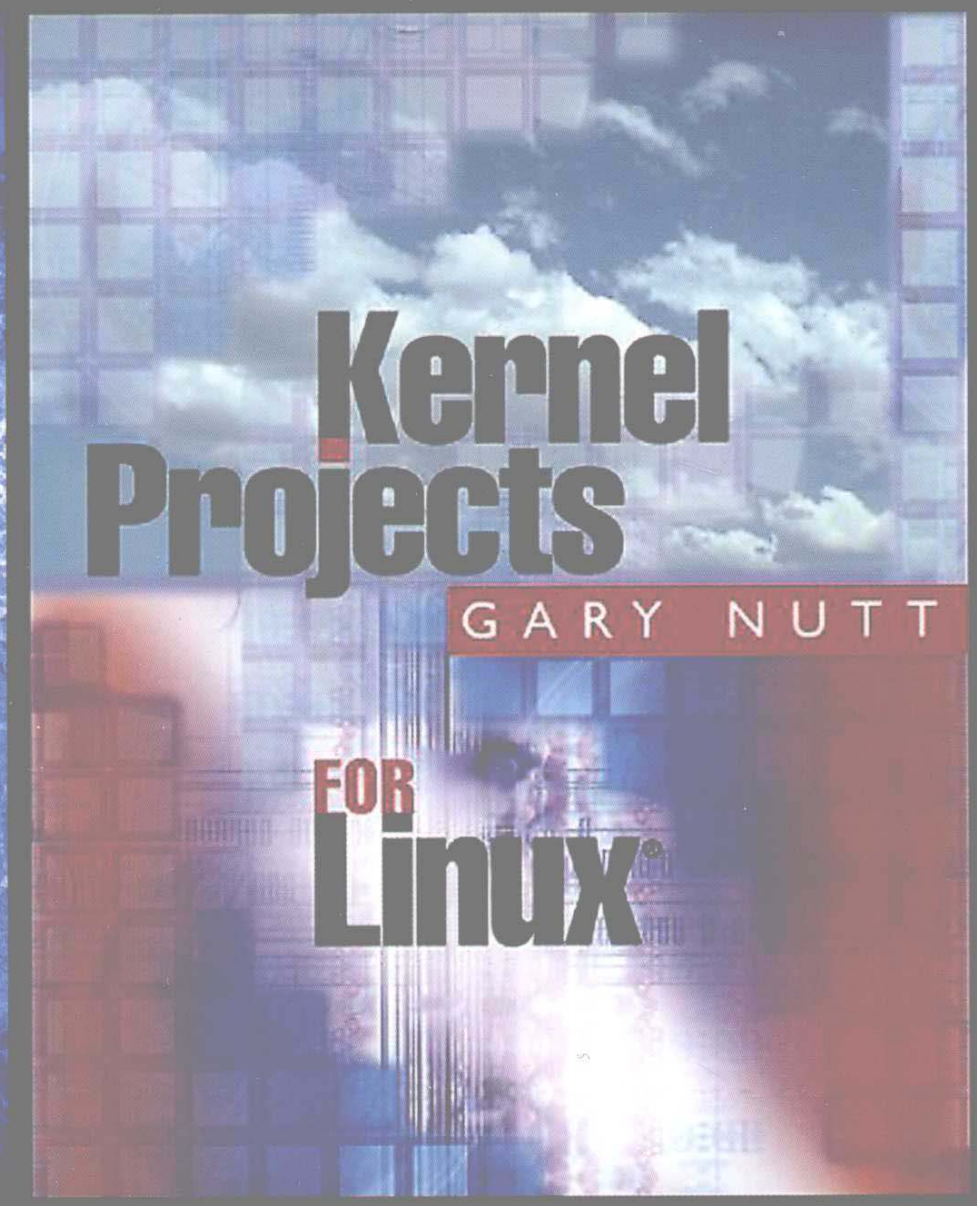
计 算 机 科 学 丛 书

# Linux操作系统内核实习

（美） Gary Nutt 著　　潘登　冯锐　陆丽娜　等译




Kernel Projects for Linux

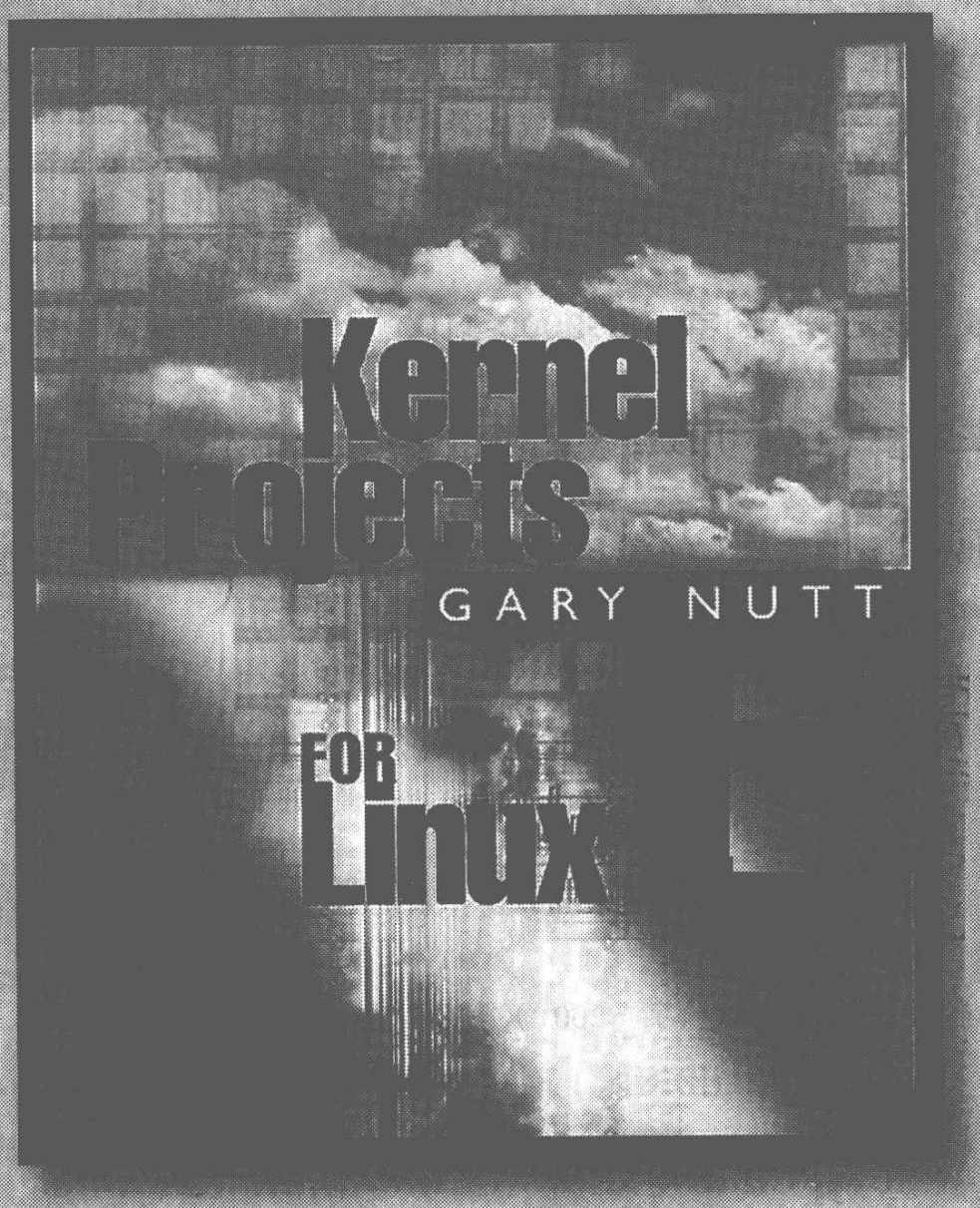
机 械 工 业 出 版 社
China Machine Press

计 算 机 科 学 丛



# Linux操作系统内核实习

（美） Gary Nutt 著　　潘登 冯锐 陆丽娜 等译




# Kernel Projects for Linux

机械工业出版社
China Machine Press

本书是一本传统操作系统教材的配套实验室教材。本书共分两部分：第一部分展示了Linux设计的概况，对Linux环境的运行时组织和进程、文件及设备管理等主题提供了分析；第二部分通过12个练习探讨了操作系统内部结构的各个方面，内容涉及Shell编程、内核模块、系统调用、虚拟存储、文件系统、文件I/O等，从而帮助读者开发自己的Linux内核函数和数据结构，使读者在实验室中真正了解理论概念是如何在Linux 中得到实现的。

本书可供计算机专业本科生使用，也是教师的辅导用书。附带光盘中的Linux源代码为读者的学习和使用提供了便利。



**本书版权登记号：图字：01-2000-4100**

**图书在版编目（CIP）数据**

Linux操作系统内核实习/（美）纳特（Nutt, G.）著；潘登等译. – 北京：机械工业出版社，2002.1

（计算机科学丛书）

书名原文：Kernel projects for Linux

ISBN 7-111-09181-7

Ⅰ. L… Ⅱ. ①纳…②潘… Ⅲ. Linux操作系统 Ⅳ. TP316.81

中国版本图书馆CIP数据核字（2001）第065437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：马 珂**

**北京牛山世兴印刷厂印刷·新华书店北京发行所发行**

**2005年9月第1版第6次印刷**

**787mm×1092mm 1/16 · 11.5印张**

**印数：10 001 – 11 000册**

**定价：29.00元（附光盘）**

# 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、研究及庋藏。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：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。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们服务，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。

“国外经典教材”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本科教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这20

多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书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，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：（010）68995265
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 | 王　珊 | 冯博琴 | 史忠植 | 史美林 |
| 石教英 | 吕　建 | 孙玉芳 | 吴世忠 | 吴时霖 |
| 张立昂 | 李伟琴 | 李师贤 | 李建中 | 杨冬青 |
| 邵维忠 | 陆丽娜 | 陆鑫达 | 陈向群 | 周伯生 |
| 周克定 | 周傲英 | 孟小峰 | 岳丽华 | 范　明 |
| 郑国梁 | 施伯乐 | 钟玉琢 | 唐世渭 | 袁崇义 |
| 高传善 | 梅　宏 | 程　旭 | 程时端 | 谢希仁 |
| 裘宗燕 | 戴　葵 | | | |

# 译 者 序

Linux在近年得到了巨大的发展，它以低廉的价格和稳定的性能在操作系统市场上成为Windows强有力的竞争对手。并且Linux的全部源代码免费公开，爱好者可以按照自己的需要自由修改、复制和发布程序的源代码。Linux的这个特点使得它对于计算机教学和科研具有重要意义。

目前国内的操作系统课程设置过多地偏重于理论学习，对动手实践重视不够或者根本没有涉及。学生普遍反映无法对实际操作系统有一个真正彻底的了解，所学知识停留于表面，不能解决实际问题。美国Colorado大学Boulder分校的Gray Nutt教授编写本书的目的正是为了解决这个问题，书中以Linux内核结构为基础设计了12个互相联系又各有侧重的练习，学生通过完成这些练习可以深入到实际操作系统的内部结构，将理论与实际联系起来。

本书分为两部分。第一部分提供一个Linux的概况，介绍与Linux相关的操作系统概念和理论。第二部分是全书的重点，由12个练习组成，这些练习按难度由浅到深组织，几乎涵盖了Linux内核的各个方面。每个练习由介绍、问题陈述、解决问题和组织方案等几部分组成。通过对这些练习进行学习和动手实践，能够将操作系统的经典理论与其在Linux中的具体实现联系起来，能够更深入地掌握Linux的内核结构，从而达到充分理解操作系统理论知识的目的。

全书由潘登、冯锐翻译，陆丽娜教授审校。由于译者水平有限，难免存在不妥的地方，尤其是一些新的技术术语，在译法上可能与其他文献有所不同，我们力求译文正确反映实际含义又符合中文习惯。若有不当之处，希望广大读者能够提出宝贵意见。

译者
西安交通大学电信学院计算机技术与科学系
2001年6月

# 前　言

## 致学生

经验表明学习一种操作系统（OS）如何工作的最好方式是对它进行实验，去阅读、修改和增强它的代码。但由于其本质，操作系统软件必须细心地构建，因为它直接控制所有运行于其上的进程和线程所使用的硬件。这样，对操作系统代码进行实验可能是十分困难的，操作系统的一个实验版本可能会损坏测试机器。本书向你提供了一个以尽量小的危险研究Linux内核的学习方法。通过这种方法学习Linux内核，你也可以培养一种学习和实验其他内核的技巧。本书将这种学习方法设计为一组分级的练习。首先，学习在不修改任何代码的情况下检查操作系统内部状态的各个方面。第二，通过编写新的代码来阅读（而不是编写）内核数据结构。第三，重新实现现存的数据结构。最后，设计和添加自己的函数和数据结构到Linux内核。

Linux内核是用C程序设计语言编写的。所以在研究内核之前，需要对使用C比较熟悉。如果知道C++，那么在理解源代码上将不会有什么困难，尽管在添加或修改内核的部分代码时将无法使用对象。

本书设计为一本普通操作系统教材的配套教材。它包含两个部分，第一部分提供Linux设计的概况。如果是初学这一门操作系统课程时使用本书，你可能会发现第一部分讨论的几个话题对你是全新的。但是，用它们进行工作会使你知道Linux是如何构建的。这一部分给出了大的总体的描述但没有太多细节，以后在进行练习需要复习时可回到第一部分。

第二部分包括一组帮助你使用Linux的实验室练习。每一个练习都是包含以下部分的自包含单元：

- 介绍
- 问题陈述
- 解决问题

练习将普遍概念与Linux细节联系起来。每个练习都以一个解释Linux概念和练习相关细节的介绍开始。介绍说明了在课堂和书本上学到的一般概念在Linux中是如何实现的。练习的下一部分提出了将要解决的问题，它包括需要用于解决问题的详细的Linux相关信息。有时，在钻研细节之前快速复习一下第一部分中的相关内容，将有助于设计练习的方案。

你所在学校的实验室可能已经建设成为一个Linux实验室。当你在进行本书中的大多数练习时，实验室管理员将向你提供一份Linux源代码的拷贝和超级用户权限，以创建操作系统的新版本。请不要滥用你作为超级用户的权限。你需要这个特权来修改内核，但一定不能把它用于其他目的。本书包括一张光盘，它包含了Linux的源代码，你可以用来在自己的计算机上安装Linux。

祝你的操作系统学习充满好运，我希望本书成为你探索Linux操作系统概念的一个有用工具。

## 致教师

今天，抽象是大多数在教室和实践中书写的软件的基础。教导学生根据对象、组件、线程、消息等等来考虑软件的解决方案，这种观点使得他们借助硬件的力量来解决日益繁杂的任务。通过这种方法，他们在重用低层抽象的同时减少了编程时间。在所有这些抽象的最底层是操作系统——进程和资源（有时是线程）。应用软件和中间件使用这些操作系统抽象来创建它们自己较高层次的抽象，包括从记账程序包、电子表格和导弹跟踪程序到视窗、数据库、对象、组件、消息和连续媒体流。

这种大量使用抽象的趋势使得一些人认为操作系统不再值得仔细研究，因为它们在很大程度上对工作在较高抽象层次上的程序员来说是透明的。但是，操作系统仍是十分基础的，因为它的设计和实现是所有其他抽象设计和实现的基础。程序员如果理解操作系统是如何工作的，他们就能够编写出更好的中间件和应用程序。此外，仍然需要理解基本操作系统技术的人员，无论他们是为新设备编写驱动程序、创建新的微内核服务器，还是提供能够高效处理发展需求（如连续媒体）的新系统。

通常情况下，一位操作系统教师必须决定操作系统课程是应集中于问题和理论，还是为学生提供一个能够对操作系统代码进行实验的环境。1991年（和2001年草案）的IEEE/ACM本科课程建议描述了一门花费大量时间在各种问题上的课程，但同时也包括了一个重要的实验室部分。虽然这个趋势是以概念性材料作为课程的基础，但学生和教师似乎都同意实际经验在学习操作系统过程中是极为重要的。许多课程试图遵照IEEE/ACM的指示，将一门课程划分为课堂和实验室两个部分，课堂部分集中于问题和理论，而实验室部分提供一些实际的动手练习。

IEEE/ACM建议支持实验室部分应使得学生学会如何使用操作系统机制的观点，尤其是学会将操作系统应用程序编程接口（API）作为主要的实验机制。这种方法背后的基本原理是，在学生能够真正理解如何设计一个操作系统之前，他们必须学习如何使用一种操作系统。这一原理促成了一本有关通过Win32 API进行Windows NT编程[Nutt，1999]以及一本有关实验室练习[Nutt，2000]的参考书的产生。

但是，后来在1998年由Addison Wesley对78所大学所进行的调查中，43所表明他们在介绍性的操作系统课程中讲授了操作系统内部结构。在这43所大学中，26所使用了UNIX的一个版本作为目标操作系统，其中13所使用了Linux，10所使用了未指明版本的UNIX，3所使用了MINIX。8所大学说他们使用了其他的操作系统作为主题系统（例如Nachos），剩下的9所没有指明他们所使用的操作系统。调查清晰地显示，尽管有IEEE/ACM建议的存在以及概念性操作系统教材的大量使用，教育界仍有很大一部分将操作系统内部结构作为介绍性操作系统课程内容的一部分讲授。它还显示，大多数这些课程使用两种教材：一本传统的操作系统理论教材（例如，[Siberschatz and Galvin，1998]或者[Nutt，2000]）和一本参考教材（例如，[Stevens，1993]、[McKusick，et al.，1996]，或者[Beck，et al.，1998]）。当然，一个学期的本科课程是不可能覆盖一本理论教材与一本描述整个操作系统的教材的所有内容。缺乏一本好的实验室手册迫使教师让学生购买一本补充教材，而它所包含的内容比学生在一个学期内有时间学习的内容要多得多。另外，教师也不得不学习两本教材中的所有内容，学习主题操作系统，设计一组合适的练习，并且通过操作系统参考资料提供某种形式的指导，以便学生

能够完成练习。

这本教材是一本Linux内部结构练习的实验室手册。它通过提供一组与Linux内部结构有关的特定练习作为对一本操作系统理论教材的补充，这些练习解释了理论概念是如何在Linux中得到实现的。教师不需要变成一位精通Linux内核的专家或者自己设计一组练习（无论提供对实验的全部文档或者只是提供对补充参考教材中适当部分的提示）。相反，教师、实验室助理和学生可以把本书作为一个背景数据和练习的自包含源来使用以研究概念如何实现。所以，这本不铺张的实验室手册取代了一本普通的参考教材，同时提供了一组内核内部结构练习的集中信息。对于想了解超出练习要求之外相关信息的学生，练习的背景内容提供了有关参考书籍（和文献）方面的指示。

一学期的操作系统课程包含15周的课程内容。根据我的经验，大多本科学生在少于一周半到两周的时间里做大量编程练习是有困难的。这意味着在一学期中学生能完成大约6~8个编程作业。本书提供了足够的练习，你可以从中选取最适合学生背景和你喜好的一部分。大部分练习都包括用以在各个学期之间调整的部分（以此减小使用前面几个学期公共解答的可能性）。如前所述，我的意图是不断发行本书的更新版本。我希望新的版本将会有需要新解答的新练习。

同时提供的还有对各个练习的一个解答。因此可以选择更难的练习，而且在必要时可以分发未在本书中公布的部分解答。

这些练习中没有一个与从头建立一个新的内核一样困难。相反，它们强调让学生通过修改和扩充Linux内核的部件来对它进行研究。第一部分比较简单，而且背景材料较为全面。后面的练习在难度上有所增加，同时减少了提示信息的数量。练习一和二通常需要一周或更少的时间来完成，但是最后三个练习可能每个需要两周的时间。如果你的学生在C编程方面需要额外的实践，你可以谨慎地考虑使用练习一和二作为指导。这可能需要你提供一些额外的提示，尤其是练习二的补充材料。

### Linux的CD-ROM版

对任何一个操作系统的实际动手研究都必须针对一个特定版本的操作系统。随着Linux的迅速发展，到本书出版之际版本2.2.x将会被淘汰。为了避免本书与操作系统代码不同步的问题，我已经采用了版本2.2.14的源代码。本书的第1版最先是为版本2.0.36所编写的。随后，在它进入发行周期的前夕，针对版本2.2.12的Linux进行了更新。当本书即将被印刷时，我发现只有版本2.2.14（而不是2.2.12）可以随本书分发。在版本2.2.12和2.2.14之间存在细小的差别——通常在于编码方式而非内容。但是，这些差别会表现在一些练习中，尤其要在虚拟存储和调度程序的内核代码部分注意它们。在下一版中我将会改正这些错误。我认为包含一张版本2.2.14的安装盘要比不带任何光盘的手册更好。虽然当在使用本教材时可以得到更新版本的源代码，但是我建议你在实验室的机器上安装这个版本，以便你的学生有一个与本书一致的合理的软件环境。我最大的希望是能够大致跟随Linux的发行而不断发行本书的新版本，下一版可能会使用例如版本2.6.x。

## 致谢

本书是我和其他人多年学习Linux的结果。我从科罗拉多大学操作系统课程的助教们的帮

助、见解和奉献中获益匪浅，他们是：Don Lindsay、Sam Siewert、Ann Root和Jason Casmira。Phil Levis提供了有关Linux和练习的生动有趣的讨论。当我第一次在一台机器上安装Linux时，它虽然能够工作但不如Adam Griff调试安装后运行得那样畅快。

许多练习都来自于科罗拉多大学本科和研究生操作系统课程的实习和练习。尤其是练习三由Sam Siewert在1996年春季为计算机科学课程而设计。练习四从Sam Siewert设计的另一个练习中取得了一些材料。练习九来自Jason Casmira在1998年秋季为研究生操作系统课程所做的一个课程实习。练习十首先由Don Lindsay在1995年秋季设计，后来由Sam Siewert在1996年春季修改。练习一也出现在我的配套操作系统课本[Nutt，2000]中，练习二是出现在那本书中的另外一个练习的延伸。练习十一和十二与出现在我的另一本手册中的Windows NT练习类似，而Norman Ramsey设计了最初的Windows NT练习。

许多评论家对原始书稿进行了审查，使得本书更加完美。Richard Guy在UCLA的一门课程中使用了本手稿的第一次公开书稿。Paul Stelling (UCLA)仔细审阅了书稿，纠正了其中的错误并对它好的和不好的方面提出了见解。Simon Gray（Ashland大学）对练习提出了非常明晰和深刻的意见。以下各位也提出了有帮助的意见，使得本书质量得到了极大的提高：John Barr（Ithaca学院）、David Binger（中心学院）、David E. Boddy（奥克兰大学）、Richard Chapman（Auburn大学）、Sorin Draghici（Wayne州立大学）、Sandeep Gupta（科罗拉多州立大学）、Mark Holliday（西卡罗莱纳大学）、Kevin Jeffy（北卡罗莱纳大学Chapel Hill分校）、Joseph J. Pfeiffer（新墨西哥州立大学）、Kenneth A.Reek（Rochester技术研究院）和Henning Schulzrinne（哥伦比亚大学）。

Addison Wesley公司在准备本书过程中提供了极大的帮助。Molly Taylor和Jason Miranda在处理评论和前期开发其他支持方面提供了广泛的援助。Lisa Hogue用了一天的时间来找到可以随书分发的Linux源代码版本。Laura Michaels在文字编辑过程中付出了她一贯辛勤的劳动，Gina Hagen在生产方面提供了帮助，Helen Reebenacker是生产编辑。最后，但并非最次要，选书编辑Maite Suarez Rivas认识到对本书的需求并大力促成了本书的出版。

所有这些人们的帮助促成了本书的诞生，当然，任何错误都应该归咎于我的责任。

Gary Nutt

于Boulder，科罗拉多

# 目　录

# 第一部分 Linux概况

Linux是一个当代开放的UNIX实现，可以在因特网上无偿得到。自从它1991年出现以来，已经成为一个高度重视的健壮的操作系统（OS）实现。用它作为一个平台来学习现代操作系统已经获得了巨大的成功，尤其是学习其内核的内部行为。同样重要的是，Linux现在被用于许多公司的信息处理系统。所以，学习Linux内部结构也就获得了重要的职业训练，同时它也是教学实验的一个好方法。今天，一些新的公司开始对Linux提供产业强度的支持。

## 1 Linux的演变

Linux起源于UNIX学术团体。在1973年的一篇经典研究论文[Ritchie and Thompson, 1974]中，UNIX操作系统首次被公诸于众。UNIX在操作系统设计上确立了两个新的趋势。第一，以前的操作系统是大型的软件包，通常也是运行在计算机上的最大软件包，而且它们为一种专门的硬件平台而设计。与之相对应，UNIX被设计为一个小型的、无虚设的操作系统，可以运行在任何小型计算机上。第二，UNIX的原则是操作系统内核应该提供最少量的基本功能，而其他功能应该在需要的基础上添加（作为用户程序）。由于这些原因，UNIX是革命性的。仅在6年之中，它已成为多厂商硬件环境（大学、研究实验室和系统软件开发组织）下程序员们所偏爱的操作系统。

虽然UNIX内核可以被引入一个新的硬件平台而不需要重新开发整个操作系统，但源代码由AT&T贝尔实验室所拥有。通过向AT&T交纳许可费，其他组织可以得到使用源代码的权利（例如，将它引入到它们所喜欢的计算机硬件上）。但是，到1980年，许多大学和研究实验室已经得到源代码，并根据它们自己的需要对其进行修改。当时最突出的工作是加州大学伯克利分校所承担的名为国防部高级研究计划署的研究合约。商业计算机厂商也开始使用UNIX源代码来衍生他们自己的UNIX操作系统版本。

到1985年，UNIX的两种主要版本运行于许多不同的硬件平台上：来自AT&T贝尔实验室的主线版本（称为*System V UNIX*）和另一个来自加州大学伯克利分校的版本（称为*BSD UNIX*）。BSD UNIX针对的主要硬件平台是DEC VAX，所以BSD UNIX也更特定地称为版本4 BSD UNIX，或者4.x BSD UNIX。虽然AT&T和BSD两个版本都实现了可被识别为UNIX的系统调用接口，但它们在一些细节方面各不相同,尤其是在两种操作系统内核的实现方式上存在本质上的差别。System V和BSD UNIX系统之间的竞争非常激烈，程序员们都效忠于一个版本或另一个。到了1988年，AT&T与4.x BSD UNIX的主要倡导者（Sun微计算机公司）达成了一个商业协议，通过这个协议两个主要的版本将会融合成为一个公共的UNIX版本，称为Sun Solaris OS。

当时，其他计算机制造商正在力推另一种UNIX系统调用接口的实现。因此到了1989年，

成立了一个委员会来开发标准接口，结果产生了一个名为*POSIX.1*[⊖]的API，它是一种系统调用接口。使用POSIX，每个人都可以自由设计和建立能够提供它所指定功能的内核。例如，卡内基梅隆大学由Richard Rashid领导的一组操作系统研究人员开发了具有POSIX/UNIX系统调用接口的*Mach*操作系统。Mach是与4.x BSD和System V UNIX不同的另一种内核实现。最终，Mach的一个版本被用作开放系统基金会OSF-1内核的基础。最初用于实现Mach中POSIX/UNIX接口的技术只是简单地将大量BSD UNIX代码加入到Mach内核中。但是到了版本3，Mach被重新设计为一个具有服务器的微内核。虽然该版本微内核不包含许可的源代码，BSD服务器仍然通过使用BSD源代码[Tanebaum，1995]来实现4.x BSD系统调用接口。

在一个较小的规模上，Andrew Tanenbaum在1987年设计并实现了一个名为MINIX的UNIX完整版本。“MINIX代表mini-UNIX，因为它是如此之小甚至初学者也能够理解它是如何工作的”[Tanebaum，1987]。MINIX实现了曾经一度流行的名为*Version 7 UNIX*的AT&T UNIX系统调用接口，它是BSD和System V UNIX的基础。Tanebaum将MINIX作为补充材料随他的操作系统教材分发，同时提供了一个关于内核如何设计和实现的详细讨论。他指出MINIX“……在UNIX 10年之后写成，并采用一种更加模块化的方式来构建”[Tanebaum，1987]。也就是说，它基于一个具有服务器的微内核，与BSD和System V UNIX的单体（monolithic）设计相对。MINIX最初作为一个教学操作系统相当成功，但最终由于它的主要特点——其简单性——而失去了一些支持。它对学习来说已足够小，但用作一个实际操作系统则不够健壮。

1991年，Linus Torvalds开始创建 Linux 版本1。[⊜] 尽管受MINIX的成功所激励，但是他打算让他的操作系统比MINIX更加健壮和实用。Torvalds对所有人公布了他的源代码（通过因特网使它在GNU公共许可下免费获得）。Linux作为一个重要的POSIX实现迅速流传开来，它是第一个源代码完全免费的UNIX版本。不久，世界各地的人们开始对Torvalds最初的代码进行修改和增强。今天，Linux的发行版本不仅包括操作系统，还包括由许多不同的贡献者所编写的补充工具。除了针对最初的Intel 80386/80486/80586（也被称为x86或i386）的实现之外，针对Digital Alpha、Sun Sparc、Motorola 68K、MIPS和PowerPC的实现也都被创建。到1996年，Linux已经成为了一个重要的操作系统。一年之后，它成为商业操作系统市场中的一个重要部分，并且继续充当免费UNIX接口实现的首要角色。

根据UNIX的原则，Linux实际上是一个操作系统的核心或者内核而非完整的操作系统。UNIX内核被作为一个最小的操作系统引入，执行直接必要的功能，而让软件为程序库和其他用户态软件提供所需的工具。早期的UNIX内核遵循“小就是美”的原则，但在其后的25年当中，新的特征不断被添加到其中，因此而带来的“蔓延特性”使它变得相当庞大。今天，UNIX再也不能被认为是操作系统功能的最小集合。

虽然Linux通常被认为是“UNIX内核”，但在1991年它启用了一种新的设计，不包含当时内核中存在的方法和补丁集合。在实现POSIX的同时，它试图遵循早期UNIX“小就是美”的

⊖ 系统调用接口通常被简单地称为“POSIX”，但这个术语可能会引起误会，因为POSIX委员会开发了几种不同的API而只有第一个标准能够处理内核系统调用接口。本书只考虑POSIX.1，因此采用更为流行但却不太精确的POSIX来指代POSIX.1系统调用接口。

⊜ comp.os.minix新闻组有一些关于Torvalds的在1991年早期的记录，其中他透露了他正忙于一个公开的POSIX实现。

原则。但是，因为它免费可得，任何人都可以向内核中增加特征，因此它也可能存在使得UNIX内核增长的相同的“蔓延特性”问题。Linux内核仍然相对较小但增长迅速。Linux版本2.2内核比版本2.0更加庞大和复杂，而版本2.0比版本1.0庞杂。

本书讨论了版本2.2 Linux内核的设计与实现。练习和解释是基于在2000年早期发行的最新和稳定的版本2.2.12的Linux。本书的第一部分描述了内核的组织结构。第二部分中的练习集中于内核的不同方面，每个练习都提供了大量与该练习相关的内核细节。