



特色教材·精选系列

FEATURE SELECTION
MATERIALS SERIES



UNIX系统 与软件开发

邵国金 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等学校计算机类专业特色教材·精选系列

UNIX 系统与软件开发

	邵国金	主 编
耿永军	张俊峰	副主编
郭 猛	沙 锋	参 编
张翼飞	王细薇	

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书以 Fedora 16、FreeBSD 9 和 Solaris 11 为蓝本,分四篇较全面地介绍了 UNIX/Linux 系统的使用、管理和开发。基础篇介绍了 UNIX/Linux 入门知识和基本操作;管理篇介绍了 UNIX/Linux 的常用管理内容,包括用户管理和密码管理,文件系统及其管理和使用,进程与作业管理,系统安装、启动分析、服务管理及常用网络应用,软硬件管理及系统的扩充与升级,网络管理与网络应用;编程基础篇介绍了 shell 编程,C 编程基础和方法;开发篇包括了文件部分系统调用与标准 I/O,进程环境,文件属性与目录编程,进程关系、进程控制与信号,进程间通信,线程编程基础,终端与 curses 库编程,数据库的使用及编程和网络编程。

本书从“应用型”人才培养目标出发,基于最流行版本的 Fedora、FreeBSD 和 Solaris 系统,基本上涵盖了比较流行的 UNIX/Linux 系统应用、管理和开发的内容;兼顾基本知识和基本理论的培养,内容翔实,结构清晰,实例充分、完整,具有较强的实用性和可操作性。

本书可作为高等院校“UNIX/Linux 操作系统管理和开发”课程的教材,也可作为 UNIX/Linux 操作系统爱好者或工作者的参考书。

图书在版编目(CIP)数据

UNIX 系统与软件开发/邵国金主编. —北京:中国铁道出版社,2013.10

普通高等学校计算机类专业特色教材. 精选系列

ISBN 978-7-113-16898-8

I. ①U… II. ①邵… III. ①UNIX 操作系统—高等学校—教材 IV. ①TP316.81

中国版本图书馆 CIP 数据核字(2013)第 200983 号

书 名: UNIX 系统与软件开发

作 者: 邵国金 主编

策 划: 巨 凤

读者热线: 400-668-0820

责任编辑: 王占清

封面设计: 一克米工作室

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.51eds.com>

印 刷: 北京华正印刷有限公司

版 次: 2013 年 10 月第 1 版

2013 年 10 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 30.75 字数: 912 千

书 号: ISBN 978-7-113-16898-8

定 价: 56.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 63549504

1. 关于本书

UNIX 历史悠久且丰富多彩,它于 1969 年诞生于贝尔实验室科学家 Ken Thompson 的脑海中,由于出生于计算机硬件还非常原始的时代,从而也造就了它少说多做的朴实风格。自 UNIX 诞生至今,UNIX 一直是各大学和计算机爱好者学习和研究操作系统的范本。

UNIX 是开源的先驱和典范,也是计算机爱好者和 hacker 们的最爱,同时 UNIX 也在不断地从爱好者们的成果中汲取营养,而使自己不断发展、不断成长,而且越来越健壮,并一直应用于生产和商业领域,为人类和社会的发展不断地做着贡献。随着 Linux 和 Android 等的出现更为 UNIX 家族增添了新的成员,也带来了更多的用户。

UNIX 家族庞大,成员众多,从大中型机到微型机,再到手持机和手机,到处都有它的身影和成功应用,很难说清它有多少厂家和版本。现在开源和免费的各种 Linux、BSD 和 Solaris,以及商用系统,如 IBM 的 AIX、HP 的 UX、SCO 的 OpenServer 和 UNIXWare、Apple 的 Mac OS X、Oracle 的 Sun OS 和 Solaris 等都是 UNIX 家族的重要成员。

笔者是 UNIX 系统的老用户和忠实的支持者,对 UNIX 有特殊的感情,有意为 UNIX 的应用和文化传播做一些工作,也愿意为 UNIX 方面的人才培养作一点贡献。但是,毕竟 UNIX/Linux 的版本和厂家太多,不能遍及,故本书只选择了可以免费获得并免费使用且无版权争议,又具有广泛代表性的 Fedora 16、FreeBSD 9 和 Solaris 11 进行介绍,同时也希望对其他版本的 Linux 和 UNIX 有所裨益。

必须要看到,将不同风格的 UNIX/Linux 放在一起介绍和讨论不是件容易的事,或者说是很有难度的,但这样做会给本书的使用者带来更多的信息和更大的益处,虽然不能完全做到“一本通”,但却能使使用者既能看到不同系统间的相同或不同点,领略不同的风格,又能从不同风格中领会程序或系统设计的意义和方法。

2. 本书的结构

本书共分 4 篇 19 章,大致结构如下。

(1) 基础篇:包括入门知识和 shell 与 shell 命令 2 章。内容包括系统简介,系统的开启与关闭、用户登录与注销;shell 基本功能与基本概念、基本操作命令和 shell 的启动。通过本部分的学习,可以使用户胜任 UNIX 系统基本操作工作。

(2) 管理篇:包括用户管理和密码管理,文件系统及管理,进程与作业管理,系统安装、启动与管理,软硬件管理及系统的扩充与升级和网络管理与网络应用等 6 章,内容几乎涉及 UNIX 系统日常管理的所有方面。通过本部分的学习,可以使用户了解 UNIX 系统日常管理任务和掌握管理工具,胜任管理工作。

(3) 编程基础篇:包括 bshell 编程及 C 编程基础和方法 2 章。通过本篇的学习可以使用户掌握 shell 编程技术和 C 编程的基础和方法,为 shell 命令的综合应用和编程打下基础。

(4) 开发篇:包括文件部分系统调用与标准 I/O,UNIX 系统进程环境,文件属性与目录编程,进程关系与进程控制,进程间通信,线程编程基础,终端与 curses 库编程,

数据库的使用及编程和网络编程等 9 章，内容涵盖了 UNIX 系统所有的基本开发部分。通过本篇的学习，可以使用户对 UNIX/Linux 系统有更深入的理解，并能从事 UNIX/Linux 应用软件的开发工作。

3. 本书的特点

本书的编写基于组织与编写者近 20 年的 UNIX/Linux 应用、管理和开发经验，层次分明、概念清楚、内容翔实、可操作性强，既便于读者循序渐进地系统学习，又能够使读者了解到 UNIX/Linux 的最新发展，具有以下特点：

- (1) 基于实际应用和开发经验，突出理论与实践的结合和实际动手能力培养。
- (2) 面向基础、面向应用、面向就业，重视应用型人才培养，具有较强的实用性和指导性。
- (3) 基于流行版本的 Fedora、FreeBSD 和 Solaris 系统，涵盖范围广，具有广泛的代表性。
- (4) 涉及 UNIX/Linux 基本操作、系统管理和软件开发的几乎所有内容。
- (5) 突出实例和操作步骤，所有示例和代码均可不加调试地直接使用。
- (6) 每章后都提供有一定数量的练习题，并提供电子课件和习题解答。

4. 编者信息

本书由邵国金任主编，耿永军和张俊峰任副主编。主编负责全书的策划与编者分工工作，并审定全部书稿；副主编协助主编做策划和审统稿工作。参编人员分工及完成情况如下：第 1~3 章由张俊峰编写，第 4~6 章由耿永军编写，第 7、8 章由沙锋编写，第 11、18 章由张翼飞编写，第 14、15 章由郭猛编写，第 10 章由王细薇编写，第 9、12、13、16、17 和 19 章由邵国金编写。

5. 适用对象

本书适合作为高等院校 UNIX/Linux 应用、管理与开发方面的教材，也可作为 UNIX/Linux 工作和管理者的参考书，更可作为 UNIX/Linux 操作系统爱好者的益友。

6. 其他说明

在本书的编写过程中，参考了大量的 UNIX/Linux 方面的书籍和网络资源，尤其是 Fedora、FreeBSD 和 Solaris 系统的网络资源、电子文档和 Wiki 百科等，在参考文献中不便逐一列出，在此一并表示感谢。

面对 UNIX/Linux 操作系统的新发展和新成就，要编写一本综合多个操作系统的高水平教材感到压力很大，加之理论水平不高和时间仓促等因素，疏漏之处难免，欢迎广大专家、学者和读者给予批评指正，本书编写组衷心希望能得到大家的支持和帮助，共同探讨 UNIX/Linux 操作系统的工作和教学体会，促进教学和应用水平的提高。

编 者

2013 年 6 月

基 础 篇

第 1 章 入门知识.....	1
1.1 UNIX 系统简介.....	1
1.1.1 UNIX 系统的产生 及演变.....	1
1.1.2 UNIX 系统的特点.....	3
1.2 所用系统简介.....	5
1.2.1 Linux 和 Fedora.....	5
1.2.2 BSD 与 FreeBSD.....	5
1.2.3 Sun OS 和 Solaris.....	6
1.3 系统的开关机.....	7
1.3.1 系统的开启与用户登录.....	7
1.3.2 用户的注销.....	9
1.3.3 系统关闭与重新启动.....	9
1.4 在线帮助.....	12
1.4.1 man 手册.....	12
1.4.2 info.....	13
习题.....	14
第 2 章 shell 与 shell 命令.....	15
2.1 shell 基本功能与基本概念.....	15
2.1.1 shell 基本功能.....	15
2.1.2 字符与保留字.....	16
2.1.3 文件命名及文件类型.....	18
2.1.4 目录结构与路径.....	19
2.1.5 shell 命令解释及执行.....	21

2.1.6 命令行编辑特性.....	21
2.1.7 环境变量与变量.....	22
2.1.8 标准流与输入输出重 定向.....	23
2.1.9 管道.....	25
2.1.10 引号机制、命令替换与 参数替换.....	25
2.1.11 shell 种类.....	26
2.1.12 shell 命令的返回值.....	27
2.2 基本操作命令.....	27
2.2.1 目录基本操作命令.....	27
2.2.2 文件操作基本命令.....	29
2.2.3 文本文件内容处理与 编辑基本命令.....	33
2.2.4 进程管理基本命令.....	42
2.2.5 时间管理命令.....	43
2.2.6 文件或目录比较命令.....	45
2.2.7 其他操作命令.....	47
2.3 shell 的启动.....	51
2.3.1 shell 的启动流程和工作 过程.....	51
2.3.2 用户登录控制与 shell 定制.....	52
习题.....	52

管 理 篇

第 3 章 用户管理和密码管理.....	53
3.1 UNIX 系统的用户和组.....	53
3.1.1 用户与 uid.....	53
3.1.2 用户组.....	53
3.2 与用户、组和密码管理相关 的文件.....	53
3.2.1 /etc/passwd.....	53
3.2.2 /etc/shadow.....	54
3.2.3 /etc/group.....	54

3.2.4 skel.....	55
3.2.5 其他文件.....	55
3.2.6 FreeBSD 9 的/etc/master.passwd 文件.....	55
3.3 用户管理命令.....	56
3.3.1 Fedora 和 Solaris 的用户 管理命令.....	56
3.3.2 FreeBSD 9 的用户管理 命令.....	57



3.3.3	更改用户的登录 shell.....	58	4.3.4	文件默认权限与 umask.....	74
3.4	组管理命令.....	58	4.4	权限管理命令.....	75
3.4.1	Fedora 和 Solaris 的组管理命令.....	58	4.4.1	umask.....	75
3.4.2	FreeBSD 9 的组管理.....	59	4.4.2	chmod.....	75
3.5	密码管理.....	59	4.4.3	chown.....	76
3.5.1	综述.....	59	4.4.4	chgrp.....	76
3.5.2	密码管理命令 (passwd)	60	4.4.5	ext2+ 文件系统的新增属性及其管理.....	76
3.5.3	用户上锁与解锁.....	60	4.5	文件系统管理及使用.....	77
3.6	与用户身份和位置相关的其他命令.....	61	4.5.1	UNIX/Linux 支持的文件系统.....	77
3.6.1	显示与用户和组相关的信息 (id)	61	4.5.2	UNIX/Linux 系统使用的存储设备.....	79
3.6.2	显示已登录用户的信息 (who) 及本用户名信息 (whoami)	61	4.5.3	文件系统创建.....	81
3.6.3	终端接收信息控制 (mesg)	62	4.5.4	文件系统的使用.....	90
3.6.4	向系统中已登录的用户发信息 (wall)	62	4.5.5	文件系统的检查、修复与同步.....	94
3.6.5	显示用户所使用的终端设备 (tty)	62	4.6	与文件系统管理相关的其他命令.....	96
3.6.6	不退出系统而将自己切换成其他用户或以其他用户身份工作 (su)	63	4.6.1	文件查找命令 (find)	96
3.6.7	以其他用户身份执行程序 (sudo)	64	4.6.2	文件复制命令 (dd)	98
3.6.8	改变进程的根目录 (chroot)	66	4.6.3	创建文件连接 (ln)	98
习题.....		67	4.6.4	特别文件创建 (mknod/mkfifo)	99
第 4 章 文件系统及管理.....		68	4.6.5	统计磁盘空间和文件系统的使用情况 (df)	99
4.1	硬盘存储结构与分区划分.....	68	4.6.6	统计目录使用磁盘空间情况 (du)	100
4.2	UNIX 文件系统内部结构.....	71	4.7	数据备份与压缩.....	100
4.2.1	UNIX 文件系统的特点.....	71	4.7.1	数据备份的任务.....	100
4.2.2	UNIX 文件系统的结构.....	71	4.7.2	文件的压缩与解压缩.....	101
4.3	文件系统权限及表示.....	73	4.7.3	磁盘文件归档管理命令 (tar)	103
4.3.1	三类人.....	73	4.7.4	文件系统备份与复制命令 (cpio)	105
4.3.2	三种权限.....	73	习题.....		107
4.3.3	权限表示.....	74	第 5 章 进程与作业管理.....		108
			5.1	程序和进程的概念.....	108
			5.1.1	程序、进程、作业和任务.....	108
			5.1.2	三类进程.....	108



5.1.3	系统的启动与进程树的形成	109	6.2.1	Linux 和 Solaris 的 GRUB 引导器	128
5.1.4	0#进程与 1#进程	109	6.2.2	GRUB 常用术语	129
5.1.5	进程状态及转换	110	6.2.3	操作界面	129
5.2	UNIX/Linux 的进程调度与信号	110	6.2.4	GRUB1 及其配置	130
5.2.1	调度策略与优先级的计算	111	6.2.5	GRUB2 及其配置	133
5.2.2	信号与软中断	111	6.2.6	系统的启动及启动参数修改	136
5.3	setuid、setgid 和 sticky 属性	112	6.2.7	FreeBSD 9 的启动	137
5.3.1	suid 和 sgid 属性	113	6.3	系统的启动过程分析	138
5.3.2	sticky 属性	114	6.3.1	Linux 系统的启动过程分析	138
5.4	进程管理与调度命令	114	6.3.2	Solaris 11 系统的启动过程分析	142
5.4.1	进程或作业的挂起及前/后运行切换 (fg/bg/jobs)	114	6.3.3	FreeBSD 9 系统的启动过程分析	142
5.4.2	查询进程状态 (ps)	115	6.4	服务管理	144
5.4.3	按名称终止进程或所有进程 (killall)	116	6.4.1	Linux 系统的服务管理	144
5.4.4	进程查找 (pgrep)	117	6.4.2	Fedor 15+ Linux 的服务管理	145
5.4.5	查找进程并向它们发信号 (pkill)	118	6.4.3	Solaris 11 的服务管理	150
5.4.6	确定使用指定文件或文件系统的进程 (fuser)	118	6.4.4	FreeBSD 9 的服务管理	153
5.4.7	让进程抗信号 HUP 运行 (nohup)	119	6.4.5	超级服务器 (inetd/xinetd)	154
5.4.8	设置程序运行的优先级 (nice)	119	6.5	日志管理	156
5.4.9	改变进程的优先级 (renice)	120	6.5.1	日志配置文件	156
5.5	作业和任务调度	120	6.5.2	常见日志文件	156
5.5.1	at 和 batch	120	6.5.3	文本型日志	157
5.5.2	crontab	122	6.5.4	非文本型日志	157
习题		124	6.5.5	内核启动日志	158
第 6 章	系统安装、启动与管理	125	6.5.6	日志滚动	158
6.1	系统安装	125	6.6	系统管理	158
6.1.1	安装任务与准备	125	6.6.1	系统管理的任务	158
6.1.2	系统安装和运行所需的基本分区	126	6.6.2	系统管理工具与命令	159
6.1.3	安装过程	128	6.6.3	内核参数调整	161
6.1.4	虚拟机的安装与使用	128	习题		162
6.2	引导器	128	第 7 章	软硬件管理及系统的扩充与升级	163
			7.1	软件包管理	163
			7.1.1	UNIX 的传统软件包管理	163



7.1.2	Red Hat Linux 和 Fedora 的软件包管理	163
7.1.3	FreeBSD 9 的软件包管理	166
7.1.4	Solaris 11 的软件包管理	168
7.2	系统的升级	173
7.2.1	软件包源及升级设置	173
7.2.2	软件包管理图形界面	173
7.2.3	系统升级	174
7.3	硬件管理概述及常见设备	174
7.3.1	硬件管理概述	174
7.3.2	Fedora 系统说明	175
7.3.3	FreeBSD 9 说明	175
7.3.4	Solaris 11 说明	176
7.4	打印机的管理与使用	176
7.4.1	CUPS 软件包安装与服务管理	176
7.4.2	CUPS 打印机浏览器管理	178
7.4.3	CUPS 打印机的命令界面管理与使用	180
7.4.4	打印机使用示例	183
7.5	交换区管理	184
7.5.1	Fedora 的交换区管理	184
7.5.2	FreeBSD 9 的交换设备管理	185
7.5.3	Solaris 11 的交换设备管理	186
7.6	串口与终端管理	188
7.6.1	综述	188
7.6.2	串口管理	188
7.6.3	stty 与终端设置	188
7.6.4	终端类型与能力	190
	习题	193
第 8 章	网络管理与网络应用	194
8.1	TCP/IP 概述	194
8.1.1	TCP/IP 协议体系结构	194
8.1.2	IP 地址	194
8.1.3	网络掩码	196
8.1.4	端口及服务	196
8.1.5	物理地址、逻辑地址和主机名	196
8.2	与网络有关的配置文件	197
8.2.1	/etc/hosts	197
8.2.2	/etc/services	198
8.2.3	/etc/resolv.conf	198
8.2.4	/etc/networks	199
8.3	TCP/IP 配置	199
8.3.1	TCP/IP 网络配置参数与配置方式	199
8.3.2	一个通用的 TCP/IP 网络工具 (ifconfig)	199
8.3.3	Fedora 16 的网络配置	200
8.3.4	Solaris 11 的网络配置	203
8.3.5	FreeBSD 9 的网络配置	208
8.4	网络管理通用命令	210
8.4.1	ping	210
8.4.2	netstat	211
8.4.3	arp	213
8.4.4	hostname	213
8.4.5	route	214
8.4.6	traceroute	215
8.5	网络应用常用命令介绍	216
8.5.1	telnet 与 ssh	216
8.5.2	ftp	217
8.5.3	mail / mailx	219
8.5.4	nslookup 和 host	221
8.5.5	finger	221
8.5.6	write	222
8.5.7	talk	222
8.5.8	r-命令	222
8.6	常用的网络服务	223
8.6.1	telnet 服务器	223
8.6.2	ssh 服务器	223
8.6.3	ftp 服务器	223
	习题	225



编程基础篇

第 9 章 bshell 编程	226	10.1.2 库文件	259
9.1 正则表达式简介	226	10.1.3 系统调用和库函数	260
9.1.1 字符集	226	10.1.4 基本数据类型	260
9.1.2 bshell 的模式表达式	229	10.1.5 错误处理	261
9.2 流编辑 (sed)	229	10.2 编译器	262
9.2.1 功能及用法	229	10.2.1 gcc/g++ 用法	262
9.2.2 sed 的命令	230	10.2.2 gcc/g++ 的工作过程	263
9.2.3 sed 示例	231	10.2.3 C/C++ 程序编译 示例	264
9.3 模式搜索与处理 (awk)	232	10.3 静态库及管理	265
9.3.1 功能及用法	232	10.3.1 一个多模块工程示例	265
9.3.2 记录和域	232	10.3.2 管理静态库	265
9.3.3 变量	232	10.4 共享库的管理	267
9.3.4 操作符	233	10.4.1 共享库的搜索及控制	267
9.3.5 控制语句	233	10.4.2 共享库的配置与查询	267
9.3.6 函数	234	10.4.3 共享库的构建与安装	267
9.3.7 awk 程序及执行	235	10.4.4 共享库的使用	268
9.3.8 awk 使用示例	235	10.4.5 共享库的使用示例	268
9.4 Bourne shell 及其编程	237	10.5 make 与 makefile	269
9.4.1 特殊字符	237	10.5.1 make 命令的用法简介	270
9.4.2 I/O 重定向	237	10.5.2 makefile 文件	270
9.4.3 变量与参数	239	10.5.3 makefile 示例	271
9.4.4 shell 的状态	240	10.6 调试器 gdb	273
9.4.5 shell 的调用与变量传递	241	10.6.1 gdb 功能	273
9.4.6 shell 程序设计	242	10.6.2 gdb 的基本内部命令	273
9.4.7 命令行参数与选项的 处理	251	10.6.3 程序调试示例	273
9.4.8 shell 脚本程序格式及 注意事项	253	10.7 其他编程工具简介	274
9.4.9 shell 程序调试	255	10.7.1 常用库与 GNOME/ GTK	274
9.5 bash 简介	255	10.7.2 Kdevelop/Qt 开发	274
9.5.1 bash 的特色	255	10.7.3 Java 开发	275
9.5.2 bash 部分特点介绍	255	10.7.4 Perl 开发	275
习题	258	10.7.5 数据库开发	276
第 10 章 C 编程基础和方法	259	10.7.6 PHP 开发	276
10.1 基本知识	259	习题	276
10.1.1 头文件	259		



开 发 篇

第 11 章 文件部分系统调用与标准 I/O	277
11.1 文件部分系统调用	277
11.1.1 文件打开与关闭	277
11.1.2 文件的读/写	279
11.1.3 简单 I/O 示例	279
11.1.4 文件指针及移动	281
11.1.5 文件的共享	283
11.1.6 文件的互斥与锁	285
11.1.7 fcntl	286
11.1.8 ioctl	287
11.1.9 原子操作	287
11.1.10 文件低级 I/O 示例	287
11.2 标准 I/O	290
11.2.1 流和文件对象	290
11.2.2 流的打开与关闭	290
11.2.3 标准 I/O 的缓冲区类型及更改	291
11.2.4 格式化 I/O	292
11.2.5 字符 I/O	294
11.2.6 行 I/O	294
11.2.7 二进制 I/O	295
11.2.8 流定位	295
11.2.9 错误判断及清除	296
11.2.10 获得文件描述符	296
11.3 FILE 对象与文件整数描述符	296
11.3.1 FILE 对象与文件整数描述符的关系	296
11.3.2 将文件作为标准 I/O	297
11.4 临时文件	297
11.4.1 临时文件的命名规则	297
11.4.2 临时文件的使用	297
11.4.3 临时文件示例	298
习题	300
第 12 章 UNIX 系统进程环境	301
12.1 命令行参数	301
12.1.1 C 程序的入口: main()	301
12.1.2 命令参数处理: getopt()	302
12.1.3 命令行参数处理示例	302
12.2 环境变量	304
12.2.1 环境变量表	304
12.2.2 环境变量的访问及设置	304
12.3 进程的终止	306
12.3.1 exit 和 _exit	306
12.3.2 atexit	306
12.4 主机信息	307
12.4.1 获得系统信息 (uname)	307
12.4.2 获得或设置主机名	308
12.5 用户标识信息	308
12.5.1 根据用户标识或用户名获取 passwd 信息	308
12.5.2 搜索 passwd 文件	309
12.5.3 根据组标识或组名获取组信息	309
12.5.4 搜索 group 文件	309
12.5.5 用户信息获取示例	310
12.6 用户进程标识信息	311
12.6.1 获得进程标识信息 (getpid 和 getppid)	311
12.6.2 获得用户标识信息 (getuid 和 geteuid)	311
12.6.3 获得组标识信息 (getgid 和 getegid)	311
12.6.4 用户进程标识信息示例	311
12.7 时间和日期信息	312
12.7.1 获取系统时间	312
12.7.2 显示与转换时间格式	312
12.7.3 设置时间	314
12.8 日志信息	314
12.8.1 日志记录	314
12.8.2 日志操作示例	315
习题	315
第 13 章 文件属性与目录编程	317
13.1 文件属性	317
13.1.1 文件属性查询 (stat、fstat 和 lstat)	317



13.1.2 文件权限管理(access、umask、 chmod 和 fchmod)	319
13.2 文件链接	321
13.2.1 创建硬链接 link()	321
13.2.2 创建符号链接 symlink()	321
13.2.3 读取符号链接的内容 readlink()	321
13.2.4 文件链接示例	321
13.3 文件删除	322
13.3.1 删除文件 unlink()	322
13.3.2 删除文件或空目录 remove()	323
13.3.3 文件删除示例	323
13.4 文件主、组管理	323
13.5 文件长度及截取	325
13.5.1 文件长度	325
13.5.2 文件的截取	325
13.6 更改文件的时间属性	325
13.6.1 utime()	325
13.6.2 utimes()	326
13.6.3 文件长度与时间示例	326
13.7 文件的更名与移动	328
13.8 目录编程	328
13.8.1 获得当前工作目录(getcwd 和 getwd)	328
13.8.2 目录的改变(chdir 和 fchdir)	328
13.8.3 目录的创建(mkdir)	329
13.8.4 目录的删除(rmdir)	329
13.8.5 目录的其他操作	329
13.8.6 目录操作综合示例	330
习题	333
第 14 章 进程关系与进程控制	334
14.1 进程的创建与程序的执行	334
14.1.1 进程创建系统调用 (fork 和 vfork)	334
14.1.2 exec() 系列函数	336
14.1.3 shell 程序的执行及示例	337
14.1.4 system()	339
14.1.5 更改用户的 uid 和 gid	340
14.1.6 suid、sgid 与安全程序设 计问题	340
14.2 子进程退出与进程等待	341
14.2.1 子进程退出与僵尸进程	342
14.2.2 进程等待(wait 和 waitpid)	342
14.2.3 孤儿进程	344
14.3 信号编程	344
14.3.1 信号	344
14.3.2 信号通信(kill、signal 和 raise)	345
14.3.3 进程的睡眠、闹钟与非正 常终止	347
14.3.4 信号集	348
14.4 用户登录、进程组与会话期	355
14.4.1 进程组	355
14.4.2 会话	355
14.4.3 控制终端	356
14.4.4 作业控制	356
14.5 chroot() 编程	357
习题	358
第 15 章 进程间通信	359
15.1 进程间通信简介	359
15.1.1 管道	359
15.1.2 IPC	360
15.2 管道编程	363
15.2.1 匿名管道(PIPE)	363
15.2.2 命名管道(FIFO)	366
15.2.3 将管道用作标准 I/O	369
15.2.4 进程管道 popen() 和 pclose()	370
15.3 消息队列编程	371
15.3.1 消息队列操作函数	371
15.3.2 消息队列编程示例	373
15.4 共享内存编程	375
15.4.1 共享内存操作函数	375
15.4.2 共享内存编程示例	376
15.5 信号量集编程	378
15.5.1 信号量集概述	378
15.5.2 信号量的定义及操作	378
15.5.3 UNIX 信号量操作函数	379

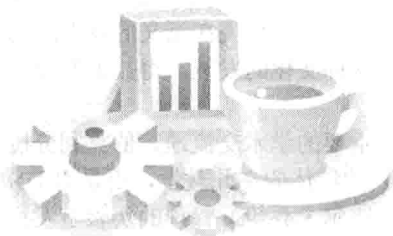


15.5.4 信号量编程示例	380
习题	383
第 16 章 线程编程基础	384
16.1 基本知识	384
16.1.1 线程基础	384
16.1.2 可重入函数与线程安全	385
16.2 线程操作基本函数及简单 示例	386
16.2.1 线程操作基本函数	386
16.2.2 线程操作简单示例	388
16.3 线程属性	390
16.3.1 线程的属性结构	390
16.3.2 属性结构的初始化与 清理	391
16.3.3 线程属性及修改	391
16.4 线程同步	392
16.4.1 用信号量实现线程间 同步	392
16.4.2 使用互斥量实现线程 间同步	395
16.4.3 使用条件变量实现线 程间同步	399
16.5 线程参数与多线程并发服 务器	402
16.5.1 线程参数	402
16.5.2 线程返回值	402
16.5.3 并发服务器框架	402
习题	403
第 17 章 终端与 curses 库编程	404
17.1 终端编程	404
17.1.1 termios 结构与终端 属性	404
17.1.2 查询和改变终端 设备属性	406
17.1.3 波特率设置	408
17.1.4 行控制函数	408
17.1.5 规范模式	408
17.1.6 非规范模式	408
17.1.7 规范模式下的密码 读取示例	409
17.2 终端数据库访问编程	411
17.2.1 相关函数	411
17.2.2 终端类型设置与性能 访问编程示例	412
17.3 curses 库编程	414
17.3.1 curses 简介	414
17.3.2 curses 程序结构	414
17.3.3 curses 库及应用	415
17.3.4 窗口与子窗体	420
17.3.5 键盘与功能键	421
习题	424
第 18 章 数据库的使用及编程	425
18.1 DBM 数据库编程	425
18.1.1 DBM 简介	425
18.1.2 库函数	426
18.1.3 DBM 使用示例	427
18.2 MySQL 数据库的使用	432
18.2.1 MySQL 的使用	432
18.2.2 MySQL 的管理	437
18.3 MySQL 数据库的编程	440
18.3.1 开发包、头文件和 库函数	440
18.3.2 MySQL C API 数据 类型	440
18.3.3 MySQL C API 函数	441
18.3.4 MySQL C API 编程 示例	447
习题	450
第 19 章 网络编程	451
19.1 网络通信基础	451
19.1.1 守候进程及工作方式	451
19.1.2 客户机/服务器 (C/S) 模型	452
19.1.3 套接字及基本类型	453
19.1.4 头文件及套接字地址 结构	453
19.1.5 与主机相关的数据 结构及函数	454
19.1.6 与服务器相关的数据 结构及函数	454
19.1.7 字节顺序及转换函数	455



19.1.8 IP 地址及主机名转 换常用函数	455	19.3.2 UDP C/S 通信模型	466
19.2 TCP 编程	457	19.3.3 UDP 示例程序	467
19.2.1 socket 编程常 用函数	457	19.4 C/M/S 结构及编程	473
19.2.2 TCP C/S 编程模型	460	19.4.1 C/M/S 结构	473
19.2.3 TCP 编程示例	461	19.4.2 交易型中间件 程序结构	473
19.2.4 TCP 应用说明	465	19.5 一个基于线程的并发 服务器程序	474
19.3 UDP 编程	466	习题	476
19.3.1 UDP 常用函数	466	参考文献	478

基础篇



第 1 章 入门知识

UNIX 是一个多用户、多任务的分时操作系统，具有稳定性好、效率高等特点，主要应用于生产领域，为人类的经济和社会发展作出了很大贡献。UNIX 系统是可移植的操作系统，在其发展过程中，一直备受计算机爱好者的青睐。UNIX 系统的发展，促进了开源和自由软件的发展。

1.1 UNIX 系统简介

1.1.1 UNIX 系统的产生及演变

UNIX 是从 Multics 的废墟中产生出来的。Multics (MULTiplexed Information and Computing System) 是 1964 年由贝尔实验室、麻省理工学院及美国通用电气公司所共同参与研发的一个用于安装在大型主机上可实现多用户、多任务的分时操作系统，目的是连接千部终端，支持 300 以上用户同时在线工作。该项目由于当时计算机软硬件水平等的限制，没有达到设计目的，于 1969 年中止。

尽管 Multics 项目垮了，但有人还在继续。当时参与该项目的贝尔实验室的 Ken Thompson 带着从 Multics 激发的灵感坚持了下来，并在一台废弃的 PDP-7 小型机上做出了“星际旅行 (Space Travel)”游戏程序。Ken Thompson 和 Dennis Ritchie 开发的程序成为了 UNIX 系统的核心，从而创造出了 UNIX。

起初 UNIX 被叫做 UNICS (UNIpIexed Information and Computing System)，这是一个与 Multics 开玩笑的双关语，意思是没有那么 Multi。

第一版 UNIX 是用汇编和 B 语言编写的。由于 B 语言在进行系统编程时不够强大，所以 Thompson 和 Ritchie 对其进行了改造，并于 1971 年共同发明了 C 语言。1973 年，Thompson 和 Ritchie 用 C 语言重写了 UNIX，使之成为了一个可移植的操作系统，为 UNIX 向其他硬件平台移植奠定了基础，使得 UNIX 适用于绝大多数计算机硬件的平台。

当时的计算机软硬件环境给 UNIX 的发展带来较大的局限性：最强大的计算机所拥有的计算能力和内存还不如现在的一部普通手机；大硬盘容量也不超过 1MB；视频显示终端才刚刚起步，六年以后才得到广泛应用；用于交互操作的设备是电传打字机，所以一般使用最短的命令名和最短的信息。比如，列目录命令 ls，它本来是 list，尽管只有 4 个字符，但也被简化为了 ls；如果一个命令执行成功了，通常不给出任何提示，而是给出一个可被查询的返回码。UNIX 命令简洁、“少说多做”的传统正是从这里开始的。

UNIX 的第一功是 1971 年为贝尔实验室的专利部门进行“文字处理”的支持工作。从此开始了



UNIX 坚持要能处理文本，现在的 UNIX 仍坚持这个原则，文档格式化、排版和通信工具的紧密结合，以文本方式编写配置文件。

1974 年，Thompson 和 Ritchie 在《美国计算机通信》(Communications of the ACM) 上发表的一篇论文 *The UNIX Time-Sharing System* 中第一次公开展示了 UNIX。文中描述了 UNIX 前所未有的简洁设计，并列举了 600 多个 UNIX 应用实例。这在当时计算机应用和普及水平均还不高的情况下，可是一个爆炸性新闻，极大地吸引了黑客们和计算机爱好者。

与 UNIX 传统的历史交织在一起的有一种隐性文化，一种传达着有关美和优秀设计的价值体系的文化，人们把这种文化称为“黑客文化”。这里的黑客(hacker)不是我们现在意义的坏人或捣乱分子，而是水平极高的、热衷于编程和计算机事业的优秀人士。这种文化的发展，得益于 UNIX 的开放源代码，没有版权和费用约束，源代码可以自由交流。

在 UNIX 的发展过程中，加州大学伯克利分校很早就成为了重要的学术交流地。伯克利分校从 1974 年就开始研究 UNIX，加之 Ken Thompson 利用 1975—1976 的年休假在此教学，更是对 UNIX 的研究注入了强劲活力。1977 年，伯克利分校毕业生 Bill Joy (比尔·乔伊) 管理的实验室发行了第一版 BSD UNIX (Berkeley Software Distribution)，从此 UNIX 走向了以 AT&T 和伯克利分校为主的开发道路，双方相互学习、相互批评，促进了 UNIX 的发展。

UNIX Sys V 和 BSD 成为了 UNIX 的两大主流，现在大部分的 UNIX 都是它们的衍生品。贝尔实验室先后在 1983 年发行了 System V 和几种微处理机上的 UNIX，1984 年发行了 System V Release2 (SVR2)；1986 年，UNIX System V 发展到了它的修订版 Res2.1 和 Res3.0；1987 年发行了 SVR3。System V 引入了许多新特征，最有代表性的就是进程间通信机制 (IPC)。BSD UNIX 的大部分特点与 UNIX System V 都差不多，但促使其迅速流行的原因是加州大学伯克利分校的计算机系统研究组 (Computer Systems Research Group, CSRG) 最早发行的一个包括页式内存管理的 UNIX 版本的一系列成就。

1980 年，国防部高级研究计划局 (Defense Advanced Research Projects Agency, DARPA) 需要在 UNIX 环境下的 VAX 机上实现 TCP/IP 协议栈，选择了具有虚存管理功能、开放源代码方面和拥有最强大的开发工具的 BSD UNIX。1983 年夏天，支持 TCP/IP 版本的 UNIX BSD 4.2 公开发布了，宣布了 TCP/IP 的诞生，这具有划时代意义。有了 TCP/IP，造就了 1994 年后互联网大爆炸，现在网络无处不在，无处不有，离开了网络人们几乎就无法生活了，这是 UNIX 对人类的重大贡献。

在 1983 年 TCP/IP 随 BSD UNIX 版发行之后，其他的网络方案也在发展，AT&T 和 Sun 都发表了网络文件系统，从而把有网络机器的文件系统结合成一个单独的大文件系统，其中 RFS (Remote File System) 和 NFS (Network File System) 至今仍在使用。

随着 UNIX 的不断成熟，功能等方面渐渐达到了商业产品的水平。许多计算机公司开始在自己的计算机平台上移植 UNIX，并出售自己的 UNIX 产品。每个公司还都在发布自己的 UNIX 时，添加上一些自己的“增值”功能。早期 Sun 公司基于 BSD 发布 Sun OS，后来又基于 SVR4 开发了 Solaris 系统。SCO 公司将 SVR3 移植到 386 上，发布了 SCO UNIX；IBM 公司发布了 AIX 操作系统；HP 公司发布了 HP-UX；Digital 公司开发了 Digital UNIX；微软也曾与 SCO 公司合作发布叫做 XENIX 的 UNIX 系统。还有，著名教授 Andrew S. Tanenbaum 为了教学开发的 MINIX 系统；我们现在众多的 Linux 系统也都基于 UNIX。商业化的 UNIX 系统直接应用于计算、管理和生产领域，给人类提供着方便、创造着价值、做着贡献。此时，UNIX 不再是黑客们的沙盘，也不再仅是研究者们的试验场。

UNIX 的商业化也带来了新的问题。为赢得用户，每个厂家都在发布或出售自己的 UNIX 产品时突出自己的个性，使自己的 UNIX 变得既像 UNIX，又不像 UNIX，操作和开发界面更是差别极大，就是专业人士也很难在不同厂家的 UNIX 轻松转行；还有，由于版权、商业机密和安全等因素，源代码也随之枯竭，开放源代码成了一句空话。

UNIX 的商业化，出现了许多不兼容问题，这为开发和操作带来了很多问题。在这种局面下，美国 IEEE (Institute of Electrical and Electronics Engineers, 美国电气及电子工程师学会) 组织成立了 POSIX (Portable Operation System Interface, 可移植操作系统接口标准) 委员会专门从事 UNIX 的标



标准化工作。POSIX 标准化仅仅基于开发层面，为用户提供一个标准的开发界面。POSIX 委员会重新描述了系统调用接口，以保证应用程序可以在源代码一级上在不同的 UNIX 间移植运行。POSIX 委员会完成了 UNIX 系统标准化后，各 UNIX 厂商又按其定义重新实现 UNIX。与 UNIX 标准化相关的组织还有 X/Open 和 FIPS 等。

UNIX 的商业化和软件的专有化造成的源代码枯竭激怒了很多热衷于开源和自由软件的人士，其中的代表人物之一就是 Richard Matthew Stallman (RMS，理查德·马修·斯托曼)。1984 年，RMS 创建了 GNU 项目 (GNU Project)。1985 年，RMS 发表了 GNU 宣言 (the GNU manifesto)，并发起了自由软件基金会 (Free Software Foundation, FSF)。RMS 将“自由软件 (free software)”这一术语大众化，于 1989 年，与律师们起草了广为使用的 GNU 通用公共协议证书 (GNU General Public License, GNU GPL)，创造性地提出了“反版权” (或“版权属左”或“开权”，copyleft) 的概念。“反版权”是一个不同于商业软件“版权所有” (Copyright) 的法律概念，既否认版权，也不反对发布软件时收取费用或取得利益。它的核心是必须把发布者的一切权利给予接受者，必须保证接受者能同时或通过其他渠道得到源程序，并将 GNU GPL 条款附加到软件的版权声明中，使接受者知道自己的权利。GNU GPL 本身也是受法律保护的版权声明。

GNU 项目是 FSF 支持的开源软件项目，其“Free as in Freedom”的哲学理念早已在国际开源社区中广为流传。GNU 项目开始于 1984 年，旨在发展一个类似 UNIX，且为自由软件完整的 GNU 操作系统。GNU 这个名字是“GNU's Not UNIX”的首字母缩写词。到现在为止，GNU 还没有开发出自己的内核，但 1991 年芬兰大学生 Linus Torvalds 在 GPL 条例下发布自己创建的 Linux 内核，操作系统被命名为 GNU/Linux (或简称 Linux)。现在各种使用 Linux 作为核心的 GNU 操作系统正在被广泛的使用，虽然它们通常被称作“Linux”。

到 20 世纪 90 年代初，GNU 项目已经开发出许多高质量的免费软件，这些软件为 Linux 操作系统的开发创造了一个合适的环境，是 Linux 能够诞生的基础之一。由于“自由”的保证，其他操作系统也同样可以使用 GNU 的软件。

GNU 操作系统也包括非 GNU 软件程序，这些程序是由其他人或工程为了他们自己的目的而开发的，之所以能用它们是因为它们是自由软件。自由软件给予使用者运行、复制、发布、研究、修改和改进该软件的自由。精确地说，自由软件赋予软件使用者以下自由：不论目的为何，有运行该软件的自由；有研究该软件如何运行，以及按需改写该软件的自由；有重新发布的自由；有改进该软件，以及向公众发布改进的自由。

说 UNIX 只是为专业人士设计的操作系统似乎有点过分，但在微软的 Windows 系统面前，对于一般用户来讲，如果说传统 UNIX 系统好用或界面友好确实有点牵强，因为黑白字符界面的可操作性确实不好，若想让 UNIX 能被一般用户所接受，还必须提供有友好、易用的操作界面的支持，这就需要提供视窗界面了。

20 世纪 80 年代，麻省理工学院 (MIT) 和 DEC 合作的 Athena 计划，合并了一个基于斯坦福大学的 W 窗口系统的窗口系统，X Window 系统可以满足这方面的需要。

1985 年，X Window 系统的创始人发布了 X Window 的源码，将其贡献给了开源世界，从而无需版权、约束和授权就可得到和使用它。这项决策的直接结果，使 X Window 成为了 UNIX 的图形引擎。现在大多 UNIX 都使用 X Window 作为自己的图形界面 (GUI)。

1.1.2 UNIX 系统的特点

1. 多任务

UNIX 是一个分时的多任务操作系统，可以同时运行很多个任务，这些任务可以是前台交互进程、后台服务器进程或批处理进程。

2. 多用户

在多任务能力基础上，UNIX 还可以同时为多个用户服务。UNIX 允许很多用户同时在不同的