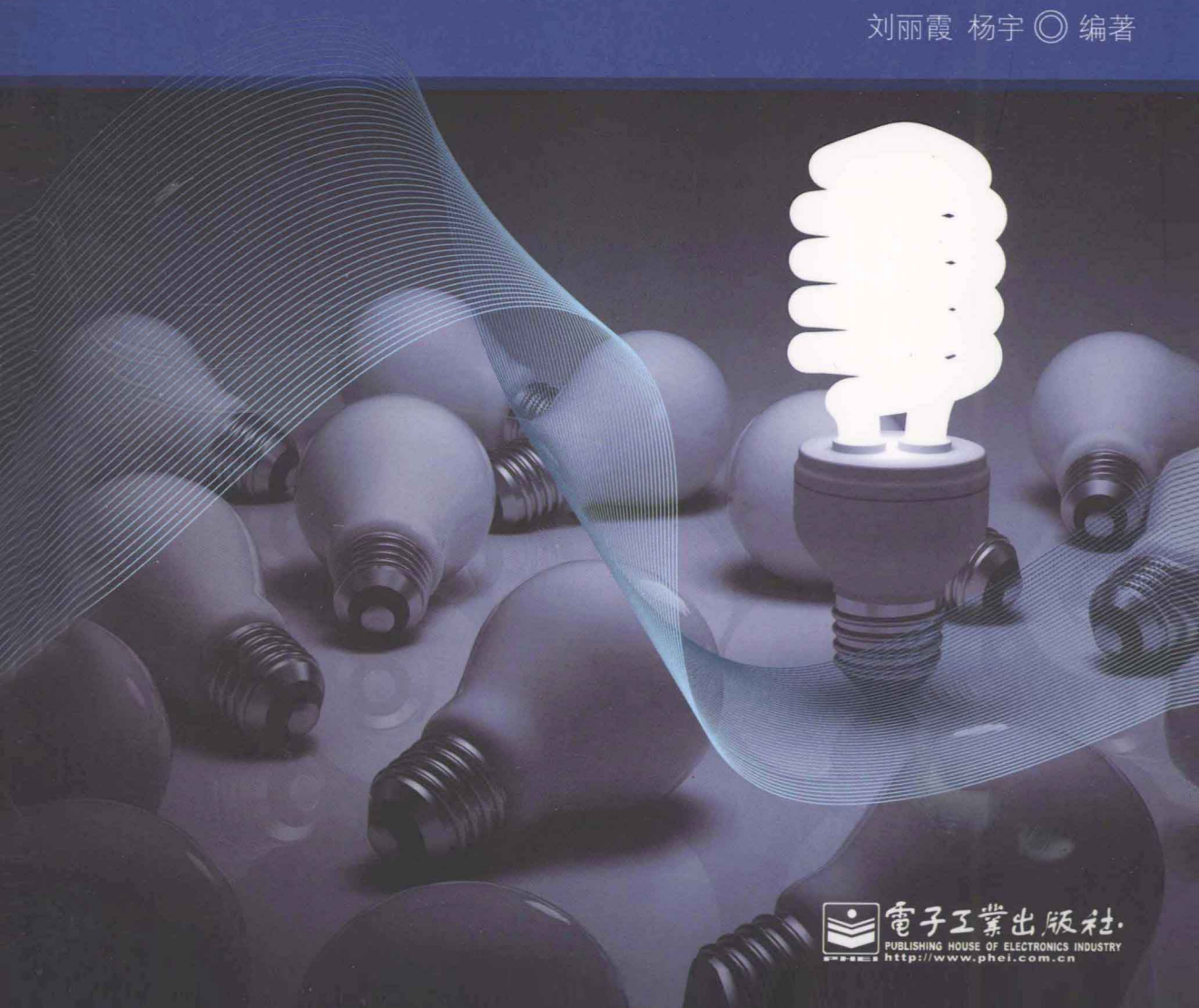


掌握Shell编程，成为真正的Linux专家

Linux Shell编程 与编辑器使用详解

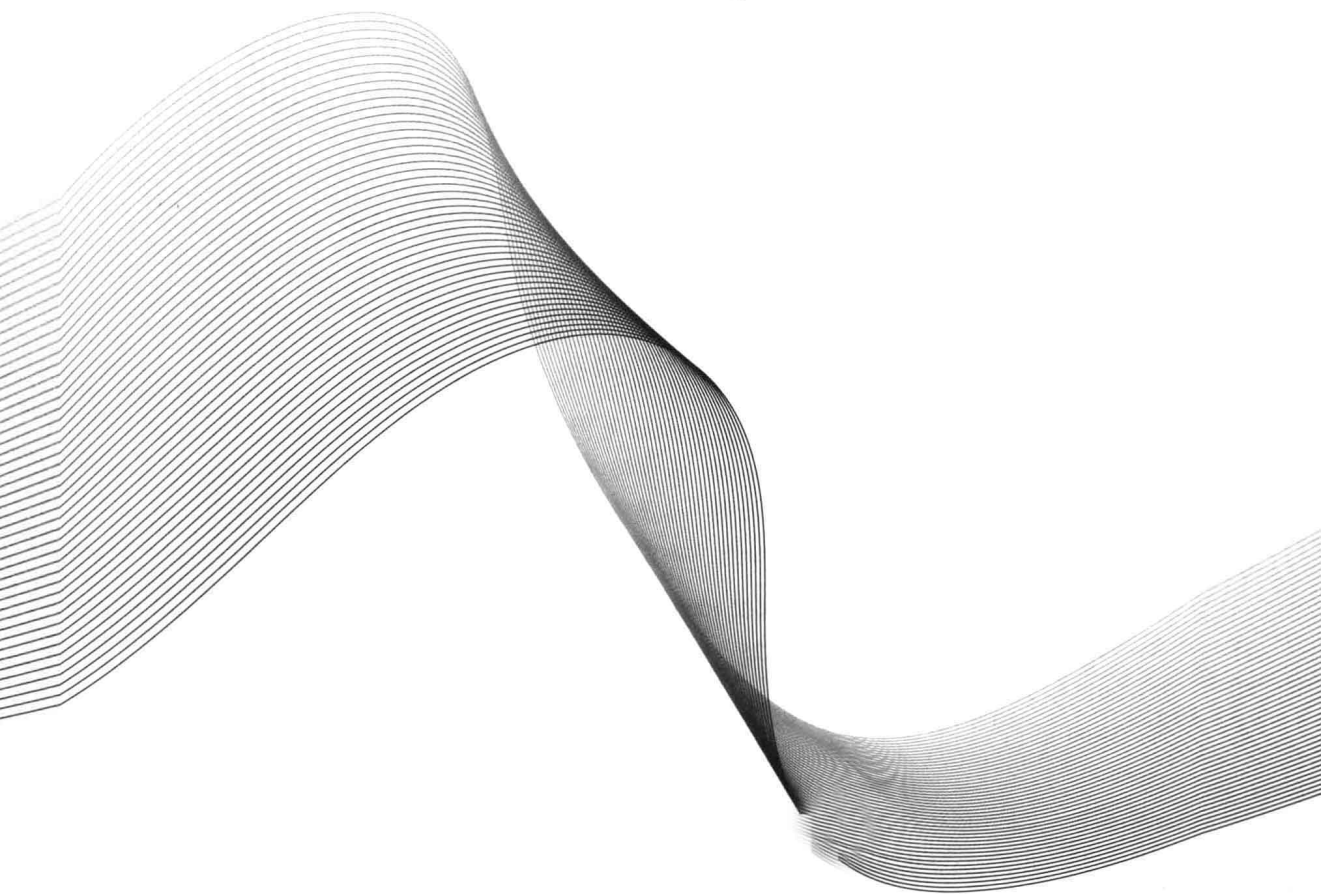
刘丽霞 杨宇 © 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Linux Shell编程 与编辑器使用详解

刘丽霞 杨宇 © 编著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书由浅入深,全面、系统地介绍了 Linux 技术,书中提供了大量实例,供读者实战演练。另外,本书有很多关于 Linux 下的命令操作内容,所以对于每个命令、每个管理设置,作者都实际运行了一遍,供读者参考,在后面的 Shell 编程中,作者详细介绍了 Shell 变量、函数、语法、控制结构等重要内容,最后一章中介绍综合实例,使读者对于 Shell 编程的理解进一步加深。

本书共分 5 篇。第 1 篇介绍 Shell 编程基础、系统管理与设置;第 2 篇介绍常用的系统管理命令、系统维护命令、正则表达式;第 3 篇介绍 vi 编辑器、emacs 编辑器;第 4 篇介绍 grep 命令、gawk 与 sed;第 5 篇介绍其他常用的文本编辑命令、Shell 变量及相关操作、条件测试、Shell 的控制结构、函数、Shell 编程实例。

本书适合 Linux 初学者、想全面学习 Linux 开发技术的人员、Linux 专业开发人员、利用 Linux 做开发的工程技术人员、Linux 开发爱好者、大中专院校的学生、社会培训班学员等。对于经常使用 Linux 做开发的人员,更是一本不可多得的案头必备参考书。

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 Shell 编程与编辑器使用详解 / 刘丽霞, 杨宇编著. — 北京: 电子工业出版社, 2013.7
ISBN 978-7-121-20717-4



I. ①L… II. ①刘… ②杨… III. ①Linux 操作系统—程序设计 IV. ①TP316.89

中国版本图书馆 CIP 数据核字 (2013) 第 131735 号

责任编辑: 付 睿

印 刷: 北京中新伟业印刷有限公司

装 订: 河北省三河市路通装订厂

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: 787×1092 1/16 印张: 26.25 字数: 739 千字

印 次: 2013 年 7 月第 1 次印刷

定 价: 59.00 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

Linux 诞生于 1991 年，由 UNIX 发展而来。Linux 是一种自由和开放源码的类 UNIX 操作系统，存在着许多不同的 Linux 版本，但它们都使用了 Linux 内核。Linux 可安装在各种计算机硬件设备中，如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型机和超级计算机。Linux 是一种领先的操作系统，世界上运算最快的 10 台超级计算机运行的都是 Linux 操作系统。几乎每天 Linux 都会以某种方式出现在我们的生活中，我们已经数不清在 Linux 上开发了多少应用程序，以及有多少机构在使用 Linux，国内外无数大型企业都在使用 Linux 系统作为服务器解决方案，而且在嵌入式领域，Linux 的应用也在不断增加。所以，对于读者来说学好 Linux 是多么重要。

如今，学习 Linux 已经成为一种潮流，而学习 Linux 命令并掌握 Shell 编程是一名 Linux 初学者成长为中、高级 Linux 用户的标志。为了满足众多 Linux 初学者、爱好者及专业人员的使用需要，作者根据自己多年来在 Linux 系统下的开发积累的经验，并融合大量的实例程序而编著成此书。本书遵循命令、编辑器、Shell 编程这一知识脉络组织各章节内容，各知识点前后贯穿、循序渐进、依次进阶。在 Linux 命令的讲解中融入了基础知识点的解析及相关应用示例，尽量避免手册、指南式的枯燥讲解；在 Shell 编程部分通过对代码的逐行剖析及执行范例的讲评，使读者能够深入掌握 Shell 编辑技巧，达到举一反三、触类旁通的目的。本书由浅入深，适合各个水平阶段的读者学习。

本书特色

1. 配合大量实际例子运行操作，学习效果好

作者为了使读者易于理解书中内容，对于每个命令、每个函数、每个实际操作，都在 Linux 系统下实际运行了一遍，供读者参考，使读者易于掌握知识。

2. 内容全面、系统、深入

本书系统地介绍了 Linux 的基础知识，Linux 系统下的常用设置、维护与管理、正则表达式、常用编辑器的使用、文件的相关操作、Shell 编程，并结合实际列举出一些综合的实例供读者参考。

3. 讲解由浅入深、循序渐进，适合各个层次的读者阅读

本书从 Linux 的基础开始讲解，逐步深入到 Linux 系统的设置、维护与管理、命令行操作、Shell 语法、Shell 程序的编写，内容梯度从易到难，讲解由浅入深、循序渐进，适合各个层次的读者阅读，并均有所获。

4. 贯穿大量的开发实例和技巧，迅速提升工作效率

本书在讲解知识点时贯穿了大量短小精悍的典型实例，并给出了大量的开发技巧，以便让读者更好地理解各种概念和管理维护技术，体验实际编程，迅速提高管理维护水平。

5. 详解典型项目案例开发，提高实战水平

本书详细介绍了 Shell 编程的十几个实际案例，以提高读者的 Shell 编程水平，从而具备独



立进行脚本编写的能力。

6. 提供技术支持，答疑解惑

读者阅读本书时若有任何疑问，可发 E-mail 到 bookservice2012@126.com 获得帮助，也可以在 CSDN 的技术论坛 (<http://bbs.csdn.net/forums/Linux>) 上留言，会有专人负责答疑。

本书内容及体系结构

第 1 篇 Linux 系统学习基础（第 1、2 章）

本篇主要内容包括：Shell 基本概念，常用的 Shell 类型，Shell 基本语法，系统管理与设置，常用的信息显示命令，系统常用信息设置，进程资源，终端命令行操作。通过本篇读者可轻松掌握 Linux 系统基础知识，熟悉命令行的操作。

第 2 篇 系统维护与管理（第 3~5 章）

本篇主要内容包括：常用的管理命令，进程任务管理，设置系统的各种任务，后台任务管理，常用的系统维护命令，系统的日常配置，自动作业处理，系统日志，正则表达式。通过本篇的学习，读者可更进一步加深对于 Linux 系统的掌握，满足日常工作的需求，提高工作效率。

第 3 篇 vi 编辑器与 emacs 编辑器（第 6、7 章）

本篇主要内容包括：vi (vim) 编辑器的概述，vi (vim) 中的常用命令，vi (vim) 中的字符与文件操作，窗口操作，emacs 概述，文本编辑，缓存区与窗口，emacs 扩展工具。通过本篇读者可以掌握 Linux 系统下最常用的文本编辑器，对于一门工具的熟练掌握，可以大大提高编程效率，工作效率。

第 4 篇 grep 命令，gawk 与 sed，其他常用文本命令（第 8、9 章）

本篇主要内容包括：grep 介绍，grep 与正则表达式，egrep 与 fgrep，过滤器 gawk，gawk 的编程设计，sed 流编辑器，文本编辑常用命令，文件属性比较，文本标准输出与输入，文本内容比较等。通过本篇读者可以掌握操纵数据和产生报告的程序语言，命令行工具，文本常用的基本操作等知识。

第 5 篇 Shell 编程及编程实战（第 10~15 章）

本篇主要内容包括：Shell 变量类型，变量的操作，变量的赋值，算术运算，Shell 的测试命令，测试数值，测试字符串，测试逻辑运算符，Shell 的控制结构，函数的定义，函数的调用，函数的嵌套与递归，文件类型编程，编写选择菜单，文档目录编程，创建用户编程，Shell 算法编程。通过本篇的学习读者可以熟练掌握 Shell 编程，使 Linux 系统维护管理达到自动化；读者还可以从这十几个综合实例中掌握各种类型的 Shell 编程，使读者在工作中 Shell 编程运用自如，大大提高工作效率。

本书读者对象

- Linux 初学者；
- 想全面学习 Linux 开发技术的人员；
- Linux 专业开发人员；
- 利用 Linux 做开发的工程技术人员；
- Linux 开发爱好者；
- 大中专院校的学生；
- 社会培训班学员；
- 需要一本案头必备手册的程序员。

目 录

第 1 篇 Linux 系统学习基础

第 1 章 Shell 编程基础	1	2.2 系统信息显示命令	21
1.1 基本概念	1	2.2.1 proc 文件系统: 访问系统 内核信息	21
1.1.1 Shell 的功能	1	2.2.2 modinfo 命令: 内核 模块信息	23
1.1.2 Linux Shell 的常用命令	1	2.2.3 cat /etc/issue 命令: 系统的 当前发行版本	23
1.1.3 启动和退出	2	2.2.4 cat /proc/version: 查看系统 内核当前版本	24
1.2 常用的 Shell 类型	3	2.2.5 cat /proc/partitions: 查看系统 当前分区	24
1.2.1 Linux 默认 Shell	3	2.2.6 cat /proc/interrupts 命令: 显示中断信息	24
1.2.2 Bourne Shell 的扩展版	3	2.2.7 cat /proc/filesystems 命令: 常用文件系统信息	24
1.2.3 网关接口 Shell	3	2.2.8 dmesg: 启动信息查询	25
1.2.4 早期 Shell	3	2.3 系统常用信息设置	26
1.2.5 支持后台作业 Shell	4	2.3.1 date 命令: 系统时间显示与 设置	26
1.3 Shell 基本语法	4	2.3.2 hostname 命令: 当前系统的 主机名显示与设置	28
1.3.1 Shell 脚本中的注释	4	2.3.3 hwclock 命令: 硬件时钟 显示和设置	29
1.3.2 Shell 通配符的使用	4	2.3.4 hash 命令: 显示、添加或 清除哈希表	30
1.3.3 大小写区分	6	2.3.5 sysctl 命令: 系统核心参数 的设置	31
1.3.4 指定使用某个 Shell	6	2.4 文件系统信息显示常用命令	32
1.3.5 IO 重定向	7	2.4.1 echo 命令: 文本行输出 显示	32
1.3.6 管道 (Pipe)	10	2.4.2 isosize 命令: iso9660 文件 系统大小的显示	33
1.3.7 前台与后台执行任务	11	2.4.3 minfo 命令: ms-dos 文件 系统的各项参数的显示	34
1.3.8 命令的执行顺序	12		
1.3.9 Shell 中 3 种引号的用法	13		
1.3.10 运行脚本	13		
第 2 章 系统管理与设置	16		
2.1 常用的信息显示命令	16		
2.1.1 cal 命令: 日期显示	16		
2.1.2 cat /proc/meminfo: 内存 使用情况查询	17		
2.1.3 free 命令: 内存详细 信息查询	18		
2.1.4 arch 命令: CPU 架构显示	19		
2.1.5 cat /proc/cpuinfo 命令: 处理器 CPU 信息显示	19		
2.1.6 cat /proc/modules: 查询当前 系统被加载模块	20		



2.4.4	pwd 命令：当前工作目录的显示.....	34	2.6.1	top 命令：进程信息显示.....	43
2.4.5	procinfo 命令：系统状态信息的显示.....	34	2.6.2	ipcs 命令：IPC 设备信息显示.....	44
2.4.6	stat 命令：文件/文件系统的详细信息的显示.....	37	2.6.3	time 命令：进程执行时间显示.....	45
2.5	Linux 操作系统信息查询.....	38	2.6.4	pstree 命令：以树状图显示程序.....	45
2.5.1	uname 命令：操作系统信息的显示.....	38	2.6.5	ps 命令：监测进程.....	46
2.5.2	vmstat 命令：虚拟内存的统计信息的显示.....	39	2.6.6	pmap 命令：显示程序的占用资源信息.....	49
2.5.3	uptime 命令：系统的运行信息的显示.....	39	2.6.7	jobs 命令：后台作业.....	49
2.5.4	locale 命令：当前系统的语言设置.....	40	2.6.8	mev 命令：监视鼠标情况.....	50
2.5.5	tload 命令：系统负载显示.....	40	2.6.9	fuser 命令：用套接口或者文件识别进程.....	51
2.5.6	runlevel 命令：系统运行级别显示.....	41	2.7	Linux 终端命令行操作.....	52
2.5.7	lsmod 命令：所有已加载模块的显示.....	42	2.7.1	fgconsole 命令：显示当前虚拟终端数目.....	52
2.6	进程资源.....	43	2.7.2	history 命令：显示历史指令.....	52
			2.7.3	gdialog 命令：Shell 脚本创建对话框.....	53
			2.7.4	yes 命令：字符串的连续输出.....	59

第 2 篇 系统维护与管理

第 3 章	常用的系统管理命令.....	61	3.2.2	fg 命令：后台任务放前台.....	67
3.1	常用的系统管理任务.....	61	3.2.3	kill 命令：结束进程.....	68
3.1.1	apm 命令：BIOS 电源管理.....	61	3.2.4	killall 命令：杀死同名进程.....	70
3.1.2	apmsleep 命令：低功耗工作.....	62	3.2.5	nice 命令：进程优先级切换.....	71
3.1.3	open 命令：开启虚拟终端.....	62	3.2.6	pgrep 命令：查找符合条件的进程.....	71
3.1.4	vlock 命令：锁定终端.....	63	3.2.7	pidof 命令：查看正在运行程序的进程 ID.....	73
3.1.5	sudo 命令：用户切换.....	63	3.2.8	renice 命令：设定进程优先级.....	73
3.1.6	rsh 命令：远程用户设置登录 Shell.....	64	3.2.9	pgrep 命令：查找符合条件的进程.....	71
3.1.7	chsh 命令：切换系统登录时 Shell.....	65	3.3	设置系统的各种任务.....	75
3.1.8	clear 命令：清除终端屏幕.....	66	3.3.1	ntsysv 命令：图形化设置系统的各种服务.....	75
3.1.9	rlogin 命令：智能化远端登录.....	66	3.3.2	symlinks 命令：管理和维护符号链接工具.....	76
3.2	进程任务管理.....	67	3.3.3	mttools：软盘操作工具集.....	77
3.2.1	apmd 命令：高级电源管理守护进程.....	67			



3.3.4	gcov 命令: coverage 测试工具.....	78	4.1.10	dircolors 命令: 配置 ls 命令 显示颜色.....	106
3.3.5	screen 命令: 视窗程序 管理工具.....	79	4.1.11	set 命令: 设置 Shell 执行方式.....	111
3.3.6	logrotate 命令: 管理 log 文件.....	81	4.1.12	telinit 命令: 设置系统的 运行等级.....	113
3.3.7	pwck 命令: 检查密码文件.....	82	4.2	系统的日常配置.....	113
3.3.8	exec 命令: 交出控制权.....	83	4.2.1	declare 命令: Shell 变量.....	113
3.3.9	&命令: 后台执行任务.....	83	4.2.2	fbset 命令: 帧缓存 设备设置.....	115
3.3.10	init 命令: 进程的父进程.....	83	4.2.3	insmod 命令: 加载模块.....	117
3.4	后台任务管理.....	84	4.2.4	lilo 命令: 安装启动 装载程序.....	119
3.4.1	bg 命令: 后台执行程序.....	84	4.2.5	modprobe 命令: 自动安装 依赖模块.....	119
3.4.2	wait 命令: 等待程序 返回状态.....	84	4.2.6	mkraid 命令: raid 磁盘 阵列.....	120
3.4.3	depmod 命令: 显示模块 依赖关系.....	84	4.2.7	losetup 命令: 设置和控制 循环设备.....	121
3.4.4	rmmod 命令: 删除模块.....	85	4.2.8	rpm 命令: RPM 包管理.....	121
3.4.5	ipcrm 命令: 进程间 机制管理.....	86	4.2.9	ldd 命令: 共享库文件的 相互依赖关系.....	125
3.4.6	watch 命令: 重复执行命令.....	87	4.2.10	ldconfig 命令: 配置动态 链接绑定.....	126
3.4.7	fc 命令: 修改执行命令.....	88	4.2.11	up2date 命令: 升级 Red Hat Linux 系统.....	129
3.4.8	nohup 命令: 退出登录 执行命令.....	89	4.3	其他常用命令.....	131
3.4.9	suspend 命令: 暂停执行 Shell.....	89	4.3.1	false 命令: false 返回值.....	131
3.4.10	sleep 命令: 系统休眠.....	89	4.3.2	true 命令: true 返回值.....	132
第 4 章	系统维护命令.....	91	4.3.3	ulimit 命令: 控制 Shell 及 进程的资源.....	132
4.1	常用的系统维护命令.....	91	4.3.4	mouseconfig 命令: 设置 鼠标相关参数.....	134
4.1.1	bind 命令: 设置按键组合.....	91	4.3.5	setup 命令: 系统公用 程序设置.....	135
4.1.2	enable 命令: Shell 内置 命令.....	92	4.3.6	swapon 命令: 卸载交换区 (swap area).....	138
4.1.3	aumix 命令: 音效装置设置.....	94	4.3.7	swapon 命令: 加载交换区 (swap area).....	139
4.1.4	配置系统的认证资源.....	96	4.3.8	resize 命令: 调整终端 视窗.....	140
4.1.5	export 命令: 设置环境变量.....	98			
4.1.6	hostid 命令: 主机数字标识.....	103			
4.1.7	alias 命令: 设置命令的 别名.....	103			
4.1.8	chkconfig 命令: 管理系统 服务.....	104			
4.1.9	eval 命令: 多个命令 同时执行.....	106			



4.3.9	setserial 命令：串口设备设置	140
4.3.10	unalias 命令：删除指定的别名	141
4.3.11	unset 命令：删除变量或函数	141
4.4	自动作业处理	142
4.4.1	anacron 命令：调度守护进程任务	142
4.4.2	at 命令：多任务执行排队处理	143
4.4.3	atd 命令：执行已经排队的任务	145
4.4.4	atq 命令：显示排队的任务	145
4.4.5	atrm 命令：删除已经排队的任务	146
4.4.6	atrun 命令：执行排好队的任务	146
4.4.7	crontab 命令：定时执行程序	146
4.4.8	batch 命令：批处理	149
4.5	系统日志	150
4.5.1	syslogd 命令：启动 syslog 日志进程	150
4.5.2	syslog.conf：系统日志配置文件	150

4.5.3	logger 命令：测试 syslog.conf	153
4.5.4	echo 清空日志	153

第 5 章 正则表达式

5.1	正则表达式介绍	154
5.2	元字符	154
5.2.1	元字符的定义	154
5.2.2	元字符的应用示例	155
5.3	字符集和类	156
5.3.1	字符集和类的定义	156
5.3.2	元字符集和类的应用示例	157
5.4	重复	157
5.4.1	重复的定义	157
5.4.2	重复的应用示例	158
5.5	子表达式	158
5.5.1	子表达式的定义	158
5.5.2	子表达式的应用示例	159
5.6	定位字符串的开始和末尾	159
5.6.1	开始和末尾的定义	159
5.6.2	定位字符串的开始和末尾的应用示例	160
5.7	分支	161
5.7.1	分支的定义	161
5.7.2	分支的应用示例	161
5.8	匹配特殊字符	161

第 3 篇 vi 编辑器与 emacs 编辑器

第 6 章 vi (vim) 编辑器

6.1	vi (vim) 编辑器概述	163
6.1.1	vi (vim) 编辑器的发展史	163
6.1.2	启动 vi (vim)	164
6.1.3	退出 vi (vim)	165
6.1.4	命令模式	166
6.1.5	末行模式	166
6.1.6	插入模式	166
6.2	vi (vim) 中的常用命令	167
6.2.1	插入文本命令	167
6.2.2	附加文本命令	167
6.2.3	光标移动命令	167
6.2.4	字符与块删除命令	168

6.2.5	添加新行命令	169
6.2.6	修改文本命令	169
6.2.7	行合并命令	170
6.2.8	文本替换命令	170
6.2.9	句点命令	171
6.2.10	撤销修改命令	171
6.2.11	复制文本命令	171
6.2.12	粘贴文本命令	171
6.2.13	重复执行命令	172
6.3	vi (vim) 中的字符与文件操作	172
6.3.1	查找指定字符命令	172
6.3.2	普通与增量查找	172
6.3.3	替换指定字符串	172



6.3.4	更改大小写命令	173	7.1.1	emacs 发展史	181
6.3.5	文件读入命令	174	7.1.2	启动和退出	182
6.3.6	文件写入命令	174	7.1.3	新建文本	183
6.3.7	定位到指定行命令	174	7.1.4	保存文本	184
6.3.8	显示状态信息命令	175	7.2	文本编辑	185
6.3.9	保存和退出命令	175	7.2.1	常用命令键	185
6.4	窗口操作	175	7.2.2	光标的移动	185
6.4.1	窗口操作的快捷方式	175	7.2.3	文本的输入	186
6.4.2	窗口垂直拆分	176	7.2.4	文本的删除	187
6.4.3	窗口水平拆分	176	7.2.5	字符的查找和替换	187
6.4.4	关闭窗口	177	7.3	缓冲区和窗口	188
6.4.5	屏幕重绘	177	7.3.1	水平与垂直拆分窗口	188
6.4.6	屏幕滚动	178	7.3.2	多个缓冲区同时 进行工作	189
6.4.7	文本定位	178	7.3.3	缓冲区中内容的滚动	190
6.4.8	交互式 Shell	178	7.4	emacs 的扩展工具	190
6.4.9	在 vi (vim) 中执行 Shell	179	7.4.1	日历显示	190
6.4.10	获得帮助	179	7.4.2	执行 Shell 命令	190
第 7 章	emacs 编辑器	181	7.4.3	游戏调用	191
7.1	emacs 概述	181			

第 4 篇 grep 命令、gawk 与 sed, 其他常用文本命令

第 8 章	grep 命令	192	9.1.4	gawk 注释	211
8.1	grep 介绍	192	9.1.5	变量赋值	211
8.1.1	grep 的功能介绍	192	9.1.6	模式的使用	212
8.1.2	grep 的工作原理	192	9.1.7	指定动作	214
8.2	grep 与正则表达式	193	9.1.8	使用重定向输出	214
8.2.1	grep 正则表达式的 元字符集	193	9.1.9	字段分隔符输出	215
8.2.2	grep 正则实例	196	9.1.10	记录与字段的介绍	215
8.2.3	grep 的常用参数选项	198	9.2	gawk 的编程设计	216
8.2.4	grep 与管道的配合使用	202	9.2.1	函数	216
8.2.5	grep 的退出状态	203	9.2.2	关联数组	217
8.3	egrep 与 fgrep	203	9.2.3	格式化输出命令	218
8.3.1	egrep 处理复杂字符串	203	9.2.4	关系运算符	219
8.3.2	fgrep 处理简单字符串	206	9.2.5	算术运算符	220
第 9 章	gawk 与 sed	207	9.2.6	流程控制语句	222
9.1	过滤器 gawk	207	9.2.7	getline: 控制输入	224
9.1.1	gawk 发展史	207	9.2.8	协进程	225
9.1.2	gawk 的程序处理模式	209	9.2.9	网络数据交换	226
9.1.3	gawk 的语法格式	209	9.2.10	gawk 的应用示例	227
			9.3	流编辑器 sed	228
			9.3.1	sed 编辑器介绍	229



9.3.2	sed 的基本语法	229
9.3.3	地址的作用	230
9.3.4	指令的作用	231
9.3.5	删除指令	232
9.3.6	追加指令	232
9.3.7	打印指令	233
9.3.8	修改指令	233
9.3.9	替换指令	234
9.3.10	插入指令	236
9.3.11	退出指令	236
9.3.12	下一条指令	237
9.3.13	文件写入指令	237
9.3.14	文件读入指令	238
9.3.15	sed 的控制结构	238
9.3.16	Pattern 区和 Hold 区	239
9.3.17	sed 的应用示例	241

第 5 篇 Shell 编程及编程实战

第 10 章	其他常用的文本编辑命令	243
10.1	文本编辑常用命令	243
10.1.1	cat 命令：标准输出设备 显示文本内容	243
10.1.2	ed 命令：行文本编辑器	245
10.1.3	fmt 命令：重排文本文件	246
10.1.4	gedit 命令：图形化文本 编辑器	247
10.1.5	ed 命令：行文本编辑器	248
10.1.6	jed 命令：jed 编辑文本 文件	249
10.1.7	cut 命令：文件剪切	250
10.1.8	colrm 命令：删除 指定的列	252
10.1.9	comm 命令：对已排序文件 逐行进行比较	253
10.1.10	od 命令：格式化输出 文件内容	254
10.1.11	joe 命令：ASCII 文本 编辑器	255
10.1.12	aspell 命令：检查文件的 拼写错误	258
10.1.13	col 命令：过滤控制 字符	260
10.1.14	view 命令：只读方式 查看文件内容	260
10.1.15	more 命令：分屏显示 文本内容	261
10.1.16	head 命令：部分显示 文本内容	262
10.2	其他常用的文本命令与技巧	264
10.2.1	cmp 命令：文件属性 比较	264
10.2.2	diff 命令：文件内容 比较	265
10.2.3	diffstat 命令：diff 结果的 统计信息	268
10.2.4	patch 命令：文件打补丁	269
10.2.5	csplit 命令：按指定行数 分割文件	270
10.2.6	uniq 命令：显示文件中 重复出现的行	271
10.2.7	pr 命令：编排文件格式	272
10.2.8	pico 命令：编辑文字 文件	273
10.2.9	tail 命令：读取文件的 末尾部分	275
10.2.10	tac 命令：反序显示 文本内容	277
10.2.11	less 命令：前后翻阅 文本内容	278
10.2.12	sort 命令：文件排序	279
10.2.13	indent 命令：调整 c 代码 文件格式	281
10.2.14	tr 命令：转换文本指定 字符串	285
10.2.15	join 命令：连接两个相同 字段文件	286
10.2.16	spell 命令：检查文本 拼写错误	287

10.2.17 tee 命令: 读取标准输入到 标准输出或文件.....	287	12.2.5 测试文件存在 且是块设备.....	310
10.2.18 wc 命令: 统计文本信息	288	12.3 测试数值	310
第 11 章 Shell 变量及相关操作	290	12.3.1 相等测试	310
11.1 Shell 变量类型	290	12.3.2 不相等测试	311
11.1.1 Shell 环境变量	290	12.3.3 小于测试	312
11.1.2 位置变量	292	12.3.4 小于等于测试	312
11.1.3 预定义的特殊变量	293	12.3.5 大于测试	313
11.1.4 用户自定义的变量	294	12.3.6 大于等于测试	313
11.2 变量的操作	294	12.4 测试字符串	314
11.2.1 创建和设置变量	294	12.4.1 相等测试	314
11.2.2 区分变量名	295	12.4.2 不相等测试	315
11.2.3 删除变量	295	12.4.3 非空测试	316
11.3 变量的赋值	296	12.4.4 长度为 0 测试	316
11.3.1 直接给变量赋值	296	12.4.5 长度不为 0 测试	317
11.3.2 使用 read 命令赋值	296	12.5 测试逻辑运算符	317
11.3.3 命令行参数赋值	298	12.5.1 逻辑与测试	318
11.3.4 命令的输出结果赋值	298	12.5.2 逻辑或测试	319
11.3.5 文件读入数据实现赋值	299	12.5.3 逻辑非测试	319
11.4 变量的输出	299	第 13 章 Shell 的控制结构	321
11.4.1 printf 命令: 格式化 输出变量	300	13.1 if 结构	321
11.4.2 echo 命令: 直接输出 变量	301	13.1.1 if 结构的语法格式	321
11.5 数组变量	301	13.1.2 if...then...fi 示例	322
11.5.1 数组的定义	301	13.1.3 if...then...else...fi 示例	323
11.5.2 数组的复制	303	13.1.4 if...then...elif...示例	324
11.5.3 数组元素的个数	303	13.1.5 if 嵌套	326
11.5.4 数组元素的长度	304	13.2 case 结构	328
11.6 算术运算	305	13.2.1 case 结构的语法格式	328
11.6.1 expr 命令	305	13.2.2 case 结构示例	328
11.6.2 let 命令	305	13.2.3 case 嵌套	334
11.6.3 浮点算术运算	306	13.3 select 结构	335
第 12 章 条件测试	307	13.3.1 select 结构的语法格式	335
12.1 Shell 的测试命令	307	13.3.2 select 结构示例	335
12.2 测试文件属性	307	13.4 while 循环	337
12.2.1 测试文件存在	307	13.4.1 while 语句的语法格式	337
12.2.2 测试文件存在且可执行	308	13.4.2 while 循环示例	338
12.2.3 测试文件存在且是目录	309	13.4.3 while 的嵌套	339
12.2.4 测试文件存在且是 字符设备	309	13.4.4 避免死循环	340
		13.5 for 循环	341
		13.5.1 for 循环语法结构	341
		13.5.2 for 循环示例	342



13.6	until 循环	344	15.7	自动下载指定文档	376
13.6.1	until 循环的语法结构	344	15.7.1	系统设计	376
13.6.2	until 循环示例	345	15.7.2	代码实现	377
13.7	break 和 continue 语句	346	15.8	自动上传指定文档	377
13.7.1	break 语句	346	15.8.1	系统设计	377
13.7.2	continue 语句	348	15.8.2	代码实现	377
第 14 章	函数	352	15.9	通过命令连续创建 50 个 用户账号	379
14.1	函数的定义	352	15.9.1	系统设计	379
14.2	函数的调用	352	15.9.2	代码实现	379
14.2.1	执行函数	353	15.10	通过操作文件创建多个 用户	380
14.2.2	函数参数传递	354	15.10.1	系统设计	380
14.2.3	函数的返回值	356	15.10.2	代码实现	381
14.2.4	载入函数	358	15.11	连续创建多个目录	383
14.2.5	删除函数	359	15.11.1	系统设计	383
14.3	函数的作用域	360	15.11.2	代码实现	383
14.3.1	全局作用域	360	15.12	检查系统资源使用情况	384
14.3.2	局部作用域	361	15.12.1	系统设计	384
14.3.3	应用示例	362	15.12.2	代码实现	385
14.4	函数的嵌套	363	15.13	账户的一致性检查	388
14.5	函数的递归	364	15.13.1	系统设计	388
第 15 章	Shell 编程实例	367	15.13.2	代码实现	388
15.1	判断文件类型	367	15.14	文件类型转换	391
15.1.1	系统设计	367	15.14.1	系统设计	391
15.1.2	代码实现	367	15.14.2	代码实现	391
15.2	使用 if 语句编写选择菜单	367	15.15	对输入的 3 个数字进行 排序	393
15.2.1	系统设计	368	15.15.1	系统设计	393
15.2.2	代码实现	368	15.15.2	代码实现	393
15.3	使用 while 循环编写 选择菜单	369	15.16	计算奇数之和	395
15.3.1	系统设计	369	15.16.1	系统设计	396
15.3.2	代码实现	369	15.16.2	代码实现	396
15.4	使用 until 循环编写 选择菜单	371	15.17	使用 select 命令生成菜单	397
15.4.1	系统设计	371	15.17.1	系统设计	397
15.4.2	代码实现	371	15.17.2	代码实现	397
15.5	对指定目录定时备份	374	15.18	打印图形	398
15.5.1	系统设计	374	15.18.1	系统设计	398
15.5.2	代码实现	374	15.18.2	代码实现	398
15.6	连续对多个目录逐个备份	375	15.19	处理压缩文件	399
15.6.1	系统设计	375			
15.6.2	代码实现	375			



15.19.1 系统设计.....	399	15.22.2 代码实现.....	403
15.19.2 代码实现.....	399	15.23 使用 gawk 反序输出各列.....	403
15.20 数值进制转换.....	401	15.23.1 系统设计.....	403
15.20.1 系统设计.....	401	15.23.2 代码实现.....	403
15.20.2 代码实现.....	401	15.24 对文件进行大小写转换.....	404
15.21 使用 sed 打印指定行.....	402	15.24.1 系统设计.....	404
15.21.1 系统设计.....	402	15.24.2 代码实现.....	404
15.21.2 代码实现.....	402	15.25 统计文件中的文本信息.....	405
15.22 使用 sed 筛选普通文件.....	402	15.25.1 系统设计.....	405
15.22.1 系统设计.....	402	15.25.2 代码实现.....	405

第 1 篇 Linux 系统学习基础

第 1 章 Shell 编程基础

Shell 是操作系统的外壳，是用户使用操作系统的命令接口，是一个介于用户和系统内核之间的桥梁。Shell 接受用户以命令行方式输入的命令（包括系统提供的内部命令、独立存在于某个目录下的程序），同时也能解释执行由 Shell 命令编写的 Shell 程序。

1.1 基本概念

本节主要让读者了解什么是 Shell，及其具有的功能，只有从本质上理解它的功能，才能在后面的章节中熟练应用这个工具。

1.1.1 Shell 的功能

在计算机科学中，Shell 俗称壳（用来区别于核），是指“提供使用者使用界面”的软件（命令解析器）。它类似于 DOS 下的 `command.com`。它接收用户命令，然后调用相应的应用程序。同时它又是一种程序设计语言。作为命令语言，它交互式解释和执行用户输入的命令，或者自动解释和执行预先设定好的一连串的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高阶语言中才具有的控制结构，包括循环和分支。Shell 的功能主要包括以下几方面。

- 接受来自键盘的命令。
- 检查命令的正确性。
- 命令错误则给出报错信息。
- 命令正确则使用相应的系统调用执行命令（产生进程）。
- 命令执行完毕，给出命令执行的结果。

1.1.2 Linux Shell 的常用命令

用户可以将 Linux 的命令以编程的方式组合起来提交给 Shell，Shell 负责解释执行这种程序设计语言所编写的代码。Linux 把由这种程序设计语言构成的文件称为 Shell Script 或 Shell 脚本。

Shell 脚本是包含一个或多个 Shell 命令的集合，这些命令会按照编写的顺序被依次执行，Shell 提供了一种利用简单命令组合实现复杂功能的机制。Shell 脚本中支持绝大多数编程语言中所能见到的程序控制结构，可以使用注释（#）对脚本的功能进行说明，可以使用条件测试表达式进行分支判断，可以使用循环语句处理大量数据，可以定义使用变量、数组等，并且也提供对函数的支持。

Shell 脚本是解释执行而不是编译执行。脚本由 ASCII 文本组成，由 Shell 命令解释器读取并执行。当执行一个 Shell 脚本时，命令解释器会逐行、逐个循环地读取所有的 ASCII 文本，并从头到尾依次执行每条语句。

Shell 编程的实质是连续自动执行一系列命令，表 1-1 中列出 Shell 编程经常使用的命令及相关约定。



表 1-1 Shell 编程常用命令

命 令	说 明	命 令	说 明
pwd	显示当前目录	&&	逻辑与
passwd	修改用户的密码	&	在后台执行
ls	列出目录中的内容	date	显示当前的日期和时间
cd	改变当前的工作目录	echo	回显，即将字符串输出到标准输出设备
*	匹配任意长度的任意字符	sleep	暂停指定的时间间隔
?	匹配任意一个字符	wc	计算文件的单词数、行数、字符数
[]	任意属于字符组的字符	head	查看文件的开始部分，默认为前 10 行
cat	一次性显示输出文件的全部内容	tail	查看文件的结尾部分，默认为后 10 行
file	显示文件的类型	diff	比较两个文件的差异
more	分屏显示文件的内容	spell	拼写检查
clear	清屏，提示符回到屏幕左上方	lp	打印文件
cp	复制文件	cal	显示日历
mv	移动文件	who	显示当前登录用户的信息
chown	修改文件或目录的所有者	Talk	与其他用户对话
chgrp	修改文件或目录的用户组	write	给指定用户发消息
chmod	修改文件或目录的权限	df	对文件系统的磁盘空间使用情况进行统计
rm	删除文件	ps	显示进程相关信息
mkdir	创建目录	netstat	显示网络状态
rmdir	删除目录，要求目录为空	basename	提取基本文件名
grep	查找指定的字符串	man	显示联机参考手册
find	查找指定的文件	su	临时切换到另一用户
egrep	支持扩展正则表达式的 grep 命令	cut	剪切文件
>>	附加到指定文件的结尾	vi	启动 vi 编辑器
>	输出重定向	emacs	启动 emacs 编辑器
<	输入重定向	awk	过滤器
	管道符，用于连接多个命令，前一命令的输出作为后一命令的输入	sed	流编辑器
	逻辑或	pr	编排文本格式

Shell 脚本具有如下优点。

- 一次性完成目标
- 不用重复键入一系列命令
- 简化用户的操作



注意：建立 Shell 脚本文件同建立普通文件的方式一样，可以使用 vi、emacs 等命令来建立。

1.1.3 启动和退出

用户在成功登录进入系统后，系统产生一个特定的 Shell（这是用户的第一个进程）负责解释执行用户的命令或 Shell 脚本。



在系统提示符后,用户键入命令或 Shell 脚本名,来执行指定的命令或 Shell 脚本。如果用户希望中止命令或脚本的执行,可以直接按 Ctrl+C 键来完成。

如果用户结束工作希望退出系统,可以按 Ctrl+D 键或使用命令 `logout` 或 `exit`。

1.2 常用的 Shell 类型

Shell 具有不同的风格,就像 UNIX 操作系统有不同的风格一样。每种不同的风格都具有各自的命令和功能,最常用的 Shell 包括 Bourne Shell、Bourne Again Shell、C Shell、Korn Shell 和 Perl Shell 等。



注意: 本书编程部分所涉及的脚本语言如非特别说明,以流行的 Bourne Again Shell 为例。

用户可以选择自己喜欢的 Shell (在系统管理员为用户创建账号时指定),在 `/etc/passwd` 文件最后一个字段中可以看到用户当前使用的 Shell 的名称。



注意: 可以使用 `grep` 命令查看 `/etc/passwd` 文件,以显示当前登录用户使用的 Shell,命令格式如下。

```
#grep $LOGNAME /etc/passwd  
root:x:0:0:root:/root:/bin/bash // 以 root 用户名登录的
```

1.2.1 Linux 默认 Shell

Bourne Again Shell (简称 Bash) 是 Linux 默认的 Shell,也是最流行的一种 Shell,各个版本的 UNIX 和 Linux 几乎都支持 Bash。Bash 继承了 Bourne Shell 的标准并扩充了人机交互的特性。Brian Fox 是 Bash 的创始人,被称为 Bash 之父。Bash 提供命令历史、命令补全、命令编辑等功能。



注意: 要启动 Bash 只需要在命令行键入 `bash` 并按回车键即可。

1.2.2 Bourne Shell 的扩展版

贝尔实验室的 David Korn 设计了 Korn Shell (简称 ksh)。Korn Shell 是 Bourne Shell 的继承和发展,使用 Bourne Shell 编写的脚本可以不必修改就直接使用 Korn Shell 运行。同时 Korn Shell 包含 C Shell 的特性,具有较强的作业控制能力,如挂起、后台执行、唤醒、终止等,还提供了窗口功能。

1.2.3 网关接口 Shell

Perl Shell 是另一个比较流行的 Linux Shell。Perl (Practical Extraction 和 Report Language) Shell 通常用来编写 Web 应用程序的通用网关接口 (GCI)。

1.2.4 早期 Shell

Bourne Shell (简称 bsh) 是 UNIX 最初使用的 Shell,由 Stephen Bourne 在 1979 年创建的,具有简单、编程能力强等特点,但操作使用不够方便,在同用户交互方面不如其他几种 Shell