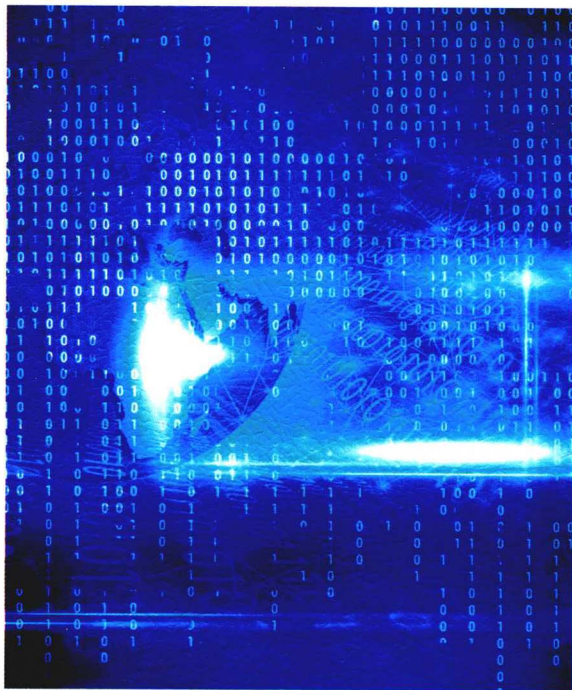


Linux编程

- ◆ Linux基本操作, Linux Shell编程
- ◆ 使用GCC、常用Linux函数库、Linux C程序排错、make
- ◆ UNIX I/O、内核文件I/O数据结构、文件组织与物理结构
- ◆ 进程控制原理, 信号机制, 基于fork、exec、exit、wait函数编写多进程并发程序
- ◆ 线程管理, 利用Pthreads线程库编写多线程并发应用
- ◆ 基于信号量与P/V操作解决同步及互斥问题, 经典同步问题
- ◆ 进程同步, 利用管道和IPC通信设施编写进程间同步应用
- ◆ 套接字、Web、HTTP协议, 网络通信程序结构, 并发网络应用编程



徐钦桂 徐治根 黄培灿 谢伟鹏 编著



清华大学出版社

应用规划教材

Linux 编程

徐钦桂 徐治根 黄培灿 谢伟鹏 编著

清华大学出版社

北 京

内 容 简 介

本书全面讲述 Linux 环境下基于 C 语言的系统编程技术以及相关的理论原理, 主要内容包括 Linux 基本操作、Shell 编程、系统 I/O 编程、文件系统、进程控制原理、多进程并发编程、信号机制、线程概念、多线程并发编程、同步与互斥的概念、基于信号量与 P/V 操作解决同步及互斥问题、经典同步问题、网络编程、并发网络应用编程等, 本书安排有大量的程序实例、课后作业, 还设计了很多示意图, 以帮助读者理解、运用书中介绍的概念、原理和技术。

本书内容丰富、结构合理、思路清晰、语言简明流畅、示例翔实, 可作为高等院校计算机类专业操作系统、Linux 编程等课程的教材, 还可作为 C 程序设计、嵌入式开发的参考资料。

本书的电子课件、习题答案和实例源代码可以到 <http://www.tupwk.com.cn/downpage> 网站下载。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Linux 编程 / 徐钦桂等 编著. —北京: 清华大学出版社, 2019

(高等学校计算机应用规划教材)

ISBN 978-7-302-51447-3

I. ①L… II. ①徐… III. ①Linux 操作系统—程序设计—高等学校—教材 IV. ①TP316.85

中国版本图书馆 CIP 数据核字(2018)第 243842 号

责任编辑: 胡辰浩

装帧设计: 孔祥峰

责任校对: 成凤进

责任印制: 丛怀宇

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市龙大印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 23 字 数: 617 千字

版 次: 2019 年 1 月第 1 版 印 次: 2019 年 1 月第 1 次印刷

定 价: 69.00 元

产品编号: 076639-01

前 言

Linux 是一种性能稳定的多用户网络操作系统，它与 UNIX 系统有相似的文件结构、用户接口和操作方式。Linux 虽然是开源免费的操作系统，但它继承了 UNIX 系统强大的功能、卓越的性能和稳定性。学习 Linux 系统编程不仅能帮助学生更好地巩固和理解操作系统的工作原理，还能培养学生的实践技能。因此，很多高校选择 Linux 系统作为操作系统原理课程的实例系统，选用 Linux 系统编程项目作为操作系统原理课程的实验内容。

由于 Linux 系统编程本身就是一门难度较大、内容繁多的课程，从中选取一些项目来开设操作系统实验，存在以下一些问题：①Linux 系统本身涉及很多理论、概念、技术、算法，操作系统这门课一般仅有十多个实验学时，由于学时太少，学生很难较好地掌握 Linux 系统编程技术，教学效果不佳；②目前很难找到将 Linux 系统编程技术与操作系统理论很好地融合的教材，结果是学习操作系统理论对学习 Linux 系统编程帮助不是很大，学习 Linux 系统编程对理解操作系统的理论帮助作用也非常有限；③一般基于 Linux 的实验指导或实验教材都写得比较简略，对 Linux 系统中多进程并发、线程编程、I/O 操作的介绍不完整、不系统，也没有补充必要的 C 语言语法知识，导致学生在学习过程中遇到很多难以克服的困难，丧失学习兴趣和信心。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实。每一章的引言部分概述了该章的作用和内容。在每一章的正文中，结合所讲述的关键技术和难点，穿插了大量极富实用价值的示例，并安排了有针对性的思考和练习，以帮助读者理解相关概念。每一章的末尾都安排了丰富的课后作业，有助于培养读者的分析能力和实际应用能力。

本书的目的是培养学生关于计算机系统的知识和能力，对操作系统和 Linux 系统编程进行整合，以 Linux 系统编程为主线，并纳入操作系统原理课程中的进程管理、信号量与 P/V 操作、文件系统等部分内容，将理论和实践有机地融合起来，可作为独立的操作系统实验或 Linux 系统编程课程开设，通过实践更好地理解课程理论，以提高教学质量。

除封面署名的作者外，参加本书编写的人员还有刘文果、李伟、肖捷、谭伟等人。由于作者水平有限，本书难免有不足之处，欢迎广大读者批评指正。我们的信箱是 huchenhao@263.net，电话是 010-62796045。

本书的电子课件、习题答案和实例源代码可以到 <http://www.tupwk.com.cn/downpage> 网站下载。

作 者
2018 年 7 月

目 录

第 1 章 Linux 系统文件操作 1	
1.1 UNIX/Linux操作系统简介..... 1	
1.1.1 UNIX简介..... 1	
1.1.2 Linux概述..... 2	
1.2 Linux系统目录结构..... 3	
1.3 Linux系统的安装、启动、登录、 用户界面与命令格式..... 5	
1.3.1 在VMware中用快照快速安装 Linux虚拟机系统..... 5	
1.3.2 启动与登录Linux..... 5	
1.3.3 三种系统操作界面..... 6	
1.3.4 Linux命令格式和说明..... 7	
1.4 Linux文件、目录操作及文件 属性、权限..... 9	
1.4.1 目录路径与目录操作..... 9	
1.4.2 文件属性与权限..... 13	
1.4.3 Linux文件操作命令..... 14	
1.4.4 修改文件属性..... 19	
1.4.5 使用通配符(“*”和“?”)匹配 文件名..... 21	
1.4.6 文件的压缩与打包..... 22	
1.5 输入输出重定向和管道..... 23	
1.6 本章小结..... 24	
课后作业..... 25	
第 2 章 Linux Shell 编程 26	
2.1 Shell编程基本概念..... 26	
2.1.1 Shell脚本程序的结构..... 27	
2.1.2 Shell脚本的创建与执行方法..... 27	
2.1.3 Shell变量与赋值表达式..... 28	
2.1.4 Shell输入输出语句..... 29	
2.1.5 终止脚本执行和终止状态..... 29	
2.2 Shell数学运算与字符串处理..... 32	
2.2.1 Shell数学运算..... 32	
2.2.2 Shell字符串处理..... 32	
2.3 Shell条件与if控制结构..... 33	
2.3.1 if语句..... 34	
2.3.2 test命令..... 36	
*2.3.3 复合条件检查..... 39	
2.3.4 case语句..... 40	
2.4 循环结构..... 40	
2.4.1 for循环结构..... 41	
2.4.2 while循环结构..... 42	
2.4.3 until循环结构..... 43	
2.5 Linux全局变量和环境变量..... 44	
2.5.1 Linux Shell层次结构..... 44	
2.5.2 Shell全局变量与局部变量..... 45	
2.5.3 Linux环境变量..... 46	
*2.5.4 Shell变量的删除和只读设置 方法..... 48	
2.5.5 Shell数组的定义和使用方法..... 48	
2.6 Linux文件I/O、I/O重定向和管道..... 49	
2.6.1 标准文件描述符..... 49	
2.6.2 I/O重定向..... 50	
2.6.3 管道..... 51	
2.6.4 从文件获取输入..... 52	
2.7 命令行参数..... 52	
*2.8 Shell函数..... 53	
*2.8.1 函数的基本用法..... 53	
*2.8.2 向函数传递参数..... 54	
2.9 本章小结..... 54	
课后作业..... 55	

第3章 Linux C编程环境	57	第4章 输入输出与文件系统	101
3.1 Linux C程序的编译与执行.....	57	4.1 文件系统层次结构.....	101
3.1.1 Linux环境下C程序的编译与 执行过程.....	57	4.1.1 文件系统层次结构简介.....	101
3.1.2 编译多个源文件.....	61	4.1.2 文件I/O库函数.....	102
3.1.3 使用头文件和库文件.....	62	4.2 系统I/O概念与文件操作编程.....	103
*3.1.4 使用gcc创建自定义库文件.....	65	4.2.1 UNIX I/O.....	103
3.1.5 gcc常用命令选项及用法.....	67	4.2.2 文件打开和关闭函数.....	104
3.2 Linux常用自带系统库.....	68	4.2.3 文件读写编程与读写性能 改进方法.....	107
3.2.1 数学函数.....	68	4.2.4 文件定位与文件内容随机读取.....	111
3.2.2 环境控制函数.....	69	4.2.5 任意类型数据的文件读写.....	113
3.2.3 字符串处理函数.....	69	4.2.6 用文件读写函数操作设备.....	115
3.2.4 时间函数.....	70	4.3 内核文件I/O数据结构及应用.....	117
3.2.5 数据结构算法函数.....	71	4.3.1 文件描述符和标准输入输出.....	117
3.3 诊断和处理Linux编程错误.....	75	4.3.2 文件打开过程.....	118
3.3.1 诊断和处理编译错误.....	75	4.3.3 内核文件I/O数据结构共享原理.....	119
3.3.2 处理系统调用失败.....	80	4.3.4 dup和I/O重定向.....	120
3.3.3 用断言检查程序状态错误.....	84	*4.4 用RIO包增强UNIX I/O功能.....	124
*3.4 用GDB/dd调试器诊断运行 错误.....	85	*4.4.1 RIO的无缓冲的输入输出函数.....	124
*3.4.1 用GDB调试程序运行错误的 实例.....	85	*4.4.2 RIO带缓冲的输入函数.....	125
*3.4.2 常用GDB命令.....	88	4.5 文件组织.....	128
*3.4.3 用ddd/GDB调试程序.....	89	4.5.1 文件属性、目录项与目录.....	128
3.5 命令行参数和环境变量的 读取方法.....	90	4.5.2 逻辑地址与物理地址.....	129
3.5.1 环境变量及其使用方法.....	90	4.5.3 创建和读写文件.....	130
3.5.2 命令行参数的使用方法.....	91	4.5.4 一体化文件目录和分解目录.....	132
*3.6 make工具.....	92	4.5.5 Linux分解式目录管理.....	133
*3.6.1 引入make工具的原因.....	92	4.5.6 读取文件元数据.....	135
*3.6.2 用makefile描述源文件间的 依赖关系.....	93	4.5.7 文件搜索和当前目录.....	136
*3.6.3 引入伪目标以增强makefile功能.....	94	4.6 文件物理结构.....	137
*3.6.4 用变量优化makefile文件.....	95	4.6.1 外存组织方式.....	137
3.6.5 用预定义变量和隐含规则 简化makefile文件.....	96	4.6.2 管理磁盘空闲盘块.....	141
3.7 本章小结.....	97	4.6.3 文件系统结构格式.....	143
课后作业.....	98	4.7 本章小结.....	144
		课后作业.....	144
		第5章 进程管理与控制	151
		5.1 逻辑控制流和并发流.....	151
		5.2 进程的基本概念.....	153
		5.2.1 进程概念、结构与描述.....	153

5.2.2	进程的基本状态及状态转换	155	6.2.3	线程间数据传递	219
5.2.3	对进程PCB进行组织	156	6.3	多线程程序中的共享变量	221
5.2.4	进程实例	157	6.3.1	进程的用户地址空间结构	222
5.2.5	操作进程的工具	158	6.3.2	变量类型和运行实例	223
5.2.6	编程读取进程属性	160	6.3.3	共享变量的识别	223
*5.2.7	进程权限和文件特殊权限位	161	6.4	线程同步与互斥	224
5.3	进程控制	163	6.4.1	变量共享带来的同步错误	224
5.3.1	创建进程	163	6.4.2	临界资源、临界区、进程(线程) 互斥问题	229
5.3.2	多进程并发特征与执行流程 分析	170	6.4.3	用信号量与P/V操作保证临界区 互斥执行	230
5.3.3	进程的终止与回收	173	6.4.4	用信号量及P/V操作解决资源 调度问题	233
5.3.4	让进程休眠	177	6.4.5	用Pthreads同步机制实现线程的 互斥与同步	237
5.3.5	加载并运行程序	178	6.4.6	共享变量的类型与同步编程小结	242
5.3.6	fork和exec函数的应用实例	180	6.5	经典同步问题	242
*5.3.7	非本地跳转	184	6.5.1	生产者/消费者问题	243
5.3.8	进程与程序的区别	186	6.5.2	读者/写者问题	245
5.4	信号机制	186	*6.6	其他同步机制	246
5.4.1	信号概念	186	*6.6.1	AND型信号量	246
5.4.2	信号术语	188	*6.6.2	信号量集	247
5.4.3	发送信号的过程	188	*6.6.3	条件变量	248
5.4.4	接收信号的过程	191	*6.6.4	管程	250
*5.4.5	信号处理问题	193	*6.7	多线程并发的其他问题	251
*5.4.6	可移植信号处理	197	*6.7.1	线程安全	251
*5.4.7	信号处理引起的竞争	198	*6.7.2	可重入性	253
*5.5	守护进程	201	*6.7.3	线程不安全库函数	254
5.6	进程、内核与系统调用间的 关系	203	*6.7.4	线程竞争	254
5.7	本章小结	204	6.8	使用多线程提高并行性	257
	课后作业	205	6.8.1	顺序程序、并发程序和并行 程序	257
第6章	线程控制与同步互斥	211	6.8.2	并行程序应用示例	258
6.1	线程概念	211	6.8.3	使用线程管理多个并发活动	262
6.1.1	什么是线程	211	6.9	本章小结	264
6.1.2	线程执行模型	212		课后作业	265
6.1.3	多线程应用	213	第7章	进程间通信	273
6.1.4	第一个线程	213	7.1	管道通信	273
6.2	多线程并发特征与编程方法	215			
6.2.1	Pthreads线程API	215			
6.2.2	多线程并发特征	217			

第 1 章

Linux 系统文件操作

本章主要介绍 Linux 系统的基本知识,包括 Linux 系统简介、Linux 系统的目录结构、文件类型、文件权限、Linux 命令格式以及文件目录的基本操作,为在 Linux 环境下进行编程打下基础。

本章学习目标:

- 了解 UNIX 与 Linux 系统的基本特点和发展历程
- 理解 Linux 系统的目录结构
- 掌握 Linux 系统的安装、启动、登录方法
- 掌握 Linux 文件属性和权限概念
- 掌握 Linux 文件路径的概念和通配符的含义
- 掌握常用的 Linux 文件与目录操作命令
- 掌握 Linux 文件的打包及解包方法
- 理解 I/O 重定向和管道的功能及基本概念

1.1 UNIX/Linux 操作系统简介

1.1.1 UNIX 简介

1969 年, Bell Labs(贝尔实验室)的 Ken Thompson 和 Dennis Ritchie 出于兴趣开发了一种多用户、多任务、多层次的操作系统 UNIX, 1971 年完成第一版的开发, 实现了多任务管理、文件操作、网络通信功能, 性能优越。

1973 年, Dennis Ritchie 创造了 C 语言, 与 Ken Thompson 一起用 C 语言重写了 UNIX 的第三版内核, 使维护和移植变得便利, 得到科研机构与企业的大力支持, 逐渐形成。UNIX AT&T System V Release 4(SVR4)和 BSD 两个版本系列:

- 加利福尼亚大学 Berkeley 分校于 1978 年开发出研究版本 BSD UNIX, 于 1994 年开发出 4.4 BSD 版本, 并成为现代 BSD 基本版本。
- AT&T 于 1983 年开发出商业版本 System V 版本 1, System V 版本 4(称为 SVR4)大获成功,

基于 SVR4 造就了 IBM 的 AIX 和 HP 的 HP-UX。

UNIX 系统在金融、教育、科研、军事等领域获得广泛应用，成为大学师生研究、学习操作系统原理首选的实例系统。UNIX 的主要版本有以下几种。

(1) AIX: IBM 基于 SVR4 开发的一套 UNIX 操作系统，性能高、安全、可靠性高，被广泛用于银行领域。

(2) Solaris: Sun Microsystems 于 1982 年推出基于 BSD UNIX 的 Sun OS，随后在接口上向 SVR4 靠拢，新版本称为 Solaris，性能高、处理能力强，有 GUI，在高校、科研院所用得多。

(3) HP-UX: 惠普(HP)公司以 SVR4 为基础研发出的类 UNIX 操作系统。

(4) IRIX: SGI 公司以 SVR4 与 BSD 延伸程序为基础研发出的 UNIX 操作系统，具有很强的图形处理功能，在游戏设计中使用广泛的三维图形编程库 OpenGL 从此而来。

尽管 UNIX 系统具有技术先进、性能高、安全性好等优点，但 UNIX 的不同版本间不兼容，给应用开发带来极大负担。搭建 UNIX 系统也涉及非常昂贵的费用，计算机硬件、UNIX 系统、开发工具、应用软件都需要分别计费，通常搭建一套带开发系统的 UNIX 工作站的费用达数十万元，很多用户负担不起，很多学校买不起。另外，UNIX 系统源码不开放，还给学习、研究带来不便。UNIX 厂商间恶性竞争削弱了 UNIX 系统的技术优势，在此过程中，微软公司的 Windows 操作系统得到迅速发展，占据桌面领域的市场。

1.1.2 Linux 概述

为方便广大师生学习、研究 UNIX 系统，1991 年，芬兰的林纳斯·托瓦兹(Linus Torvalds)开发了一套多用户、多任务、多线程的类 UNIX 操作系统，即 Linux。Linux 继承了 UNIX 系统强大的功能和性能，采用与 UNIX 系统兼容的操作命令，兼容 UNIX 编程接口规范 POSIX。学会操作使用 Linux，一般就可操作 UNIX 系统，掌握了 Linux 环境编程，就能在 UNIX 环境下做编程开发。

Linux 系统运行于廉价的 PC 上，免费使用，加入 GNU 联盟，开放源码，鼓励广大师生使用和开发 Linux 环境的配套软件，使得 Linux 环境的各种开发工具(如 gcc)和应用软件变得非常齐全，而且全部开放源码、免费使用、免费升级。

2001 年，Linux 2.4 版本内核发布。2003 年，Linux 2.6 版本内核发布，使 Linux 逐渐成为一种成熟的操作系统，在很多关键领域得到应用。目前常见的 Linux 内核版本有 Linux 2.4、Linux 2.6、Linux 3.2、Linux 4.6 等。

Linux 系统自 1991 年诞生以来，借助 Internet，通过全世界各地编程爱好者的共同努力，已成为今天世界上使用最多的一种类 UNIX 操作系统。Linux 可安装在各种计算机硬件设备上，比如个人计算机、大型机、超级计算机、Android 手机、平板电脑、路由器，世界上运算最快的 10 台超级计算机全部运行 Linux 操作系统。目前主流大数据平台 Hadoop 的每个节点都是一个 Linux 系统。

Linux 系统开源、完全免费、安全性好、可靠性高、支持多种平台；它功能强大，目录结构、基本命令与 UNIX 一致，可为未来学习 UNIX 系统打下很好的基础。不同厂商将 Linux 内核与外围实用程序和文档包装，提供安装界面和系统配置、管理工具等，形成发行系统，目前主要发行版本有：Red Hat Enterprise、Fedora、Ubuntu 等。不同 Linux 发行版本都采用至今仍由 Linux 系统创始人林纳斯·托瓦兹维护的同一个内核版本，在某种发行版本下编写的源程序和可执行程序不加修改即可在另一种 Linux 发行版本下运行，完全克服了 UNIX 不同发行版本间的不兼容问题。

✎ 思考与练习题 1.1 UNIX 系统得以发展的主要原因是什么？进入 20 世纪 90 年代和 21 世纪后，阻碍 UNIX 进一步发展的原因又是什么？

✎ 思考与练习题 1.2 近年来，Linux 系统从诞生到得以广泛应用的原因又是什么？

1.2 Linux 系统目录结构

图 1-1 是 Linux 系统目录结构，它与 UNIX(如 IBM AIX、Sun Solaris)系统具有大体一致的目录结构。与 Windows 系统不同，Linux 系统目录结构是一棵树，树根是/，每个文件和目录的路径都是以“/”开始的一条路径，如/home/can。Linux 系统没有盘符概念，除根分区外，其他硬盘分区的文件系统都挂载到某个路径以“/”开始的目录下。其中主要目录路径、功能，以及 Windows 环境下的对等目录或设施如表 1-1 所示。

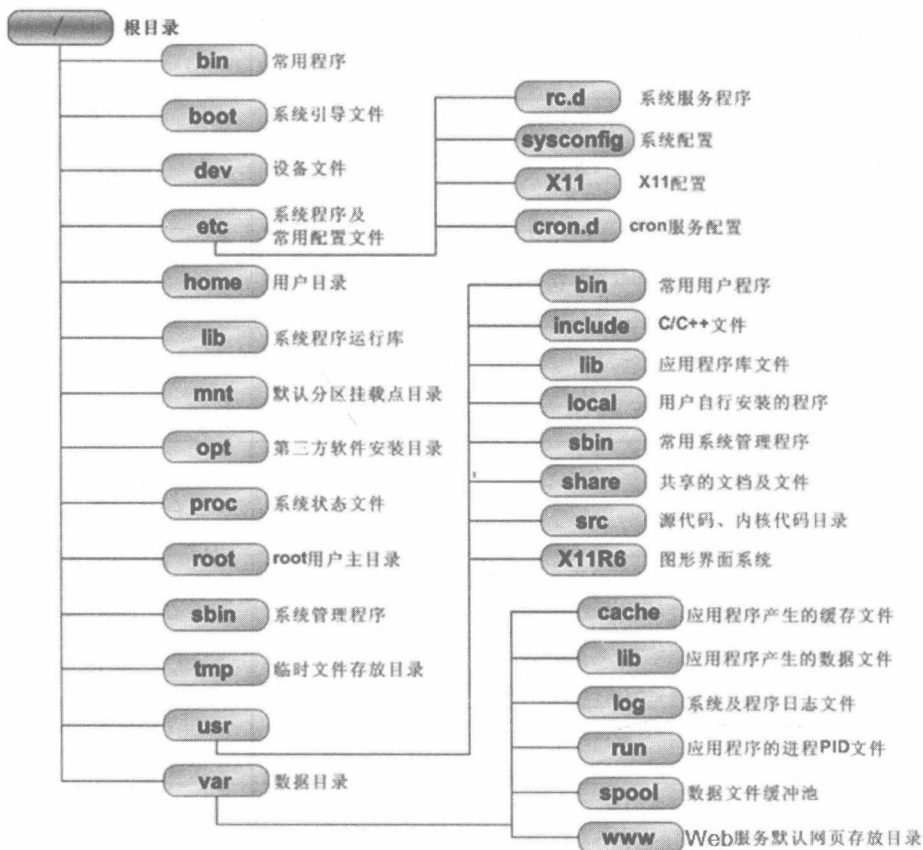


图 1-1 Linux 系统目录结构

Linux/UNIX 采用以上目录结构规范的好处有：

(1) 用户创建的文件全部放在/home 目录下，一来便于实施用户权限管理，二来可创建一个专门用于保存用户文件的分区，挂载到/home 目录下，方便管理；

(2) 可创建专用系统分区，保存 Linux 系统文件，以只读方式挂载在/usr 目录下，防止恶意用户

或病毒破坏系统目录，提高系统安全性；

(3) 可创建一个专用分区，保存动态增长的文件，以读写方式挂载到/var目录下，万一该分区受到破坏，整个系统不受影响，提高系统可靠性；

(4) 这种目录结构规范来自于 UNIX 系统，所有的 UNIX 和 Linux 目录结构与上述规范大体相似，使 UNIX/Linux 系统具有很好的向前兼容性，同时也方便人们的学习。

表 1-1 Linux 目录结构说明

目录	描述	助记单词	Windows 系统 对应目录
/	根目录，所有的目录、文件、设备都在根目录/下，根目录/就是 Linux 文件系统的组织者，也是顶级目录		
/bin	bin 就是单词binary的英文缩写，含义是二进制。在一般的系统中，可以在这个目录下找到 Linux 常用的用户命令	binary (二进制)	C:\WINDOWS\ system32
/boot	Linux 内核及系统引导过程所需的文件所在目录，比如 vmlinuz initrd.img 文件就位于这个目录中。一般情况下，GRUB 或 LILO 系统引导管理器也位于这个目录中	boot	
/dev	dev 是单词device的英文缩写，这个目录中包含 Linux 系统中使用的所有设备文件，是访问这些外部设备的一个入口，使用户能够用操作文件的方法操作这些外部设备	device	
/etc	目录/etc 中存放各种系统配置文件，其中还有子目录、系统网络配置文件、文件系统、X 图形界面配置、设备配置、用户设置信息等都在这个目录中	etcetera (等等)	注册表
/home	普通用户的家目录，如果建立一个用户，用户名是"xx"，那么在/home 目录下就有一条对应的/home/xx 路径，它是用来存放用户文件的主目录		C:\Documents and Settings
/include 或 /usr/include	C/C++源程序所需的系统头文件默认所在的目录		
/lib /usr/lib	lib 是 library 的缩写。这个目录用来存放系统的库文件。几乎所有的应用程序都会用到这个目录中的库文件	Library (库)	C:\WINDOWS\ system32
/lost+found	在 ext2 或 ext3 文件系统中，当系统意外崩溃或机器意外关机而产生一些文件碎片时，将它们放在这里		
/mnt	这个目录一般用于管理存储设备的挂载目录，比如/mnt/cdrom子目录下挂载 cdrom、/mnt/C 下挂载 Windows C 盘	mount	
/opt	这里主要存放那些可选的程序	option	
/proc	可以在这个目录下获取系统及各种进程信息，这些信息在内存中，由系统自己产生	process	注册表
/root	Linux 超级权限用户root的家目录		
/sbin 或 /usr/sbin	这个目录用来存放系统管理用的命令与程序，执行其中的命令时需要具备超级权限用户 root 的权限	system binary	

(续表)

目录	描述	助记单词	Windows 系统 对应目录
/selinux	对SELinux的一些配置文件目录, SELinux 可以让 Linux 更加安全	secure linux	
/srv	服务启动后所需访问的数据目录, 如 www 服务启动后, 读取的网页数据就可以放在/srv/www 中	server	
/tmp	临时文件目录, 用来存放不同程序执行时产生的临时文件。有些用户程序在运行过程中会产生临时文件	temporary	C:\Windows\Temp
/usr	这是系统存放程序的目录, 比如命令、帮助文件等, 当我们安装 Linux 发行版官方提供的软件包时, 大多安装在这里	UNIX System Resources	C:\Program Files
/var	这个目录中的内容经常变动, 其名字为单词 variable 的缩写, /var 下的子目录/var/log 用来存放系统日志; /var/www 子目录用于存放 Apache服务器网站的文件; /var/lib 子目录用来存放一些库文件, 比如 MySQL 的库文件以及MySQL数据库	variable	

✎ 思考与练习题 1.3 Linux 目录结构有何意义? /home、/usr、/etc、/var 等目录的用途是什么, 这样安排有何好处?

✎ 思考与练习题 1.4 简述/etc、/home/guest、/root、/var、/usr、/usr/lib、/bin、/tmp、/usr/include、/usr/sbin、/boot 等目录保存何种用途的文件。

1.3 Linux 系统的安装、启动、登录、用户界面与命令格式

1.3.1 在 VMware 中用快照快速安装 Linux 虚拟机系统

假设安装于 D 盘, 具体步骤如下:

- (1) 下载 VMWare;
- (2) 安装 VMWare(安装过程中需要重启计算机);
- (3) 下载 Ubuntu 快照并解压缩到 D 盘, 目录名为 D:\ubuntu;
- (4) 启动 VMWare, 输入序列号;

(5) 选择 File→Open, 选择打开 D:\ubuntu 下的.vmx 文件, 双击 My Computer 下的 Linux 或 Ubuntu, 启动 Ubuntu Linux;

(6) 选择 VM→Update VMWare Tools Installation, 更新 VMWare 工具, 安装可在主机与 Linux 虚拟机之间以拖放方式复制文件的功能。

1.3.2 启动与登录 Linux

1. 启动与登录系统

观看“Linux 系统启动与登录(以普通用户身份登录).exe”视频。与 Windows 系统一样, 为安全起见, Linux 系统启动后, 提示用户输入用户名与密码以完成登录, 才能使用系统执行任务。Linux 系统有两类用户: 普通用户与超级用户(管理用户)。超级用户为 root, 在系统安装过程中创建, 可执

行系统管理、维护、软件安装等工作，具有执行任何操作的权限，相当于 Windows 下的管理用户 Administrator。超级用户 root 的家目录为/root。

在实验中启动 VMWare 后，选择 Ubuntu 并启动 Linux 系统后，以普通用户 can 的身份登录，输入密码 123456 后进入系统。打开命令窗口，可输入用户命令，命令提示符为“\$”，视频中还演示了创建 hello.c 文件的过程。

当然也可以 root 用户身份登录，打开终端窗口，可执行任何操作，root 用户的密码也是 123456，为了能够与普通用户的操作环境有明显区别，root 用户的命令提示符是“#”。

2. 如何切换成 root 用户身份

在普通用户打开的以“\$”为提示符的终端窗口中，输入命令“su -”，接着输入 root 用户的密码，就可以暂时切换成 root 用户身份，命令提示符变成“#”，之后就可执行需要 root 用户权限的各种操作和命令了。执行完需要 root 用户权限的操作后，可执行“exit”命令，退出 root 登录身份，恢复成原来的普通用户身份。

3. Linux 系统的一般操作方式

一般以普通用户身份登录 Linux 系统，通过在终端窗口中输入操作命令来使用系统。Linux 系统提倡使用终端命令来执行各种操作的原因是，Linux 操作命令的种类异常丰富，功能强大，如果设计成由桌面系统启动，反而过于复杂，操作不便。

以普通用户身份登录的原因是，普通用户的操作权限受限，凡涉及系统配置、管理、维护的命令都无权执行，Linux 系统遭受攻击的威胁较小，系统安全性和可靠性较高。例如，当普通用户因误操作执行硬盘格式化或对系统文件执行删除操作时，系统将拒绝执行，可避免不必要的损失；又如，当普通用户意外双击执行病毒或木马程序时，病毒或木马程序的影响仅限于该用户的文件，而不至于危及整个系统的安全。

1.3.3 三种系统操作界面

Linux 提供了三种方法，以方便用户操作和使用 Linux 系统。

1. 图形界面

一般启动 Linux 系统后直接进入图形用户界面，通过选择主菜单或单击文件管理器，可执行系统管理和文件操作，图 1-2 是 Ubuntu 下的文件管理器界面，可通过下拉菜单、弹出菜单等方法执行文件、目录操作。

2. 命令界面

由于 Linux 系统的功能非常强大，系统操作种类异常丰富，而桌面环境通常难以支持太多操作功能，因此 Linux 的多数功能难以通过图形界面运行。为此，Linux 系统提供了大量的操作命令，通过终端窗口或命令窗口，输入命令来操作 Linux 系统，命令输出也显示在终端窗口中。命令界面还有一个好处，在未启动图形界面的系统中，也可方便地操作 Linux 系统。对计算机专业人员来说，往往更多地使用命令界面进行各种开发工作，因为这样效率更高。在图 1-3 中，输入命令 `ls` 可显示当前目录下的文件列表，而输入 `cat /etc/passwd` 则显示用户数据库文件/etc/passwd 的内容。

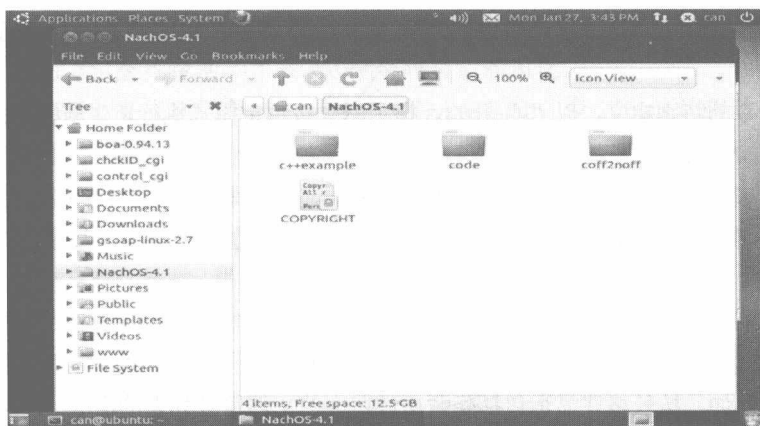


图 1-2 Ubuntu Linux 文件管理器

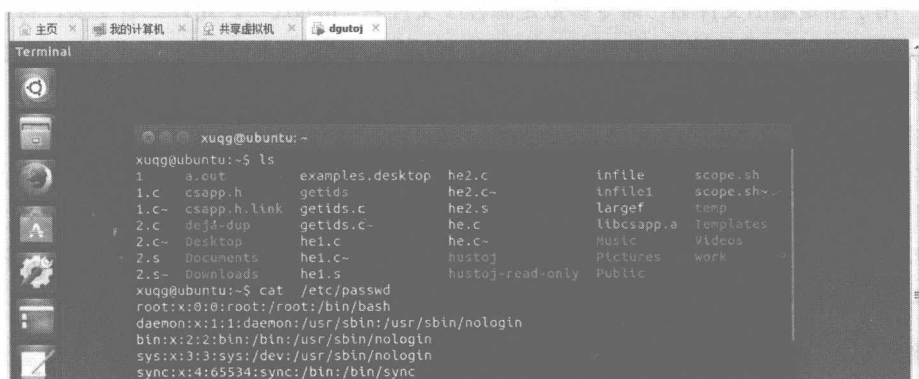


图 1-3 Linux 命令界面

3. 编程接口

编程接口是指在 C/C++ 语言程序(也包括其他程序)中调用 Linux 系统功能的方法, 一般是通过一些称为系统调用的库函数来实现的。例如: 文件操作流接口——`fopen`、`fread`、`fwrite`、`fclose`、`fseek`, 在程序中调用这些函数可打开、读、写、关闭文件, 以及移动读写指针; 文件与设备操作系统调用接口——`open`、`read`、`write`、`close`, 在程序中调用这些函数可打开、读、写、关闭文件或设备。本书后面各章主要讨论如何使用 Linux 库函数来进行 Linux 系统编程。

1.3.4 Linux 命令格式和说明

1. Linux 命令格式

Linux 系统命令遵循一种统一的格式, 一般形式为:

```
$ 命令名 选项 参数1 参数2 .....
```

命令名、选项、参数间以空格分隔, 在很多命令中, 选项与参数都有默认值, 各部分说明如下:

(1) 命令名

一般是由小写英文字母构成的字符串, 往往是表示相应功能的英文单词的缩写。例如, `date` 表示日期; `who` 表示谁在系统中; `cp` 是 `copy` 的缩写, 表示拷贝文件等。Linux 系统支持的用户命令主

要放在目录/usr/bin 和/bin 下。命令名中出现大写字母的命令一般都不是正确的系统命令。

(2) 选项

选项是对命令的特别定义，以“-”开始，指示命令按特定模式执行，生成输出。例如：加-l 选项的ls 命令“ls -l”表示以长格式显示文件列表，每个文件一行，显示文件名与属性；加选项-a 的ls 命令“ls -a”表示文件名以点(.)开头的隐藏文件也要显示出来；若同时使用多个选项，多个选项可共用一个“-”连起来，“ls -l -a”命令与“ls -la”命令的功能完全相同，显示包括隐藏文件的当前目录文件属性。

不同选项的书写一般没有先后限制，但如果选项本身带有参数，那么选项和参数必须在一起，中间不允许插入其他命令参数或命令选项。Linux 命令的选项一般位于命令参数之前，有时也放在命令参数之后。例如，按长格式显示包括隐藏文件在内的文件目录列表的命令“ls -l -a”也可写成“ls -a -l”。在将C 程序hello.c 编译成可执行程序hello 的命令“gcc -o hello hello.c”中，命令选项“-o hello”用于指定输出文件名，命令参数是源程序文件名“hello.c”，允许二者交换，将命令写成“gcc hello.c -o hello”，但不能写成“gcc -o hello.c hello”，因为命令选项名“-o”与其参数“hello”被命令参数hello.c 隔开了。

(3) 参数

提供命令运行的信息或是命令执行过程中使用的文件名。通常参数是一些文件名，告诉命令从哪里可以得到输入，以及把输出送到什么地方，例如命令“cp file1 file2”将文件file1 复制到文件file2。

✎ 思考与练习题 1.5 指出命令“ls -l -a /home/can”与“gcc -o he he.c”中，哪个是命令名、选项与参数，它们的含义是什么？

2. 命令说明

(1) 判断命令执行是否成功

一条Linux 命令仅在命令名、选项、参数全部正确时，才能正确执行，否则执行将失败，并显示出错信息，出错信息的格式为“命令名: 出错描述”。以下是命令执行失败的示例：

```
$ LS #命令名错误，显示目录列表的命令是小写字符串ls
bash: LS: command not found #显示命令LS 未找到错误
$ ls -P #命令ls 无选项-P
ls: invalid option - P
$ ls -l PP #文件PP 不存在
ls: cannot access PP: No such file or directory
```

(2) 命令输出

如果命令执行后有输出，成功执行后的输出信息紧接着命令串显示。由于很多Linux 命令没有输出，如下面将要介绍的目录与文件操作命令cd、mkdir、rmdir、rm、mv，它们执行完毕后，将立即显示命令提示符“\$”或“#”。由于含有错误的命令一定会显示出错信息，因此一条命令如果执行后没有任何输出而立即显示命令提示符，则说明该命令的执行一定是成功的。例如：

```
$ cd #该命令无任何输出，执行必然成功
$
```

(3) 联机帮助

Linux 操作系统的联机帮助对每个命令(包括主要系统配置文件)的准确语法都做了说明，可以使用man、info 等命令来获取相应命令的联机说明，如“man ls”和“info ls”。man 和info 命令一般

按字符 q 即可退出。

✎ 思考与练习题 1.6 查找帮助，了解 wc 命令的功能和用途。

✎ 思考与练习题 1.7 查找联机帮助，请给出文件/etc/fstab、命令 pwd 的用途。

(4) 本书命令输入说明

若命令提示符为“#”，则假定是以管理用户 root 身份登录；若命令提示符为“\$”，则假定是以普通用户 can 身份登录。为方便读者练习，书中的用户输入命令和信息以斜体文本行显示，系统显示内容以常规字体文本行显示，用户输入命令行后面以“#”开头的文字是对命令功能的解释，如图 1-4 所示。

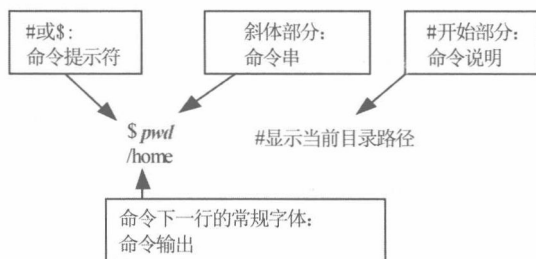


图 1-4 本书命令输入说明

1.4 Linux 文件、目录操作及文件属性、权限

通常，普通用户的主要工作一般是处理文件档案，通过命令从文件中读取输入数据，处理后，保存到另一文件。Linux 系统为每个普通用户在/home 目录下创建了一个以用户名命名的“家”目录，如用户 can 的“家”目录是/home/can，用户 guest 的“家”目录是/home/guest，但根用户 root 的“家”目录是/root。按照 Linux 系统权限管理规范，普通用户仅能在其“家”目录(用户主目录)下创建、修改、删除文件，而不能增删“家”目录之外其他目录中的文件，从而使系统有较好的文件保护能力。与文件操作相关的 Linux 命令主要包括：文件与目录操作命令、文件内容查阅命令、文件目录权限设置命令、文件搜寻命令等。

1.4.1 目录路径与目录操作

1. 绝对路径、工作目录和相对路径

Linux 里面的目录是“树状结构”，一般每个叶子节点为一个普通文件，每个分支节点为一个目录文件。要操作或访问某个文件，应通过路径方式给出文件所在位置。

- 绝对路径：要对某个文件(或目录)进行操作，可在命令中给出从根目录开始，直到所要操作文件名、中间以“/”隔开的完整路径，称为绝对路径，如显示文件内容命令 `cat /etc/passwd` 和 `more /home/can/NachOS-4.1/code/test/`。Linux 系统中文件路径的目录分隔符是“/”而不是“\”，这是与 Windows 系统另一个不同的地方。
- 工作目录：为缩短文件路径字符串长度，Linux 系统为每个命令窗口(Terminal，终端)和应用进程设置了一个工作目录(初始设置为用户的“家”，可用命令 `cd` 改变)，当用户操作工作