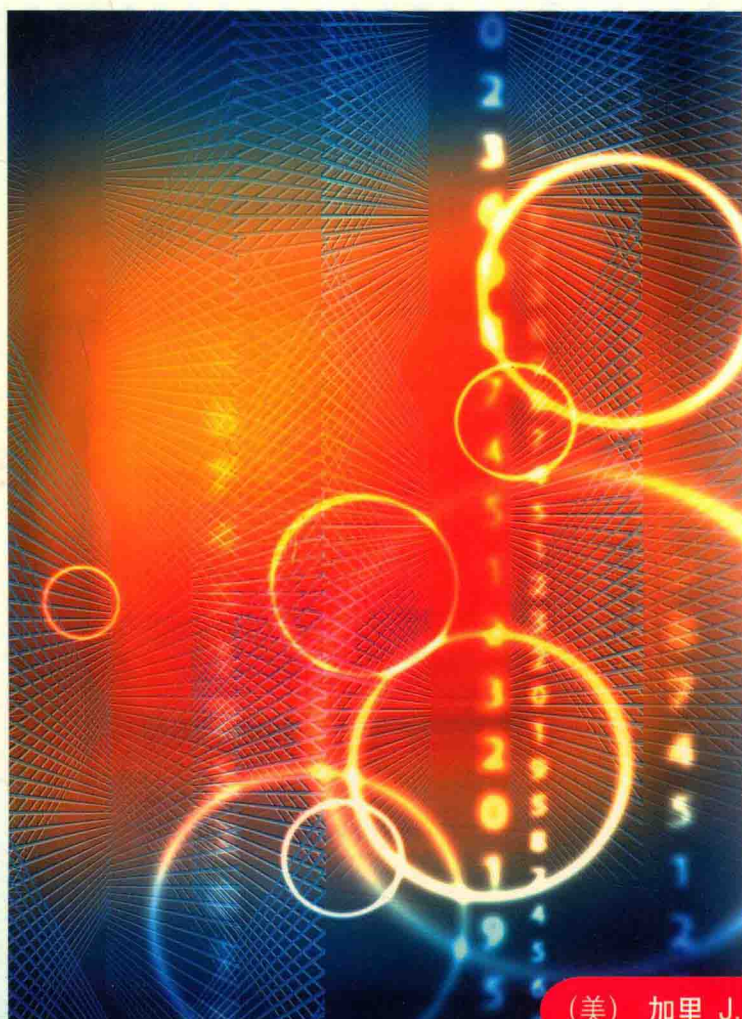


经 典 原 版 书 库

操作系统

(英文版·第3版)



本书只供在
中国大陆销售

(美) 加里 J. 纳特 著



机械工业出版社
China Machine Press



经 典 原 版 书 库

操作系统

(英文版·第3版)

(美) 加里 J. 纳特 著



机械工业出版社
China Machine Press

English reprint edition copyright © 2005 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Operating Systems, Third Edition* (ISBN 0-201-77344-9) by Gary J. Nutt, Copyright © 2004.

All rights reserved.

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

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版由Pearson Education Asia Ltd.授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

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。

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: 01-2005-1180

图书在版编目(CIP)数据

操作系统(英文版·第3版)/(美)纳特(Nutt, G. J.)著. —北京:机械工业出版社, 2005.3

(经典原版书库)

书名原文: *Operating Systems, Third Edition*

ISBN 7-111-15894-6

I. 操… II. 纳… III. 操作系统—英文 IV. TP316

中国版本图书馆CIP数据核字(2004)第141011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: 迟振春

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2005年3月第1版第1次印刷

787mm×1092mm 1/16·58印张

印数: 0 001-3 000册

定价: 79.00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

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、研究及收藏。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，有的还专诚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: hzedu@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
周立柱
范明
袁崇义
谢希仁

王珊
吕建
李伟琴
陆丽娜
周克定
郑国梁
高传善
裘宗燕

冯博琴
孙玉芳
李师贤
陆鑫达
周傲英
施伯乐
梅宏
戴葵

史忠植
吴世忠
李建中
陈向群
孟小峰
钟玉琢
程旭

史美林
吴时霖
杨冬青
周伯生
岳丽华
唐世渭
程时端

秘 书 组

武卫东

温莉芳

刘江

杨海玲

前言

致学生

操作系统是一个令人激动的软件领域，因为操作系统（OS）的设计对计算机的总体功能和性能都有着重要的影响。在初次学习操作系统时，理解所有操作系统设计背后的原理（principle）是非常重要的，而且还要留意这些原理如何在真实操作系统中实现。

本书特色

本书提供了对各种有关的操作
系统原理和实现的一个全面描
述，并在补充内容里提供了背景
信息、真实示例以及编程练习。

- 理解性示例：表现了在UNIX和Windows操作系统系列中如何实际运用书中的原理。
- 实验练习：通过使用Windows和Linux/UNIX，让学生获得亲身体验。

本书的目标就是要提供一个操作系统原理的全面讨论，并补充解决问题的算法、代码和实现工具，通过实验练习来帮助读者理解当代操作系统实践，特别是对UNIX和Windows操作系统的理解。本书试图通过在正文中讨论原理，并把大部分的实践材料放在补充讨论和实验中，把概念性内容与应用性内容区分开来。

问题的核心在于概念性的内容。很多操作系统原理可以使用形式化的（数学的）术语或者在非形式化的讨论中进行描述。非形式化的描述相对容易阅读，但形式化的描述更为准确。例如，字典的一种非形式化的解释可能为：“它是一个术语的列表，同时有各个术语的定义说明。”然而形式化的描述可能指明字典是“一种机制 f ，将某术语 x 映射到该术语的定义 $f(x)$ 。”第一种解释

是直观的，而第二种解释主要关注字典的逻辑描述。第一种描述表明字典是一个列表或表的实现，而第二种描述包含的实现方式范围，可以从表到列表，到关联存储器，到数据库，到Web服务器等。非形式化的定义说明有如一部字典，但形式化的定义说明还适用于编译器符号表。本书的目标是解释操作系统的一般原理，并使读者对如何设计操作系统有一个深入的理解。实现该目标最好使用形式化的描述，因为它集中于概念的逻辑目的，而不是如何实现概念的一个例子。本书在前面的章节中使用非形式化的或专门的叙述来描述操作系统的概念，很少有形式化的描述。但随着讨论的深入，形式化的描述会逐渐增加。概念的形式化讨论中总是伴随有非形式化的讨论和示例。

操作系统是围绕着性能问题进行设计的。如果性能太低，则它是一个失败的操作系统。然而，对性能的详细讨论往往会使概念模糊。因为在学习操作系统时，概念是十分重要的一部分，所以本书在各个部分逐步涉及分析和性能理论。本书将经常提及性能问题，并提供性能问题的非形式化的解释。这将使读者对性能问题有一个直观认识，可以在以后再正式地学习它。如果关于性能的评价符合概念的描述，那么会把它们包括在概念的讨论之中。

如前所述,使用真正的操作系统代码进行实验,有助于理解操作系统概念是如何在真正的系统中实现的。同样本书也提供了两种其他类型的资料,帮助读者学习有关当前操作系统的实践:文本框中的示例和实验练习。

- 文本框中的示例解释了在UNIX、Linux、Windows或者其他操作系统中如何应用或实现这些概念。其中不少示例都是代码例子,有助于读者领会操作系统如何实现书中的理论。大部分例子是来自完整程序的C代码段,其中这些完整程序已经编译和执行过。其他例子使用C语言对算法和实现技术进行了描述,这些描述故意忽略了细节,这些细节在实际实现时是必不可少的,但并不影响对算法的理解。当代码是一个实际的实现时,代码的上下文中会说明,否则总是假定它是一个算法或技术的描述。我已经使用伪代码语言进行了实验描述,但学生和本书的审阅者通常喜欢使用C语言,请留意不要认为用C语言的描述是完整的实现。
- 书中也包括实验练习。每个练习中都提出一个问题,然后提供了解决问题所需的全面的背景知识,并帮助你设计解决方案。这些练习用到了各种UNIX和Windows操作系统,将带给你宝贵的实践经验。
- UNIX还是Windows? 对于讨论哪个系列的操作系统更好这一问题,老师可能有不同的看法。我在教授操作系统课程时,有时使用UNIX,有时使用Windows。无论使用哪一种操作系统,书中都提供了足够的示例以及实验练习。你的老师可能会选择某一种操作系统,并指导你阅读这些示例并解决实验练习。不要轻易跳过其他操作系统的实现细节,它将拓宽你的视野,并对你未来的编程生涯十分有益。

在此提及几个要注意的地方。首先就是有关术语的使用。我们很难避免在高级科学和工程课程上使用一些术语,所以,为了学习高级专题(如操作系统),你至少要知道一些术语。这样在开始操作系统的学习时,你已经积累起一些术语(如“算法”、“随机存储器”、“千兆字节”和“链表”),并能使用这些术语和其他编程人员相互讨论。没有谁知道或能够记住所有技术术语。为了解决这个问题,本书的后面提供了一个术语表,其中可能会遗漏一些术语,如果碰到一些并没有定义的术语,请参看软件术语字典,如<http://inf.astrian.net/jargon/>。

操作系统中有很多复杂的理念。我们在碰上这类系统时,要循序渐进地学习。例如,当你学习如何骑自行车时,必须要学会转弯、停止、平衡还有移动。如果你要写一篇如何骑自行车的论文,则先要粗略地涉及转弯、停止、平衡还有移动,然后再详细描述它们。第一阶段需要说明所有的部分而不用告诉读者太多的细节。第二阶段需要提供转弯的细节(例如,在你慢慢地学习转弯去保持平衡以及向前运动,还有诸如此类的动作时,都需要去转动车把)。操作系统也是很难在一个阶段中学会的。当开始学习进程时,你会对内存和文件感到很惊奇。在“两阶段”策略中,你在第一阶段先了解一下各部分的大致情况,然后在第二阶段开始学习细节。有时还有第三阶段,特别适合于理解这些细节在一个特定的操作系统中如何实现。这种多阶段策略是递归进行的。对本书而言,第1~3章是第一阶段。第4~14章是第二阶段,是操作系统的主要部分。第15~18章也是第二阶段,涉及相关核心技术的主题。第19章也是第二阶段,涉及有关操作系统的设计问题。第20、21章是第三阶段,介绍Linux和Windows NT。每一章的论述也采取了类似的策略:首先对内容进行概述,然后进行详细介绍,最后安排实验练习。

对操作系统的研究一直以来就是计算机科学中最具挑战性和令人激动的领域之一。我希望本书使得操作系统的复杂结构变得容易理解，并且避免使其中简单的内容让人厌烦。祝你在操作系统的学习中好运，并且希望你比我想象得更喜欢这门课程。

致教师

我写这本书的初衷是：我试图寻找一本书，其中有比现有的书中更多的原理。同时，我感到如果学生们不能投身到广泛的操作系统实践中去，那么原理将是难以接受的。这仍然是本书第3版所持有的观点：书中对原理的讨论是综合的（无论从深度和广度），另外增加了一些有关操作系统设计和编程的实质性的辅助信息。

第3版中的变化

本书是在前两个版本和第3版的草稿上，以及前两版的读者的积极指点下出版的。我们的目标是：

- 修订表示方式。花费了大量的时间在解释普通的概念上，使用了更多的类比和实际的例子。
- 提供一个全面而综合的教学内容，是一本十分合适的大学操作系统教材。
- 不像市场上的其他教材，本书有更多的示例和编程项目，扩展了教学方式。

许多使用过本书第2版（特别是实验更新版）的老师建议进程描述部分应该更新，增加多线程的进程内容。有趣的是，第3版草稿的一些审阅者反对将线程内容增加到此书中。（写一本每个人都喜欢的书真不容易！）

现在，大部分的操作系统都支持内核线程（在Linux和其他的UNIX系统中，内核线程由POSIX线程接口提供支持，所以，它看起来就像在用户空间中实现一样）。如果重新描述进程模型时不涉及线程，是我的失职。本书以普通的方式描述了经典进程，并解释经典进程模型如何演化到

简介：第3版中的变化

在本书的3个版本中我们都使用文本框中的信息来提供例子和其他的主题，这些主题通常与本书的主要线索不太相关。有关这些文本框中的信息的反馈是积极的，尽管有时在性能和例子间有些让人混淆。在这一版中使用了一种新的设计，用一个单独的方框来包含例子。我们还增加了许多UNIX和Windows代码示例。

第3版中的文字已重写，其中包含了许多新内容，并以更容易阅读和学习的方式表示了这些内容。

为了反映当代操作系统中线程的使用，几乎每一章都进行了重新组织和修订。在各章中也增加了一些新的内容，来对移动计算、嵌入式操作系统、多处理机和新设备进行讨论。另外，我们还增加了其他的一些重要主题的讨论：为了提供更多的现代安全机制的内容（特别是加密），对安全这一章进行了重写。专门为分布式程序设计运行时系统（distributed programming runtime system）增加了新的一章，其中介绍如何将Java和Microsoft.NET应用到操作系统课程中。最后，在介绍Linux和Windows NT/2000/XP内核之前，我们对原有的材料进行重新组织，增加一章通用的操作系统设计方法学。

在第2版中，我们引进了一套综合的实验练习。本书增加了新的实验练习，使总的练习数目达到了15个。有的实验练习仅适用于UNIX系统，有的仅适用于Windows系统，其中有4个练习对Windows和UNIX系统都适用。

具有多线程的现代进程。本书试图使用“进程”或“进程/线程”来表达单线程经典进程或有一个单线程的现代进程。

对概念和问题的描述部分已完全重写，这使得它们更易理解。专业出版编辑在这上面做了大量工作，确保了本书通俗易懂。我们也增加了更多的图和例子。使用早期版本的读者希望有更多的代码，所以本书中增加了大量的代码，如果不想看，可以跳过。

为了全面探讨操作系统，我们打算加入由ACM/IEEE课程推荐的所有主题。为此我们请求以前的学生和老师告知哪些部分没有涉及到。结果，本书包含了线程、移动和无线计算、嵌入式操作系统技术、新设备的使用，以及更多的有关多处理器方面的内容。本书不仅更新了安全这一章，使得它以更合理的方式表示重要的概念，而且还提供了更多密码系统的讨论。这一版反映了操作系统领域自本书第2版出版以来的大体发展。除了像将进程更新成线程/进程这样的变换外，每一章都进行了修订以反映当代操作系统技术。有的章节改变很少（例如调度和死锁这一章），但是，其他章节均有较大的变化来反映当代操作系统技术（例如，第1、5、6、14、17和18章）。第18章介绍分布式程序设计运行时系统，是全新的一章，它反映了系统软件技术（如Java虚拟机和微软通用语言运行时库等）和主流操作系统技术间的关系。

实验环境

商业化操作系统只有少数被广泛应用。虽然研究这些操作系统很有价值，但在课堂中使用它们进行实验有很多实际的障碍。首先，商业化操作系统非常复杂，因为它们必须对商业应用提供全部的支持。使用此类复杂软件进行实验是不切实际的，因为有时我们很难领会特殊问题是如何在软件内处理的，对代码的任何微小改变都可能对整个操作系统的运行产生不可预测的影响。其次，通常公司对其实现的操作系统软件有明确的专利权保护。结果是，公司可能不愿意提供操作系统源代码给任何希望研究和学习操作系统的用户。

在课堂中，我使用了两个方法来处理这个问题：

- 课程要基于实际操作系统的外部视点。这基本是ACM/IEEE 2001 课程推荐的方法。
- 课程要基于某个“可管理的”操作系统的内部视点。

我已经与很多教操作系统课程的教师讨论过这个问题。关于大学操作系统课程，选择合适的实验是一个普遍的难题。然而，在OSDI会议上，与会者一致同意操作系统的外部视点应该用于初级的操作系统课程。我们所有的实验练习都要求学生编写UNIX或Windows用户空间代码。通过练习应用编程的方式（而不是修改内核代码），学生可以对内核的工作方式有一个特定角度的洞察。对特定的操作系统课程来说，这种方法有额外的好处，并不要求你有“可崩溃的”实验设施。

虽然大家普遍认为在一开始就教授操作系统的内核非常困难，不过许多人仍希望尽可能早地开设关于操作系统内核的课程。如果你决定教授操作系统内核的课程——作为初级或中级课程，那么可供选择的目标操作系统会比较有限。如果你想研究一个真正的操作系统，那最好选择Linux或者FreeBSD，否则只能选择一个教学系统。在补充站点上，我们为一个学期的课程提供了足够的Linux 2.2.12内核内部材料。

实验练习

这本书的每一个版本都试图提供最好用、最流行的概念化材料，以及相应的编程练习。在第1版中，编程练习是作为常规练习出现的，并没有什么背景知识。在第2版中，增加了一些实验练习，这些练习以问题的方式引入，提供了全面的背景材料，并给出了解决方案（我是在

Window NT项目手册[Nutt, 1999]和Linux内核手册[Nutt, 2001]的指导下写这些实验练习的)。这些实验练习为助教提供了辅助教学材料,它们是在学生们仅有基本概念的情况下去教学生解决编程问题的。这允许你在上课时花费更多的时间来讲授基本原理,但也要确保学生主动去阅读一些特定的操作系统例子。

在1998年,我在春秋两季分别使用UNIX版本和Windows版本来进行操作系统教学。这意味着我需要为两个版本的操作系统提供实验练习和代码示例。简而言之,我需要两本书,一本重点讲解操作系统原理并提供大量UNIX系统的例子和练习,另一本也是类似的,不过是基于Windows系统。无论研究UNIX还是Windows系统,本书都提供了足够的实验练习。本书提供的实验练习有10个能在UNIX环境下实现,有9个能在Windows环境下实现。当选定某一个操作系统时,学生就能认真地学习为该操作系统准备的例子和实验练习,并可以跳过另一个操作系统相关的例子和实验练习。求知欲强的学生可以学习两种操作系统,并能从中受益匪浅。最后,通过把实验练习加入本书,书中的原理与实践很好地结合在了一起,十分有利于增强学生的动手能力。

主题组织

本书的内容安排基于读者对本书第1版和第2版的反应、我的操作系统教学经验,以及从很多其他教师那里得到的信息。本书反映了知识的融合以及众多教师的教学实践,我相信这种合乎逻辑的安排能够被大多数教师所接受。

第1~4章的内容是重要的引论部分,它为操作系统的学习提供了一个坚实的基础。读者可以快速地复习一下这部分内容;也可将它作为课外阅读材料,尤其是如果这部分内容已经在预修课程中学过了。然而,在从第5章开始真正深入学习操作系统之前,理解这部分内容是关键的经验告诉我第2章的内容非常值得阅读。因为在学习操作系统之前,编写过并发程序的学生比较少。第2章的练习可以让学生研究基本的并发概念(尽管他们还不会表示同步问题)。如果你想要练习编写命令解释程序,在第2章中即可看到它的框架,它是作为一个扩展的例子出现的。在第2~9章中,会以实验练习的形式逐渐完善它。

本书从设备管理开始讨论操作系统的细节。尽管这种方法遵循了传统操作系统的演化过程,但你还是会发现这种方法不一般。如在第4章讨论完中断之后,第5章再对设备管理进行讨论比较合理。这种描述方法为引入独立线程(硬件和软件)、并发、同步提供了基础。在读完设备管理这一章之后,你就会对进程、资源管理、调度、同步和死锁这些概念产生深刻印象。

存储管理也是一个很重要的主题,老师们常常绞尽脑汁地想把它表达清楚。本书把它放在进程管理之后、文件系统之前介绍。然后,在学生理解了进程和各种资源(如普通资源、内存以及文件)的概念后,本书对保护和安全这些问题进行了讨论。

任何现代的操作系统都必须能在分布式的环境下运行。分布式操作系统一直影响着操作系统的研究。在所有的传统主题讨论完之后,第15~17章引入了分布式操作系统。由于商业系统和网络的流行,许多老师希望介绍这些内容。然而,在一个学期的课程中,不能花费太多时间来讲解这些内容。

受Java(和.NET以及CLR)成功的驱动,本书引入了分布式程序设计运行时系统这一新的内容。教操作系统课程的老师常常想要谈论类似于Java的技术,或者想要描述用Java编写的操作

系统。不幸的是（至少在我写作本书之时，2001~2003），这些技术并没用于构建操作系统内核，也没用于嵌入商业内核。Sun公司在Solaris平台上实现了Java虚拟机，微软公司在Windows平台上实现了CLR。本书反映了如下事实：操作系统内核技术仍然是用基于C的技术来描述的。然而，类似于Java的技术也很重要，书中也有对它的详细解释。本书为运行时系统，特别是支持并发（分布）程序设计的运行时系统这一部分增加了新的一章，这么做十分合理。随着时间的推移，存在于运行时系统中的一些功能会在内核中出现。

- 第1章展现了操作系统是如何适应日益进步的计算机产品（从大型机到工作站、移动计算机）以及其他软件技术的。在早期的草稿中曾包括了一些历史内容，教师往往喜欢有一点历史和前后关系的内容，但很多学生却很厌烦。所以本书已经将历史内容分散到各个部分中。当讨论特定主题时，操作系统各个部分的历史背景很自然就包括进来了。
- 第2章在概念性操作系统书籍中是独特的，其中考虑了如何使用一个操作系统，尤其是如何写一个使用多线程或进程的程序。增加这一章是因为我的教学经验，对于计算机科学专业大三或大四的学生，他们可能已经编写了相当多的单线程代码，但很可能极少编写过或者学习过多线程软件。如果学生用Java或另一种基于线程的语言进行并发编程的话，这一章的材料仍然是有价值的，因为它有并发C编程的例子和讨论。UNIX命令解释程序和Windows实验练习就是这一章的编程和练习内容。
- 第3章描述了操作系统的基本组织结构，包括实现策略。实验练习提供了初步的UNIX操作系统内部操作。
- 第4章为进一步学习操作系统提供了过渡——计算机组织结构。对于已经修过计算机组织结构这门课的学生来说，第4章的前半部分将是进行回顾，后半部分描述了中断，着重强调了对于操作系统至关重要的方面。
- 第5章描述了设备管理，尤其是一般的技术、缓冲以及设备驱动程序。该章试图完全基于Linux设备驱动程序进行讨论。然而，该章的主要部分却重点介绍中断驱动I/O的宏观目标以及一般组织结构。避免了列举一个实际Linux驱动程序导致的缺乏普遍性的缺点，包括有对设备驱动程序的扩展讨论。实验练习提供了一个很好的用户空间设备驱动例子。学生们喜欢这种尝试，因为他们能直接写一些代码来操纵软盘（即使是通过文件接口）。该章在考虑进程之前分析了设备，因为设备提供了一种基本的物理并发例子，并且必须仔细地设计软件来控制并发运行。这自然也为学习进程管理打下了基础。
- 第6~10章致力于讨论进程管理。从基本的任务、进程的组织结构以及资源管理器（第6章）开始，到调度（第7章）、同步（第8、9章）以及死锁（第10章）。这些章的实验练习扩展了这些主题。
- 第11章讨论存储管理的传统问题，第12章涉及使用虚存的存储管理技术。由于分页技术的普及，我们会对该技术进行更详细的讨论。无论如何，当前流行的存储技术忽视分段技术也许是一个错误。我们将在该章讨论分段技术，但是，最佳可靠段式存储的例子还是Multics系统。第11章的实验练习涉及了UNIX共享内存机制，在第12章，实验练习涉及了Windows存储映射文件。
- 第13章描述了文件管理。对比操作系统类图书的惯用做法，文件管理部分显得有些少，这

是因为它不像进程管理和存储管理那样难以理解。在实验练习中提供了一种详细观察文件管理的手段。该部分讨论在第16章中又有所延伸，其中涉及到远程文件。实验练习可以在UNIX环境下实现，也可以在Windows下实现。做这些练习是有挑战性的，因为工作量非常大（不是因为文件系统的复杂性）。

- 第14章提供了保护机制和安全策略的一般性讨论。虽然其中多数技术恰好与文件、存储器以及其他资源有着密切联系，但它可能被认为应该属于进程管理讨论的范围。在大家熟悉了进程、存储器以及文件管理器后，可能更容易接受保护和安全的需要。
- 第15~17章介绍了支持分布式计算的技术。分布式计算是现代操作系统的一个主要方面，并且我强烈地感到在操作系统的所有介绍中都应该涵盖这个重要的主题。这一章的实验练习是关于TCP/IP协议和远程过程调用的。
- 第18章是全新的一章，它描述了如何构造现代分布式程序设计运行时系统以便扩大内核服务。这是一个令人兴奋的领域，它与操作系统、编程语言、编译器以及分布式编程都有关联。
- 第19章的篇幅很大，它从软件的设计和实现角度重新考虑了所有的操作系统技术。这一章试图描述操作系统设计师必须做的高级设计选择，并指导学生如何使用学过的方法进行成功的操作系统设计。这一章中包含了Mach操作系统的细节性讨论。
- 第20、21章分别是Linux和Windows NT/2000/XP操作系统的实例研究。本书提供了这两种操作系统的许多实现实例，这一章为每一种操作系统提供了统一的描述。

最后，尽管我尽了最大的努力，但要将书中的内容组织成满足每个老师的期望是不可能的。本书反映了我授课时的内容安排，然而，你们也可以根据自己的需要重新组织这些内容。

祝愿读者学习顺利

现在，万维网上有大量的有关操作系统方面的信息，建议学生充分利用万维网。在这一版中，我给出了一些网址，上面有许多相关信息，例如标准化组织的一些材料。

最后，感谢你考虑将本书作为教材。非常欢迎你的提问、评论、批评和建议（我将以积极的方式接受你的批评）。你可以通过Gary.Nutt@colorado.edu和我联系。

致谢

很多人帮助编辑和改进过本书。首先是那些科罗拉多大学的学生们，Jason Casmira、Don Lindsay、Ann Root以及Sam Siewert都是非常好的教学助手，他们设计了上机练习以及解决方案，并且帮助逐渐完善了本书。Scott Brandt对如何组织书中的内容提供了十分有益的见解。Adam Griff花费了许多时间帮助完善对Linux系统的介绍。Scott Morris帮我准备好了Windows NT机器，并且提供了关于它的运行机制的权威提示。

Addison-Wesley从其他学院安排了更多学生检查手稿：蒙大拿州立大学的Eric F. Stuckey、Shawn Lauzon、Dan Dartman和Nick Tkach，以及Berbee信息网络公司的Jeffery Ramin。有很多老师花费了大量时间来查看草稿，或者对本书的组织方式和内容的改进提出建议：Divy Agrawal（加利福尼亚大学圣芭芭拉分校）；Vladimir Akis（加州大学洛杉矶分校）；Kasi Anantha（圣迭戈州立大学）；Charles J. Antonelli（密歇根大学）；Lewis Barnett（里奇蒙大学）；Lubomir

F. Bic (加利福尼亚大学Irvine分校); Paosheng Chang (朗讯科技公司); Randy Chow (佛罗里达大学); Wesley J. Chun, Carolyn J. Crouch (明尼苏达大学Duluth分校); Peter G. Drexel (普利茅斯州立学院); Joseph Faletti, Gary Harkin (蒙大拿州立大学); Sallie Henry博士 (弗吉尼亚理工大学); Mark A. Holliday (西卡罗来纳大学); Marty Humphrey (弗吉尼亚大学); Kevin Jeffay (北卡罗来纳大学Chapel Hill分校); Phil Kearns (威廉与玛丽学院); Qiang Li (圣克拉拉大学); Darrell Long (加州大学圣克鲁兹分校); Junsheng Long, Michael Lutz (罗切斯特理工学院); Carol McNamee (萨克拉门托州立大学); Donald Miller (亚利桑那州立大学); Jim Mooney (西弗吉尼亚大学); Ethan V. Munson (威斯康星大学密尔沃基分校); Deborah Nutt, Douglas Salane (John Jay学院); Henning Schulzrinne (哥伦比亚大学); C. S. (James) Wong (旧金山州立大学); 以及Salih Yurttas (得克萨斯A&M大学)。

最后,在出版这本书的过程中,Addison-Wesley的编辑部以及几个自由作家顾问也付出了无法估量的努力。在第1版中,Christine Kulke、Angela Buenning、Rebecca Johnson、Dusty Bernard、Laura Michaels、Pat Unubun、Dan Joraanstad以及Nate McFadden都提供了宝贵的帮助和指导。第1版的责任编辑Carter Shanklin对如何编写本书有一种先见之明,并且我受惠于他对第1版的内容编排所做出的巨大努力。

第2版的责任编辑Maité Suarez-Rivas促成了在这一版本中包含实验练习,以及将应用的和概念性的内容紧密地集成在一起。Maité和她的助手Lisa Hague、Molly Taylor以及Jason Miranda都不知疲倦地工作以改进第1版,然后在新的版本中提供了特殊的支持。第2版的制作成员,尤其是Karen Wernholm、Amy Rose以及Tracy Treeful,努力工作并将想法转化到完成的作品中。Helen Reebenacker为第2版的实验更新版本处理了很多细节事务。

Rebecca Ferris和Maxine (“Max”) E.Chuck 对本书第3版的文字改进和编辑作出了努力。Juliet Silveri负责本书出版工作的协调及包含新的艺术表现形式。Kathy Smith 处理由于新版式设计及新图示所需的日复一日的拷贝编辑等工作。Holly McLean-Aldis是校样的阅读者,Regina Hagen Kalinda设计了封面和内页,Gillian Hall进行了文字和艺术合成。Maria Campo除了协助Maité外还参与了Juliet小组的工作。感谢他们的长时间工作使得本书得以顺利出版。

本书极大地受益于这些集体成员的努力,当然任何可能存在的错误完全是我的责任。

加里J. 纳特

Contents

Chapter 1

INTRODUCTION 1

1.1 COMPUTERS AND SOFTWARE 3

General System Software 4

Resource Abstraction 7

Example: An Abstraction of a Disk Drive 9

Resource Sharing 11

Abstract Machines and Transparent Resource Sharing 11

Explicit Resource Sharing 16

1.2 OPERATING SYSTEM STRATEGIES 18

Batch Systems 20

The User's Perspective 20

Batch Processing Technology 22

Example: Batch Files 23

Timesharing Systems 23

The User's Perspective 24

Timesharing Technology 24

Example: The UNIX Timesharing System 26

Personal Computers and Workstations 27

The User's Perspective 28

OS Technology 28

Contributions to Modern OS Technology 29

Example: The Microsoft Windows OS Family 29

Embedded Systems 31

The User's Perspective 31

OS Technology 32

Contributions to Modern OS Technology 33

Example: VxWorks 33

Small, Communicating Computers 34

The User's Perspective 34

OS Technology 34

Example: Windows CE (Pocket PC) 36

	Networks	37
	The Genesis of Modern Operating Systems	38
1.3	SUMMARY	38
1.4	EXERCISES	39
Chapter 2	USING THE OPERATING SYSTEM	42
2.1	THE PROGRAMMER'S ABSTRACT MACHINE	42
	Sequential Computation	44
	Multithreaded Computation	46
2.2	RESOURCES	47
	Using Files	48
	<i>Example:</i> POSIX Files	48
	<i>Example:</i> Windows Files	50
	Using Other Resources	52
2.3	PROCESSES AND THREADS	52
	Creating Processes and Threads	55
	FORK(), JOIN(), and QUIT(): The Historical Perspective	55
	<i>Example:</i> Using FORK(), JOIN(), and QUIT()	56
	Classic Process Creation	57
	Creating Modern Processes and Threads	58
2.4	WRITING CONCURRENT PROGRAMS	58
	Multiple Single-Threaded Processes: The UNIX Model	58
	<i>Example:</i> Executing Commands in UNIX	61
	Multiple Processes and Multiple Threads Per Process:	
	The Windows Model	65
	<i>Example:</i> Launching Windows Processes	68
2.5	OBJECTS	72
2.6	SUMMARY	74
2.7	EXERCISES	74
Lab Exercise 2.1	A SIMPLE SHELL	76
	Background	76
	Attacking the Problem	81
Lab Exercise 2.2	A MULTITHREADED APPLICATION	83
	Background	83
	Attacking the Problem	86

Chapter 3 **OPERATING SYSTEM ORGANIZATION** 89

3.1	BASIC FUNCTIONS	89
	Device Management	90
	Process, Thread, and Resource Management	92
	Memory Management	93
	File Management	94
3.2	GENERAL IMPLEMENTATION CONSIDERATIONS	94
	Performance	95
	Exclusive Use of Resources	95
	Processor Modes	96
	Kernels	98
	Requesting Services from the Operating System	99
	Software Modularization	101
3.3	CONTEMPORARY OS KERNELS	103
	UNIX Kernels	104
	<i>Example:</i> Linux	106
	The Windows NT Executive and Kernel	107
3.4	SUMMARY	110
3.5	EXERCISES	110
	Lab Exercise 3.1 OBSERVING OS BEHAVIOR	112
	Background	113
	Attacking the Problem	113

Chapter 4 **COMPUTER ORGANIZATION** 115

4.1	THE VON NEUMANN ARCHITECTURE	115
	Evolving to the von Neumann Architecture	116
	The Basic Idea	116
4.2	THE CENTRAL PROCESSING UNIT	119
	The Arithmetical-Logical Unit	119
	The Control Unit	121
	Implementing the Processor	122
4.3	THE PRIMARY (EXECUTABLE) MEMORY	124
4.4	I/O DEVICES	125
	Device Controllers	126
	Direct Memory Access	128
	Memory-Mapped I/O	129