

中国计算机学会教育专业委员会 推荐
全国高等学校计算机教育研究会 出版

高等学校规划教材
国家精品课程教材



Linux操作系统 实验教程

罗 宇 陈燕晖 文艳军 等编著

计 算 机 学 科 教 学 计 划



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校规划教材

国家精品课程教材

Linux 操作系统实验教程

罗宇 陈燕晖 文艳军 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是国防科技大学国家精品课程“操作系统”配套教材，系统地讲解了 Linux 操作系统原理和基于 Linux 的各种编程，特别是 Linux 操作系统内核编程。本书内容分为三部分：第一部分介绍 Linux 操作系统原理；第二部分介绍 10 个基于 Linux 的实验；第三部分附录包含 Linux 环境下的操作及与编程有关的命令和函数列表。

本书适合作为高等学校计算机科学与技术、软件工程、信息安全、信息与计算科学、信息管理与信息系统等专业操作系统实验和课程设计教材，也是 Linux 开发人员熟悉 Linux 环境下应用及内核编程的入门参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

Linux 操作系统实验教程 / 罗宇等编著. —北京：电子工业出版社，2009.2

高等学校规划教材

ISBN 978-7-121-08217-7

I. L… II. 罗… III. Linux 操作系统—高等学校—教材 IV.TP316.89

中国版本图书馆 CIP 数据核字（2009）第 013267 号

策划编辑：童占梅

责任编辑：童占梅

印 刷：北京市海淀区四季青印刷厂

装 订：涿州市桃园装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1 092 1/16 印张：11.25 字数：276 千字

印 次：2009 年 2 月第 1 次印刷

印 数：4 050 册 定价：19.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前 言

操作系统是计算机系统必不可少的关键软件。操作系统教学不但需要讲授操作系统概念、原理与方法,还要让学生动手在实用操作系统上进行编程实践,只有这样才能让学生真正理解操作系统的概念、原理与方法。编写本书的目的就是为了在学习完操作系统原理后,为操作系统实践教学提供指导。

Linux 操作系统是源代码公开的广泛使用的操作系统,利用 Linux 作为操作系统课程的实验平台,不但能帮助学生进行以理解操作系统原理为目的的实验,同时也可以看作是操作系统开发实战的演练。由于 Linux 源代码公开以及 Linux 的广泛普及,我们认为将 Linux 作为操作系统实验教学平台还将为学生毕业后快速进入实战工作状态打下良好的基础。

本书是国防科技大学**国家精品课程“操作系统”建设的成果之一**,之前出版的**《操作系统(第2版)》**和本书**组成了完整的理论和实践教学体系**,并在教学中取得了很好的教学效果。**欢迎登录“操作系统”国家精品课程网站 <http://618.net.cn>。**

本书分为三部分。第一部分介绍 Linux 操作系统原理;第二部分介绍 10 个基于 Linux 的实验;第三部分附录包含 Linux 操作及编程所需的参考资料。本书可作为 CCC2002《操作系统实验和课程设计》专用指导教材,其中部分内容也可以作为操作系统原理课程的课后实验。本书还是 Linux 技术人员的系统编程入门参考书。

本书作者长期从事计算机操作系统研究开发及教学工作,根据多年操作系统开发及国家精品课程建设的教学经验,参考国内外近年出版的各类操作系统实验教程特色,设计了一组基于 Linux 环境的操作系统课程实验,它包括三种类型:系统管理用 shell 脚本编程及用户态运行应用程序实验;内核入门实验;内核综合实验。用户态编程实验主要是让学生体会操作系统功能及接口,在内核实验中,有内核模块实现这样的内核入门实验,也有文件系统、USB 盘驱动这样的实用综合实验,目的是让学生由浅入深地实际体验 Linux 操作系统的系统能力及操作系统设计原理。

本书提供了 Linux 操作系统基本实现原理,实验内容、实验背景知识、解决方案的描述,附录给出了应用和内核编程可能涉及的系统调用、多线程库函数等,读者可以在无须其他参考书的情况下实现基本的实验编程。在操作系统实验课程教学实践中,不需局限于本书所列的实验,建议任课老师将最终实验题(如动态加载模块方式的某驱动程序实现或由学生自选的某个实验)在第一次课时就布置给学生,让学生自己理解实现目标和实验所要掌握哪些基本实验,遇到问题,鼓励学生通过互联网探索解决问题的方法,这样可以发挥学生的主观能动性,以收到更好的实验效果。建议第 8 章实验作为操作系统原理课程的课后实验,第 10 章的实验作为独立操作系统实验课的综合实验由学生任选,而第 9 章实验由学生根据所选综合实验需要自行实验。整个实验教学建议在实验机房进行,**本书所有实验在 Linux 发行版 Fedora 7 及内核版本 2.6.21.7 下经过了验证。**

本书由罗宇负责组织和统稿,罗宇、陈燕晖编写了 Linux 操作系统原理部分,文艳军编写了实验 10,其他部分均由陈燕晖编写。个别实验参考了罗宇、储瑞编著、机械工业出版社出版的《操作系统课程设计》。参加本书编写工作的还有李冬、晏益慧。由于编者水平有限,错误在所难免,有对本书的任何批评和建议可发送邮件到 yuluo@nudt.edu.cn。

编著者

于湖南长沙国防科技大学计算机学院

目 录

第一部分 Linux 操作系统基本原理

第 1 章 Linux 操作系统简介	3
1.1 Linux 的渊源和发展简史	3
1.2 Linux 的基本特性	4
1.3 Linux 内核的开发模式与内核版本号	5
1.4 Linux 发行版介绍	6
1.5 Linux 内核源代码组织结构	7
1.6 学习 Linux 的辅助软件介绍	8
第 2 章 Linux 的进程管理	10
2.1 进程与进程描述符	10
2.2 进程状态及切换时机	11
2.2.1 Linux 的进程状态	11
2.2.2 进程的切换时机	12
2.3 进程的调度算法	13
2.4 进程的创建与消亡	14
第 3 章 Linux 的存储器管理	17
3.1 物理内存的管理	17
3.1.1 页帧与区域	17
3.1.2 伙伴算法	17
3.1.3 slab 分配器	18
3.2 进程地址空间的管理	19
3.2.1 页表机制	20
3.2.2 vm_area_struct 结构	21
3.2.3 进程地址空间的相关系统调用	22
3.2.4 页面异常的处理	22
第 4 章 Linux 的文件系统	24
4.1 VFS	24
4.1.1 VFS 的作用	24
4.1.2 进程描述符中与文件系统相关的成员	24
4.1.3 VFS 的文件模型	25
4.1.4 文件系统的注册与安装	27
4.1.5 各种对象的操作接口	28
4.2 EXT2 文件系统	30
4.2.1 EXT2 在磁盘上的物理布局	30

4.2.2	主要的数据结构及其基本操作	31
4.2.3	磁盘块的分配与释放	34
4.3	主要文件系统的系统调用处理流程	34
4.3.1	文件的 open 操作	34
4.3.2	文件的 read 操作	35
第 5 章	Linux 的设备管理	37
5.1	设备文件的概念	37
5.2	设备模型基础	37
5.3	相关数据结构	38
5.3.1	字符设备管理	38
5.3.2	块设备管理	39
5.3.3	buffer	40
5.3.4	设备请求队列和 I/O 调度算法	41
5.4	块设备文件的 open 和 read 操作	41
5.4.1	块设备驱动程序组成	41
5.4.2	open 函数	41
5.4.3	read 函数	42
第 6 章	中断、异常及系统调用	43
6.1	中断和异常的基本知识	43
6.2	异常处理函数	43
6.3	系统调用	44
6.4	中断的处理	45
6.4.1	中断控制器	45
6.4.2	管理中断的数据结构	45
6.4.3	中断的处理过程	47
6.5	软中断	47
第 7 章	Sys V 进程间通信	49
7.1	共有的特性	49
7.2	信号量	50
7.3	消息队列	52
7.4	共享内存	54

第二部分 基于 Linux 操作系统的实验

第 8 章	用户态编程实验	59
8.1	实验 1——bash 脚本编程	59
8.1.1	实验内容	59
8.1.2	bash 脚本编程简介	59
8.1.2.1	注释和简单命令	59
8.1.2.2	环境变量	60

8.1.2.3	控制结构	60
8.1.2.4	函数	64
8.1.3	实验指南	64
8.2	实验 2——观察 Linux 行为	65
8.2.1	实验内容	65
8.2.2	proc 文件系统简介	65
8.2.3	实验指南	69
8.2.3.1	Linux 环境下 C 语言编程环境简介	69
8.2.3.2	实验程序框架	69
8.3	实验 3——实现 Linux 命令解释器	70
8.3.1	实验内容	70
8.3.2	myshell 的语法	70
8.3.3	myshell 的程序框架	71
8.3.4	myshell 命令行的语法分析	72
8.3.5	简单命令的执行	74
8.3.6	myshell 的 Makefile	74
8.3.7	实验指南	75
第 9 章	内核编程基础实验	76
9.1	实验 4——内核模块	76
9.1.1	实验内容	76
9.1.2	Linux 内核模块简介	76
9.1.3	内核符号表	76
9.1.4	内核模块编程介绍	77
9.1.4.1	内核模块实例	77
9.1.4.2	模块编程的基本知识	78
9.1.4.3	Makefile 介绍	79
9.1.5	实验指南	80
9.1.6	测试	82
9.2	实验 5——proc 文件系统编程	83
9.2.1	实验内容	83
9.2.2	proc 文件系统编程简介	83
9.2.2.1	proc 文件系统编程示例	83
9.2.2.2	proc 文件系统的核心数据结构	85
9.2.2.3	proc 文件系统编程接口	86
9.2.3	实验指南	88
9.3	实验 6——编译内核及增加 Linux 系统调用	88
9.3.1	实验内容	89
9.3.2	Fedora 下编译内核	89
9.3.2.1	第 1 步——下载内核	90

9.3.2.2	第 2 步——生成内核配置文件.config	90
9.3.2.3	第 3 步——编译和安装新的内核	91
9.3.3	添加 psta 系统调用	92
9.3.4	测试新增系统调用 psta	94
9.3.5	noexec 系统调用的实现	95
第 10 章	内核编程综合实验	96
10.1	实验 7——进程隐藏	96
10.1.1	实验内容	96
10.1.2	背景知识介绍	96
10.1.3	proc 文件系统实现简介	98
10.1.4	实验指南	102
10.1.4.1	功能(1)的实现	102
10.1.4.2	功能(5)的实现	103
10.1.4.3	功能(7)的实现	106
10.2	实验 8——字符设备驱动开发	106
10.2.1	实验内容	107
10.2.2	字符设备驱动开发介绍	107
10.2.2.1	测试字符设备	109
10.2.2.2	描述设备的数据结构	110
10.2.2.3	设备号的操作	110
10.2.2.4	字符设备的注册与注销	111
10.2.2.5	文件操作集	111
10.2.2.6	同步	112
10.2.3	字符设备 chatdev 的实现	113
10.2.4	聊天程序 chat 的实现	114
10.3	实验 9——naive 文件系统的设计与实现	115
10.3.1	实验内容	115
10.3.2	项目的准备工作及建议	115
10.3.3	实验指南	116
10.3.3.1	第 1 步——创建设备	116
10.3.3.2	第 2 步——格式化分区	116
10.3.3.3	第 3 步——定义并注册 naive 文件系统	117
10.3.3.4	第 4 步——安装/卸载文件系统分区	118
10.3.3.5	第 5 步——显示根目录的内容	121
10.3.3.6	第 6 步——在根目录下创建内容为空的文件	124
10.3.3.7	第 7 步——写文件和读文件	126
10.3.3.8	第 8 步——删除文件	127
10.3.3.9	第 9 步——创建目录	128
10.3.3.10	第 10 步——删除目录	129

10.4 实验 10——块设备驱动开发	130
10.4.1 实验内容	130
10.4.2 实验基础和思路	130
10.4.2.1 参考驱动程序 1——块设备驱动程序 sbull	130
10.4.2.2 参考驱动程序 2——USB 字符设备驱动程序 usb-skeleton	132
10.4.3 U 盘驱动的帮助函数	133
10.4.3.1 函数原型及其使用	133
10.4.3.2 工作原理和过程	136
10.4.4 实验指南	138

第三部分 Linux 环境下的操作及常用命令和函数

附录 A Linux 常用命令	143
A.1 用户终端命令	143
A.2 vi 编辑器的用法	151
附录 B Linux 常用函数	154
B.1 进程管理函数	154
B.2 文件管理函数	156
B.3 进程间通信	158
B.4 多线程库	161
附录 C 内核配置文件的生成	164
C.1 配置文件初步生成	164
C.2 修改内核配置文件	165
C.3 内核编译选项介绍	166
参考文献	167

第一部分

Linux 操作系统基本原理

Part one

第一部分共分 7 章，简要介绍 Linux 操作系统原理。

第 1 章介绍 Linux 的发展历史和开发模式，以及内核代码的组织结构。

第 2 章介绍 Linux 的进程管理机制。Linux 特殊的 clone 系统调用导致它支持线程的方式与操作系统原理介绍的支持线程的方式有着明显的区别，读者需要仔细体会。此外，Linux 调度器采用的算法实际上是多种算法的糅合，请注意与本科教材介绍的算法进行比较。

第 3 章介绍 Linux 的存储器管理。作为支持虚拟存储器的系统，先介绍分页存储采用的伙伴算法，然后介绍内核各子系统广泛用到的 slab 分配器，该分配器的粒度细且效率高。最后介绍进程地址空间。读者应熟悉支撑“分页技术”的核心数据结构，了解相应的硬件支持，而为了提高性能所采用的“写时复制”则是非常实用的技术。

第 4 章介绍 Linux 的文件系统。文件系统为用户提供了抽象易用的接口，如打开、读/写和关闭等，读者应该抓住“这些操作如何映射到具体的物理文件系统”这条主线来贯穿各个知识点。Linux 为了支持多种不同的文件系统，引入了中间层 VFS，VFS 实际上采用了面向对象的设计方法。本章随后介绍了物理文件系统 EXT2 的物理存储布局 and 核心数据结构，最后讲述几个系统调用的实现。文件系统的性能对系统影响极大，内核设计中对多种对象都采用了缓冲技术，请读者仔细体会。

第 5 章介绍设备管理。由于设备的多样性，本章只介绍一些最基本的概念和核心数据结构。更详细的一些细节则放到了第二部分的实验环节中讲述。

第 6 章介绍中断、异常和系统调用。其中相当一部分知识的介绍依托于 i386 平台，读者在阅读过程中应该注意区分。如果可能，了解另外一种体系结构的相关实现则更有裨益。

第 7 章介绍 Sys V 的进程间通信在 Linux 上的实现。

从整体上来说，Linux 操作系统的实现是操作系统原理教材中所讲述的基本概念、原理及算法的运用，但实际情况是，真正的实现往往比一般原理要复杂得多。内核黑客 Jens Axboe 谈及 I/O 调度算法时说过 “The classic IO scheduling algorithms you find in OS text books aren’t really useful in a modern system”，这句话不仅仅适用于 I/O 调度算法。所以学习 Linux 内核原理可看成是进一步的深化学习，如果同时参看源代码，又辅以动手环节，无疑是更加有益的。

第 1 章 Linux 操作系统简介

Linux 的发展史极富传奇性。本章首先介绍 Linux 的渊源和发展史，接着介绍 Linux 的特点和开发模式，随后介绍内核代码的组织结构和 Linux 的学习方法。

1.1 Linux 的渊源和发展简史

1965 年，美国麻省理工学院、通用电气公司及贝尔实验室联手开发在当时看起来非常先进的 MULTICS 操作系统。经过数年开发后，贝尔实验室认为该项目没有成功的希望便撤出了自己的开发人员，这些人中便有 Ken Thompson 和 Dennis Ritchie。随后，Ken Thompson 在实验室闲置的 PDP-7 机器上开发出了 UNIX 操作系统。最早的 UNIX 是用汇编语言开发的，20 世纪 70 年代初期，Dennis Ritchie 发明了 C 语言并用 C 语言重写了 UNIX 操作系统，直到如今 C 语言依然是系统编程语言的首选。

1974 年，Ritchie 和 Thompson 在 Communications of the ACM 发表了一篇经典论文“The UNIX Time-Sharing System”，使 UNIX 得到广泛关注。1975 年，美国贝尔实验室发布了 UNIX 第六版（V6），当时一些大学及研究机构得到该系统及其源代码，并用于研究和教学。美国加州大学伯克利分校就是依据这个版本开始研究并加以发展的，并在 1977 年发布 1 BSD（1st Berkeley Software Distribution），BSD UNIX 后来成为 UNIX 家族最重要的分支之一。

20 世纪 80 年代初期，UNIX 开始商业化，AT&T 发布了 System III 和 System V 等。一些商业公司也陆续跟进，发布自己的商用 UNIX 版本，如 IBM 公司的 AIX、Sun 公司的 SunOS（后来命名为 Solaris）等，这些商业版本的 UNIX 源代码不再开放。商业公司的利益角逐导致 1984 年之前 UNIX 蓬勃发展的景象不再复现。

一些有识之士认为，计算机软件的商业化将有碍知识的传播和人类的进步，妨碍用户自由，其中尤以 Richard Stallman 表现最为激烈。Richard Stallman 于 1984 年着手 GNU 计划，计划开发一套与 UNIX 相互兼容且自由开放的软件。1985 年，Richard Stallman 又创立了 FSF（Free Software Foundation）为 GNU 计划提供各方面支持。Richard Stallman 提倡软件自由，为反对软件商业化针锋相对地制订了 GPL（General Purpose License，通用公共许可证）。GPL 力图保证用户共享和修改自由软件的自由，保证自由软件对所有用户是自由的。到 20 世纪 90 年代初，GNU 计划已经发布了数量可观的基于 GPL 的软件，其中包括著名的编译器套件 GCC、编辑器 Emacs 等，这些软件广泛用于各种工作站的商用 UNIX 系统上。但是，GNU 计划并未完全成功，因为它缺少最核心的部分，属于自己的操作系统。

1991 年，芬兰赫尔辛基大学的学生 Linus Torvalds 由于在使用教学型操作系统 Minix 的过程中感到不满意，萌生自己开发一个操作系统的念头。1991 年 9 月，Torvalds 通过网络发布了 0.01 版，后来该系统命名为 Linux 并基于 GPL 发布。Linux 是一个类 UNIX 的操作系统，正好弥补 GNU 计划中缺失的一环，两者相得益彰，所以有 GNU/Linux 的说法。分布在世界各地的许多志愿开发者陆续加入 Linux 内核的开发，使 Linux 的发展速度非常快。1994 年 3 月，Linux 1.0 正式发布。1996 年 6 月，Linux 2.0 发布。1999 年 1 月，Linux 2.2 发布。此时

Linux 已经获得世界性的知名度,以 IBM 为首的许多大型商业公司也大力支持 Linux 的发展, Linux 在操作系统的市场份额也稳步上升。Linux 2.4.0 在 2001 年 1 月发布, Linux 2.6.0 在 2003 年 12 月发布, 目前 Linux 内核的最新版本已经到 2.6.28。

Linux 的蓬勃发展也带动了更多的软件以 GPL 许可发布,但是与此同时, 自由软件的领袖 Richard Stallman 对商业软件的激进态度导致一些商业公司对此驻足观望。为此 Bruce Perens 和 Eric Raymond 提出了“Open Source”(开放源码)的概念,下面列出 Bruce Perens 定义“Open Source”的 10 条标准:

- ① 自由再散布。允许获得源代码的人可自由再将此代码散布。
- ② 源代码。程序的可执行文件在散布时, 必须以附带完整源代码或是可让人方便地获得源代码。
- ③ 衍生著作。对源代码修改后, 在依照同一授权条款的情形下再散布。
- ④ 原创作者程序源代码的完整性。意即修改后的版本, 需用不同的版本号码, 以与原来的程序代码有所区别, 保障原来程序的代码完整性。
- ⑤ 不得对任何人或团体有差别待遇。
- ⑥ 对程序在任何领域内的利用不得有差别待遇, 意即不得限制商业使用。
- ⑦ 散布授权条款。若软件再散布, 必须用同一条款散布。
- ⑧ 授权条款不得专属于特定产品。
- ⑨ 授权条款不得限制其他软件。
- ⑩ 授权条款必须技术中立。

开放源码相比自由软件的要求相对宽泛一些, 如以 GPL, BSD License, LGPL 等许可证下发布的软件均可称为开放源码软件, 在一定程度上鼓励了更多的公司和组织加入到开放源码的阵营。

1.2 Linux 的基本特性

Linux 之所以能跟各种商业操作系统一争短长, 主要在于它丰富的特性。简单概括如下:

- ① Linux 是免费的, 比起 Windows 之类的操作系统, 优势一目了然。
- ② 支持多用户多任务, 用户界面良好。多用户多任务是现代操作系统的一个基本特点。早期的 Linux 在图形界面方面跟 Windows 还有一些差距, 但是现在差距已经非常微小。而命令行界面向来是 UNIX 系统的强项。
- ③ 支持多处理。Linux 不仅支持单处理器, 而且也支持 SMP(对称多处理, 包括近年来的多核芯片)和 NUMA(非一致性存储访问)体系结构。此外, 在集群计算领域, Linux 也占据了非常重要的地位。
- ④ 良好的可移植性。Linux 是支持硬件平台最多的操作系统, 这方面远远超过 Windows 等商业操作系统。从掌上设备到微型计算机, 从大量的 RISC 工作站再到 IBM 的大型计算机, 都可以见到 Linux 的身影。
- ⑤ 可完全定制, 从而灵活应用于各种不同场合。Linux 内核有上千个内核编译选项, 用户可根据自身需要选择相应的功能, 即可以裁剪成很小的内核用于嵌入式设备, 也可以定制成功能丰富的服务器级别内核。用户甚至可以亲自修改内核以满足自己的需要。
- ⑥ 提供了丰富的网络功能。Linux 本身就是通过 Internet 发展起来, Linux 在通信和网络

功能方面的支持也优于其他操作系统。它不仅支持通用的如以太网、无线网、蓝牙技术等，连不太常见的 DECnet 和 Acorn Econet 等都提供支持。

⑦ 性能出色且安全稳定。Linux 内核开发者总是把性能作为设计的首要目标，许多性能测试比较都证实 Linux 非常高效。Linux 可以连续运转数月乃至数年而无须重新启动，这是许多服务器架设在 Linux 平台上的原因。Linux 采取许多安全技术措施，包括权限控制、审计跟踪、授权机制等。功能强大高效的防火墙 Netfilter/Iptables 能有效防止非法入侵。美国国防部将计算机安全等级划分为四类七级，这七级从低到高依次为 D, C1, C2, C3, B1, B2, B3, A1。Linux 2.6 内核吸收了由美国国家安全局 NSA 开发的访问控制体制 SELinux (Security-Enhanced Linux)，使 Linux 的安全级别从原来的 C2 级可以达到 B1 级。

⑧ 良好的兼容性。这表现在多个方面。首先，Linux 是一个与 IEEE POSIX (Portable Operating System Interface) 兼容的操作系统。POSIX 标准是为提高 UNIX 系统之间的移植性而制订的。其次，其他操作系统的二进制程序可以直接到 Linux 运行，如 Windows 程序就可以直接跑在 Linux 下的 Wine 模拟器上。此外，许多操作系统的文件分区可以被 Linux 直接访问，如 Windows 的 FAT 分区和 NTFS 分区在 Linux 下均能直接识别，反之 Linux 的 Ext3 分区 Windows 就无法识别。

⑨ Linux 社区支持快速。Linux 社区有良好的运作机制，当用户的问题发到相关新闻组或邮件列表时可能马上得到回应。如果发现 Linux 的 bug 并报告，也许没过多久补丁就出来了。当有新的硬件推出时，Linux 社区的开发人员马上就能提供支持，典型的例子是 Intel 的 Itanium 处理器和 IBM 的 Cell 处理器推出后不久 Linux 就能运行在上面。

1.3 Linux 内核的开发模式与内核版本号

内核版本是在 Linus 领导下开发的系统内核的版本号。Linux 内核版本号的递增方式在 2.6.0 前后是有所不同的。下面分别介绍。

在 2.6.0 版本之前，内核版本号以 A.B.C (如 1.2.3, 2.3.12, 2.4.18, 2.5.20) 的形式发布，其中：

① A 表示内核主版本。它很少修改，只有代码出现重大变化时才可能变动。目前只变动过两次，1994 的 1.0 和 1996 年的 2.0。

② B 表示内核次版本。如果 B 是偶数，表明该版本是稳定内核，如 1.2.3 和 2.4.18 都是稳定内核。如果 B 是奇数，表明该版本是开发版，不一定可靠，如 2.3.12 和 2.5.20。市场上发行的 Linux 当然都是基于稳定版本的内核，而与开发版内核打交道的主要是内核开发人员。

③ C 表示发布号。以 2.4.18 的发布为例，该版本相对 2.4.17 内核并没有重要特性上的区别。可能只是修订一些 bug 和安全上的漏洞，还可能添加了一些硬件驱动。但是开发版本不是这样，见下面的说明。

重大特性的修改在稳定版本中基本是不会出现的，因为可能导致内核不稳定。如果添加重要特性，必须在开发版本中进行，我们以 2.4 到 2.6 的演进过程为例说明，2.4.0 发布后，Linus 持续发布 2.4.x 稳定版本，一直到 2.4.15。Linus 把 2.4.15 后续版本的维护工作交给 Marcelo Tosatti，而 Linus 基于 2.4.15 分支开发版本 2.5.0。开发版本允许开发者对内核尝试全新的特性，如果被证实有效，则可能被吸收进 2.5.x，像 O(1) 调度和可剥夺内核特性都是这样加入的，当然加入的新特性如果发现不足之处也可能在后面的版本中被剔除。众多内核开发

者的主要注意力都集中在 2.5.x，使得 2.5.x 的新功能快速增加，发布号的变迁也非常快。当 Linux 发布 2.5 系列的最后一个开发版本 2.5.75 后，基于该版本 Linux 终于发布了全新的（相对 2.4.x 而言）稳定内核版本 2.6.0。

上述开发方式的一个缺点是发布周期太长，从 2.4 到 2.6 花了将近三年时间，企盼新特性的 Linux 发行商和广大用户都不愿意见到这种情况，于是 2.6.x 内核的开发模式就发生了变化。

在 2.6.0 版本之后，所有稳定内核版本号以 2.6.C[D]（如 2.6.8，2.6.8.1）的形式发布。C 编号的变化可能意味着内核代码的大量修改以及重大特性的引入。数字 D 的引入主要是处理严重的错误或修订 bug 以及安全补丁，如 2.6.8.1 的发布是修改 2.6.8 版本中 NFS 代码的错误。

在 2.6 开发过程中，Andrew Morton 的地位仅次于 Linus Torvalds，他维护 2.6.x-mm 分支，该分支包含了大量补丁，如果一个补丁在 -mm 分支里证实了自己的价值，Andrew 或者子系统维护者就会把它提交给 Linus，以求被收录于 Linux 发布的内核。为叙述方便，我们以 2.6.26 到 2.6.27 的发布说明其开发流程。在 2.6.26 发布后，2.6.26.x 的发布由一个稳定版小组而非 Linus 本人负责，与此同时，大约两周的时间合并窗口被打开，数量众多的补丁（可以包含新特性而且大部分在 -mm 树中经过了一段时间的测试）被提交给 Linus，然后是 2.6.27-rc1 的发布（rc 是 release candidate 的缩写）。在 rc1 发布后，涉及新特性的补丁暂时不被接纳，只能等到发布 2.6.28-rc1 的时候。但是其他的补丁依然可以进入 2.6.27-rcx（x 为 2，3，…）系列。Linux 一般每周发布一次 2.6.27-rcx，经过多次这样的发布，当 rc 系列逐渐稳定后，如 2.6.27-rc9 发布后，2.6.27 正式发布，同时 2.6.28-rc1 的合并窗口开启。这种开发模式每年可以发布多个 2.6.x 版本，极大地缩短了发布周期。

1.4 Linux 发行版介绍

人们常说的 Linux 可能有两个含义，一个指 Linus 领导发布的内核，另一个指 Linux 的发行版。对于普通用户来讲，即使有内核依然做不了什么事情，还需要各式各样的软件。目前有大量软件是开源的，下面列举若干：

- Web 服务器，如 Apache。
- 数据库，如 MySQL，PostgreSQL。
- 网页设计语言，如 PHP。
- 图像处理软件，如 GIMP。
- 网页浏览器，如 Firefox。
- 办公套件，如 OpenOffice。
- 邮件，如 Thunderbird。
- 开发工具与库：GCC，GDB，Glibc。

Linux 发行版可能是由一个组织、公司或个人发行的。通常来讲，一个典型的 Linux 发行版包括：Linux 内核，GNU 工具和库，图形界面的 X Window 系统和相应的桌面环境，如 KDE 或 GNOME，还包括其他的自由软件或开源软件。一些商业 Linux 发行版中也有一些专有软件。发行版往往基于不同的目的而制作，例如，国内本土化的红旗 Linux、应用嵌入式场合的 uclinux 或着重桌面用户体验的 Fedora 等。目前 Linux 的发行版已经有几百种，下面介绍几个比较流行的发行版。

① Debian。最早由 Ian Murdock 于 1993 年发布，现在由来自世界各地的志愿者开发和

维护，它不属于任何的商业公司，完全由开源社区所有。Debian 软件包管理工具 `dpkg` 非常强大，在 Debian 上安装、升级、删除和管理软件极为容易。Debian 以稳定性闻名，所以很多服务器都使用 Debian 作为其操作系统。追求稳定性导致 Debian 新版本的发布周期较长，较少桌面玩家会选用它。

② Fedora。是由红帽公司赞助，由开源社区与红帽公司合作开发的项目。Fedora 的定位主要是桌面用户，对于使用者而言，它是一套功能完备、更新快速的免费操作系统，而红帽公司则将它作为许多新技术的测试平台，被认为可用的技术最终会加入到它的商业发行版 Red Hat Enterprise Linux 中。Fedora 大约每 6 个月发布一个新版本，目前最新版本是 Fedora 10。

③ Ubuntu。由 Mark Shuttleworth 在 2004 年 10 月发布首个版本，目前最新版本是 8.10。Ubuntu 基于 Debian 发行版和 GNOME 桌面环境，与 Debian 的不同在于它每 6 个月会发布一个新版本。Ubuntu 十分重视系统安全，所有系统相关的任务均需使用 `sudo` 指令，这是它的一大特色，这种方式比传统的以系统管理员账号进行管理的方式更安全；Ubuntu 还十分注重系统的易用性，标准安装完成后，一般就可以投入使用。

④ Suse。原是以 Slackware Linux 为基础，并提供完整德文使用界面的产品，在欧洲市场占有相当份额。2004 年 1 月被 Novell 收购，目前最新版为 Suse Linux 11.0。Suse 在安装时能自动调整 NTFS 分区，为初学者减少了不必要的麻烦。

1.5 Linux 内核源代码组织结构

Linux 内核源代码可以从官方站点 <http://www.kernel.org/> 下载。下载的源代码是压缩包格式，可以在 Linux 或者 Windows 环境下解压到某个目录，可以看到该目录下包含许多子目录，下面对它们进行简单介绍。

Arch——包含体系结构相关的代码，每种体系结构都有一个相应的子目录。如 x86 相关的代码放在 `i386` 目录下。

Block——Block I/O 层的代码，包含多种磁盘 I/O 调度算法。

Crypto——各种加密算法。

Documentation——与内核相关的文档，组织比较零乱，涉及面很广。但有些非常有用，介绍了内核某些功能模块的设计原理。

Drivers——各种设备驱动程序。

Fs——内核支持的各种文件系统，如 EXT3，NTFS 等。

Include——包含了绝大部分内核头文件。

Init——内核启动和初始化代码。

Ipc——进程间通信代码。

Kernel——最核心部分，包括进程管理、同步原语的实现等。

Lib——内核的辅助函数。

Mm——存储管理子系统，与平台相关的部分在 `arch/*/mm` 目录下。

Net——网络子系统，包含多种网络协议的实现。

Scripts——包含构建内核的脚本文件。

Security——包含 SELinux 的实现。

Sound——音频子系统。

Usr——EarlyUserSpace 特性的相关代码。

1.6 学习 Linux 的辅助软件介绍

使用本书的前提当然是读者手头有可用的 Linux 发行版，本书绝大部分内容适用于各种发行版，但极小一部分内容是基于 Fedora 的，所以**推荐使用 Fedora**。现在的 Linux 发行版安装已经非常简单，即便是新手，按照安装向导的提示一路下去也很容易成功。对很多初学者安装最大的困扰是一开始机器安装了 Windows，没有为 Linux 预留磁盘分区，解决该问题的办法有很多，一种简单的方法是在 Windows 里面安装虚拟机软件，再在该虚拟机中安装 Linux。

虚拟机产品有很多，如 VMware，Virtual PC 和 Bochs 等。笔者试用的是 VMware 工作站版本，可在网址 http://www.vmware.com/download/desktop_virtualization.html 下载。这个软件可在 Windows 环境下模拟出一台甚至多台新的计算机，然后便可在这些新计算机上面安装任何操作系统，包括 Windows 和 Linux，而且对目前的操作系统没有影响。如图 1-1 所示，笔者的虚拟机是 VMware Workstation 6.0，虚拟机里面安装了两个 Linux 发行版，Redhat Linux 8.0 和 Fedora Core 5。使用虚拟机对内核编程有特别的好处，当编程错误导致内核崩溃时，机器重启只是软件重启而不是硬件级别的重启，用户依然可以做其他的工作。其次使用虚拟机的快照（snapshot）功能，系统可以快速启动。VMware 的具体安装和使用都比较简单，请读者自行参考有关文档，这里就不详细介绍了。

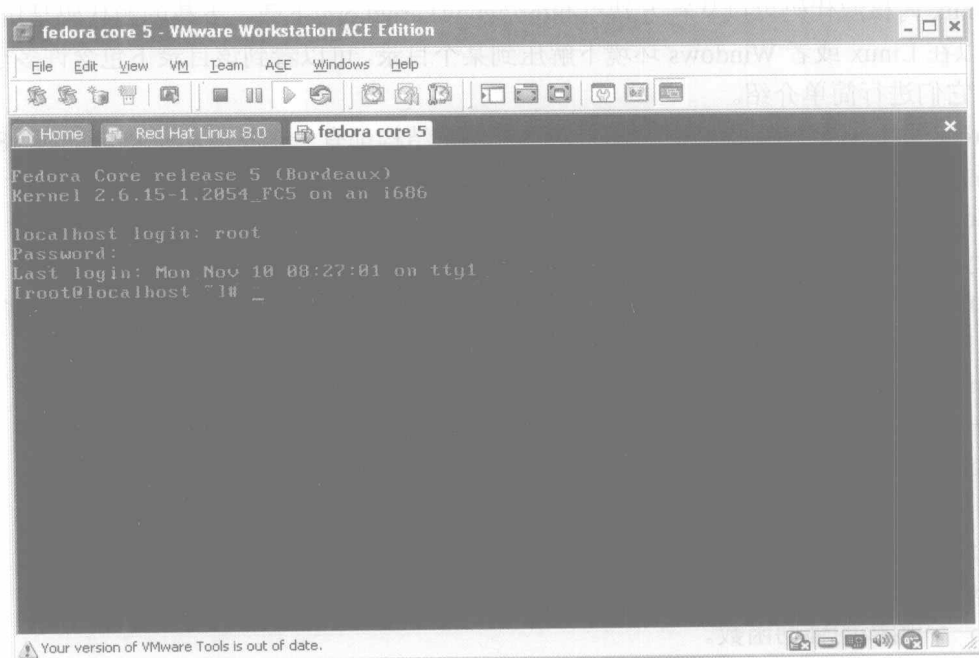


图 1-1 虚拟机 VMware Workstation

在此要介绍的第二个工具软件是 Source Insight，它在阅读 Linux 源代码的时候特别有用。Source Insight 可以从 <http://www.sourceinsight.com/downeval.html> 下载，运行该软件可免费试