方式使用：

⊖ 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师访问爱思唯尔的教材网站 textbooks.elsevier.com 进行申请。——编辑注

- 多数情况下，提供了完整的应用程序源代码，使读者能够学习整体应用程序逻辑，并把它和教材中的解释相联系。有些情况下，教材中提供了简短的代码片段来解释一个关键知识点，那么，读者可以检测整个应用程序代码，把程序片段插入程序逻辑的正确位置。许多示例代码都以三种语言提供：C++、Java、C#。
- 示例应用程序代码可用作开发章末练习解决方案的起点，在这种情况下，我们给出了最合适资源的相关指导。
- 也有一些针对章末编程练习的特定的示例解决方案（如果这些解决方案还没有在其他部分资源中给出示例）。

可执行文件。许多应用程序也提供了可执行文件的形式。读者可以直接运行应用程序，研究其表现，而不用先编译。在读者跟随着课内活动和例子学习时，这点尤其重要，它减少了花费在阅读和实践之间的切换时间。

Workbench 教学工具。本书配有一组复杂的交互式教学应用程序（称为工作台）。我已经花了超过 13 年的时间开发它，提升了以学生为中心的灵活的教学方法，使学生能够远程、交互式、按照自己的节奏学习。三种版本的 Workbench 提供了可配置的模拟、仿真、实现组合功能，以方便读者针对系统中的许多基本概念展开实验。目前，Workbench 已经在许多学生群体中尝试和测试，在一些教师的课上也用作了辅助支撑工具。书中特定的技术内容与特定的练习和实验相关联，可用 Workbench 软件工具开展实验。例如，网络版 Workbench 中有特定的实验，可处理寻址、缓存、阻塞或非阻塞的套接字 I/O 模式以及 TCP 和 UDP 传输协议的操作。

致谢

我谨向技术评审人员表示诚挚的感谢。他们抽出宝贵时间仔细阅读书稿，并提供了一些非常有价值的反馈意见和建议。

我还要感谢 Morgan Kaufmann 编辑团队在规划和撰写本书的过程中给予的建议和指导。

同时也要感谢过去学习我的系统编程课程的许多学生。他们提供了关于课程结构和内容的反馈，或者直接通过评论与建议，或者间接通过他们日后所取得的出色成就。特别感谢那些多年来建议我写这本书的学生们，我终于做到了。

还要感谢许多同事、朋友和家人给予我的鼓励。特别是，永远感谢可爱的 Maxine 一直以来给予我的极大耐心和支持，以及在我学习期间的端茶倒水。

第 1 章

Systems Programming: Designing and Developing Distributed Applications

绪　论

本书自成体系，是一本分布式应用程序设计与开发的导论读物。书中汇集了来自多个关键主题领域的基本支持理论。多种相关材料以带有清晰说明的举例方式，放置在真实的应用场景及其上下文环境中。全书都特别强调实用性，包含了读者参与的编程练习和实验。同时，还配备了三个功能完整的详尽案例研究，其中一个案例贯穿四个核心章节，把理论知识放到应用程序视角，横跨各章的不同主题，交叉链接。本书是本科学位课程的理想配套教材。

本章介绍全书，包括其结构、技术资料的组织方式以及背后的动机等。它提供了一个历史的视角，解释了分布式系统在现代计算中的重要性。同时，本书还阐述了其（采用的）叙述方式的集成特性和跨学科特性，以及其底层的“系统思维”方法。它描述和论证了 4 个精选视角（涵盖系统、结构、组织、行为等）的资料组织方式的合理性。如果因为教学目的将学习内容划分为操作系统、计算机网络、分布式系统、编程等传统分类，就人为地引入了不同学科主题间的边界。本书选择了系统性视角，旨在克服多学科间的人为边界。实际上，很多分布式系统的关键概念都在其中的若干领域之间相互重叠，或者正好处在这些领域间的交叠区域。

本书的总体目标是为读者提供对分布式应用系统的体系结构和通信等知识的全面讲解，以及提供理解各种可选的设计方案的必要理论基础，帮助读者领会不同的设计决策和折衷。读完这本书时，读者将会具备设计和构建第一个分布式应用程序的能力。

通过采用交互式风格，书中的技术内容变得更加生动有趣。本书的交互式风格包括实践活动、举例、示例代码、类比、练习、案例研究。在作者开发的特定版本的 Workbench 教学资源套件中，提供了许多实用的实验和模拟活动。应用程序举例贯穿全书，把概念知识归纳到相应的视角。为支撑这些特点，本书提供了大量源码，以及理论和实际练习相结合的实践活动，来全面提高读者的技能和背景知识。

1.1　基本原理

1.1.1　计算机科学的传统讲授方法

计算机科学是一个极其宽泛的知识领域，涵盖不同的主题，包括系统体系结构、系统分析、数据结构、编程语言、软件工程技术、操作系统、网络与通信等。

习惯上，鉴于大学授课目标的实用性，计算机科学涵盖的主题教学资料都按照上面列举的专题领域（学科）划分。因此，学生将以一门单独课程的形式学习操作系统，计算机网络是另一门课程，计算机编程语言则又是另外一门课程。作为一种结构化的授课方法，这种模式总体上运行得很好，在计算机科学领域得到广泛应用。

然而，计算机系统的许多知识横切多学科边界，从任何单一学科的角度都不能全面领会。因此，为了获得对（计算机）系统工作方式的深入理解，有必要同时整体地跨越多个学