北京科海培训中心

新概念 Linux 教程

杜 炜 编著

北京科海集团公司 出品

2001.9

内容提要

Linux 是当今最流行的操作系统之一,而 Red Hat Linux 是 Linux 众多版本中最流行的一种。本书以 Red Hat Linux 7.0 版本为例介绍 Linux 系统的安装配置、网络功能的配置、常用操作、系统管理和编程简介等内容。

全书由 21 章组成。考虑到绝大多数操作人员的实际情况,本书介绍的都是最基本 的内容,并在此基础上进行适当扩展。因此,本书适合初学者和具有一定基础知识并希 望进一步提高的读者,也适合作为各类计算机培训班的使用教材。

本书内容丰富全面,叙述深入浅出、清晰易懂,对操作进行条理化,并将许多选项 及参数的说明列表显示,便于读者对比。学习完本书后,将能够胜任 Linux 系统管理和 网站维护等工作。

本书配有随书光盘和多媒体光盘,为读者的学习提供更加快捷方便的途径。

书 名:新概念 Linux 教程

作 者:杜 炜

责任编辑:安靖

- 出 品:北京科海集团公司
- 印 刷 者:北京门头沟胶印厂
- 发行:新华书店总店北京科技发行所
- 开 本: 787×1092 1/16 印张: 21.875 字数: 504 千字

版 次:2001年9月第1版 2001年9月第1次印刷

印 数:0001~5000

- 盘 号: ISBN 7-89999-330-X
- 定 价:29元(1张多媒体光盘)

前言

欢迎开始使用 Linux 系统!

过去, PC 机平台上运行的操作系统主要是纯粹的商业产品 Windows 系列。也许有人 使用 UNIX,但 UNIX 是一种高端的操作系统,虽然性能可靠但对硬件的要求超出了 PC 的 承受能力。

现在,无论是自身还是用户群,成长最快的操作系统当属 Linux,它已经成为许多用户 最喜欢的操作系统。Linux 之所以如此流行,是因为它提供了类似 UNIX 的稳定性,又可以 在低配置的硬件条件下流畅运行,最重要的一点是该系统的使用,包括源代码是完全免费 的。Linux 是这个时代不可多得的优秀软件。

可以毫不夸张地说, Linux 系统提供了 Windows 操作系统所提供的几乎所有功能,有的功能则更加强大。例如,在 Linux 环境下,如果重新配置了网络协议(如 IP 地址),不需要重新启动计算机就可以立即生效,是不是比 Windows 9x 系列更方便?当然, Linux 带来的好处远不止这些。Linux 作为服务器表现得也非常出色。使用 Linux 可以简单轻松地架设各种服务器,包括 WWW 服务器、邮件服务器、FTP 服务器、Telnet 服务器、DNS 服务器、NFS 服务器以及 Samba 服务器等。

Linux 对于用户是没有秘密可言的,只要愿意,所发生的一切用户都有权知道。究竟在 什么程度上使用 Linux,完全由自己决定。

本书循序渐进、深入浅出地介绍了 Linux 的简史、Red Hat Linux 7.0 系统的安装、配置和常用操作,以及 Linux 系统管理、网络配置以及网络服务配置和管理等内容。

全书由 21 章组成:

第1章~第4章 介绍了 Linux 的安装配置。包括 Linux 支持的硬件设备、Linux 的安装过程以及配置 X Window 系统等内容,能够让读者做到独立安装和配置 Red Hat Linux 系统。

第 5 章~第 7 章 介绍了 Linux 的常用操作,包括 GNOME 环境、Linux 常用命令以及 Linux 系统工具的使用等内容。

第 8 章~第 15 章 介绍了 Linux 的网络服务器配置,包括系统的启动和关闭、配置网 络、各种基本网络服务和协议以及相应的服务器的配置等内容,能够使读者使用 Linux 强 大的网络功能。

第 16 章~第 18 章 介绍了 Linux 的常用系统管理,内容包括 Linux 下的设备管理、用 户管理等。

第 19 章~第 21 章 介绍了 Linux 环境下的简单编程。包括 Shell 脚本的编制、Perl 程 序编制以及 Tcl 程序编制等内容,能够使读者对所谓"高难"的 Linux 编程消除恐惧感, 并会发现,原来这一切自己也可以做到。

如果您刚接触 Linux 系统, 那么建议您从第1章顺序阅读, 以获得系统的学习; 如果

您的计算机系统上已经安装了 Linux 系统,那么可以学习系统的管理和编程部分;如果您 是一位对网络服务感兴趣的高级用户,希望同别人分享您的计算机资源,那么可以从介绍 服务配置的部分获得帮助。

由于时间仓促,疏漏之处在所难免,恳请广大读者指正和谅解。

编者

2001年9月

目	录

第1章	Red Hat Linux 简介	1
1.1	Linux 简史	1
1.2	自由软件和 GPL	
1.3	Linux 发布版本简介	
1.4	Linux 的特点与适用场合	5
	1.4.1 Linux 的特点	5
	1.4.2 Linux 的适用场合	6
	1.4.3 选择 Red Hat Linux 的原因	
1.5	获取帮助的途径	
第2章	安装 Red Hat Linux 7.0	11
2.1	安装之前的准备	11
2.2	安装设置	
	2.2.1 设置 Red Hat Linux 7.0 Work station	
	2.2.2 设置 Red Hat Linux 7.0 Server System	
	2.2.3 定制 Red Hat Linux 7.0	
2.3	安装	
第3章	使用 LILO	22
第3章 3.1	使用 LILO 多重引导加载程序简介	22
第3章 3.1 3.2	使用 LILO 多重引导加载程序简介 安装 LILO	22 22 22
第3章 3.1 3.2	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置	22 22 22 22
第3章 3.1 3.2	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置 3.2.2 安装 LILO	22 22 22 23 23
第3章 3.1 3.2 3.3	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置 3.2.2 安装 LILO 配置 LILO	22 22 22 23 23 23 25
第3章 3.1 3.2 3.3	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置 3.2.2 安装 LILO 配置 LILO	22 22 22 23 23 23 25 25
第3章 3.1 3.2 3.3	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置 3.2.2 安装 LILO 配置 LILO	22 22 22 23 23 23 25 25 25 26
第 3 章 3.1 3.2 3.3	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置 3.2.2 安装 LILO 配置 LILO 3.3.1 引导多个操作系统 3.3.2 启动引导延时 3.3.3 缺省的操作系统	22 22 22 23 23 23 25 25 25 26 26
第3章 3.1 3.2 3.3	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置 3.2.2 安装 LILO 配置 LILO 3.3.1 引导多个操作系统 3.3.2 启动引导延时 3.3.3 缺省的操作系统 3.3.4 配置选项	22 22 22 23 23 23 25 25 25 26 26 26 26
第 3 章 3.1 3.2 3.3	使用 LILO 多重引导加载程序简介 安装 LILO 3.2.1 LILO 程序安装位置 3.2.2 安装 LILO 配置 LILO 3.3.1 引导多个操作系统 3.3.2 启动引导延时 3.3.3 缺省的操作系统 3.3.4 配置选项 3.3.5 确定磁盘参数	22 22 22 23 23 23 25 25 25 26 26 26 26 26 27
第3章 3.1 3.2 3.3	使用 LILO. 多重引导加载程序简介	22 22 22 23 23 23 25 25 25 26 26 26 26 26 26 27 28
第 3 章 3.1 3.2 3.3	使用 LILO. 多重引导加载程序简介. 安装 LILO. 3.2.1 LILO 程序安装位置. 3.2.2 安装 LILO. 配置 LILO. 3.3.1 引导多个操作系统. 3.3.2 启动引导延时. 3.3.3 缺省的操作系统. 3.3.4 配置选项. 3.3.5 确定磁盘参数. 3.3.6 删除 LILO 和恢复 MBR. 3.3.7 代替 LILO 的方案.	22 22 22 23 23 23 25 25 25 26 26 26 26 26 26 27 28 29
第3章 3.1 3.2 3.3	使用 LILO. 多重引导加载程序简介	22 22 22 23 23 23 25 25 25 26 26 26 26 26 26 26 27 28 29 29
第3章 3.1 3.2 3.3 3.3	使用 LILO. 多重引导加载程序简介	22 22 22 23 23 23 25 25 25 26 26 26 26 26 26 26 27 28 29 29 29
第3章 3.1 3.2 3.3 3.3	使用 LILO. 多重引导加载程序简介. 安装 LILO. 3.2.1 LILO 程序安装位置. 3.2.2 安装 LILO. 配置 LILO. 3.3.1 引导多个操作系统. 3.3.2 启动引导延时. 3.3.3 缺省的操作系统. 3.3.4 配置选项. 3.3.5 确定磁盘参数. 3.3.6 删除 LILO 和恢复 MBR. 3.3.7 代替 LILO 的方案. 引导 Linux 系统. 3.4.1 正常启动. 3.4.2 LILO 引导提示选项.	22 22 22 23 23 23 25 25 25 26 26 26 26 26 26 26 26 26 29 29 29 29 29 30

	3.4.3 诊断使用启动引导时挂起错误	
第4章	配置 X 系统	
4.1	XFree86 系统配置简介	
4.2	XF86Config 文件	
	4.2.1 Files 部分	
	4.2.2 ServerFlags 部分	
	4.2.3 Keyboard 部分	
	4.2.4 Pointer 部分	
	4.2.5 Monitor 部分	
	4.2.6 Graphics Device 部分	
	4.2.7 Screen 部分	
4.3	使用配置工具	
	4.3.1 使用 Xconfigurator	
	4.3.2 使用 xf86config	
4.4	习题	
第5章	轻松使用 GNOME 环境	53
5.1	GNOME 简介	
5.2	GNOME 桌面环境	
5.3	使用 GNOME 客户程序和工具	
	5.3.1 使用 Calendar 程序定制提醒本	
	5.3.2 使用 GNOME Control Center 定制桌面	
	5.3.3 使用 Panel 菜单定制面板	61
5.4	习题	
第6章	Linux 常用命令	63
6.1	Linux 常用命令简介	
6.2	常用文件、目录命令	
	6.2.1 cd 命令	
	6.2.2 pwd 命令	
	6.2.3 ls 命令	
	6.2.4 touch 命令	
	6.2.5 rm 命令	69
	6.2.6 mkdir 命令	
	6.2.7 rmdir 命令	
	6.2.8 cp 命令	
	6.2.9 mv 命令	
	6.2.10 cat 命令	
	6.2.11 more 命令	74

	6.2.12 less 命令	
	6.2.13 head 和 tail 命令	
	6.2.14 ln 命令	
	6.2.15 find 命令	
	6.2.16 whereis 命令	77
	6.2.17 locate 命令	
	6.2.18 gzip 命令	
	6.2.19 split 命令	
	6.2.20 chmod 命令	
	6.2.21 chown 命令	
6.3	其他常用命令	
	6.3.1 date 命令	
	6.3.2 time 命令	
	6.3.3 jobs/ps 命令	
	6.3.4 bg 命令	
	6.3.5 fg 命令	
	6.3.6 kill 命令	
	6.3.7 df/du 命令	
6.4	习题	
第7章	Linux 的系统工具	84
7.1	经典编辑器 vi	
	7.1.1 vi 的模式	
	7.1.2 打开和退出 vi	
	7.1.3 vi 的输入模式	
	7.1.4 vi 命令模式	
	7.1.5 文件指令	
	7.1.6 vi 命令及其作用列表	
	7.1.7 vi 命令模式的输入选项(:命令)	
		0.0
7.0	7.1.8 设直选坝	
1.2	7.1.8 设置选项 多功能管理工具 linuxconf	
1.2	7.1.8 设直选坝多功能管理工具 linuxconf7.2.1 用户管理	
1.2	 7.1.8 设直选坝 多功能管理工具 linuxconf 7.2.1 用户管理 7.2.2 启动管理 	
7.2	 7.1.8 设置选坝 多功能管理工具 linuxconf 7.2.1 用户管理 7.2.2 启动管理 软件管理工具 RPM 	92 93 94 97 100
7.2	 7.1.8 设直选坝 多功能管理工具 linuxconf 7.2.1 用户管理 7.2.2 启动管理 软件管理工具 RPM 7.3.1 下载应用程序 	
7.2	 7.1.8 设置选项 多功能管理工具 linuxconf 7.2.1 用户管理 7.2.2 启动管理 尔件管理工具 RPM 7.3.1 下载应用程序 7.3.2 RPM 基础 	92 93 94 97 100 100 102
7.3	 7.1.8 设直选坝 多功能管理工具 linuxconf	92 93 94 97 100 100 102 103
7.2	 7.1.8 设置选项	92 93 94 94 97 100 100 100 102 103 103

	7.4.1	优良备份的品质	
	7.4.2	备份或者恢复操作之前需要考虑的几个因素	
	7.4.3	备份什么	111
	7.4.4	压缩备份	
	7.4.5	备份媒体	
	7.4.6	备份策略	
	7.4.7	备份工具	
7.5	习题		
第8章	Linux	x 网络基础	
8.1	TCP/I	P 协议网络简介	
8.2	TCP/I	P 协议及网络工作原理	
	8.2.1	以太网	
	8.2.2	IP 协议	
	8.2.3	TCP 协议	
	8.2.4	UDP 协议	
	8.2.5	端口	
8.3	IP 网络	备	
	8.3.1	子网	
	8.3.2	网关	
	8.3.3	路由表	
	8.3.4	Internet 控制消息协议	
8.4	域名系	系统	
	8.4.1	输入 DNS	
	8.4.2	利用 DNS 进行名字查找	
	8.4.3	域名服务器	
第9章	配置	Linux 网络	
9.1	网络酉	2置文件	
9.2	手工西	7置网络	
	9.2.1	设置主机名	
	9.2.2	设置 IP 地址	
	9.2.3	主机和网络文件	
	9.2.4	配置网络接口	
	9.2.5	配置回送接口	
	9.2.6	配置以太网接口	
9.3	网络香	2置工具	
	9.3.1	网络配置工具 netcfg	
	9.3.2	网络配置工具 netconf	
	9.3.3	网络配置工具 netconfig	

	9.4	习题	
第	10 章	配置域名服务	
	10.1	域名服务简介	
	10.2	使用 hosts 文件	
		10.2.1 配置 host.conf 文件	
		10.2.2 解析器环境变量	
	10.3	DNS 简介	
		10.3.1 配置工具	
		10.3.2 DNS 术语	
		10.3.3 域名服务器和解析器	
	10.4	配置服务器和解析器	
		10.4.1 配置服务器	
		10.4.2 配置解析器	
	10.5	配置 Cache-Only 域名服务器	
	10.6	习题	
第	11章	配置 FTP 服务器	
	11.1	FTP 服务器简介	
	11.2	安装 FTP 服务器	
		11.2.1 FTP 服务器的工作原理	
		11.2.2 获取和安装 FTP 服务程序	
	11.3	配置 FTP 服务器	
		11.3.1 /etc/ftpaccess 文件	
		11.3.2 /etc/ftphosts 文件	
		11.3.3 /var/log/xferlog 文件	
		11.3.4 配置示例	
	11.4	FTP 管理工具	
		11.4.1 ftpshut 命令	
		11.4.2 ftpwho/ftpcount 命令	
	11.5	使用 FTP 客户程序	
		11.5.1 ftp 命令	
		11.5.2 ncftp 命令	
		11.5.3 gftp 命令	
	11.6	习题	
第	12 章	配置 Apache 服务器	164
	12.1	Apache 服务器简介	
	12.2	安装 Apache 服务器	
		12.2.1 使用 RPM 工具安装	

		12.2.2	自己编译源代码安装	
	12.3	配置	Apache 服务器	
		12.3.1	httpd.conf 配置文件	
		12.3.2	htaccess 文件	
		12.3.3	虚拟主机	
		12.3.4	启动和停止服务器	
	12.4	示例		
	12.5	习题		
第	13 章	配置	邮件服务器	
	13.1	电子曲	8件简介	
		13.1.1	使用的协议	
		13.1.2	邮件消息	
		13.1.3	邮局协议	
		13.1.4	访问协议	
	13.2	配置I	.inux 邮件服务器	
		13.2.1	sendmail 的工作情况	
		13.2.2	安装 sendmail	
		13.2.3	sendmail 的配置文件	
		13.2.4	自动生成 sendmail.cf 文件	
		13.2.5	测试 sendmail 和 sendmail.cf 文件	193
	13.3	设置 F	POP3	
	13.4	邮件客	§户程序	195
		13.4.1	elm 程序	195
		13.4.2	pine 程序	197
	13.5	习题		
第	14 章	配置	Samba 服务器	
	14.1	Samba	服务器简介	
	14.2	安装 S	amba	
		14.2.1	Samba 的组成	
		14.2.2	安装 Samba	
	14.3	启动 S	Samba 守护进程	
		14.3.1	从 inetd 中运行	
		14.3.2	从系统的开机引导脚本程序中运行	
		14.3.3	不从系统的开机引导脚本程序中运行	
		14.3.4	手动启动 Samba 守护进程运行	
	14.4	配置 S	amba	
	14.5	使用 S	amba	
		14.5.1	配置 Linux 到 Windows 共享	

	14.5.2	配置 Windows 到 Linux 共享	
	14.5.3	测试 Linux 共享设置	
	14.5.4	测试 Windows 共享设置	
	14.5.5	配置 Linux 打印机共享	
	14.5.6	配置 Windows 打印机共享	
14.6	测试 S	amba	211
14.7	习题		
第 15 章	维护	系统的网络安全	213
15.1	服务器	安全保护简介	
	15.1.1	Linux 系统的文件安全	
	15.1.2	用户访问安全	
15.2	防火墙	简介	
	15.2.1	防火墙的作用	
	15.2.2	防火墙的分类	
15.3	Linux	的安全管理	
	15.3.1	关闭简单的薄弱环节	
	15.3.2	列出成功登录的记录清单	
	15.3.3	列出不成功登录的记录清单	
	15.3.4	查找安全漏洞	
15.4	设置]	Linux 防火墙	
	15.4.1	使用 ipchains 创建防火墙	
	15.4.2	代理服务	221
第 16 章	管理	硬盘和文件系统	
16.1	设备的]概念	
16.2	各类设	台	
	16.2.1	硬盘	
	16.2.2	软盘	
	16.2.3	CD-ROM	225
	16.2.4	磁带	
	16.2.5	并行端口	
	16.2.6	CD-ROM 刻录机	
16.3	磁盘管	理	
	16.3.1	格式化	
	16.3.2	分区	
16.4	文件系	统简介	
	16.4.1	文件系统的概念	
	16.4.2	文件系统的种类	
	16.4.3	文件类型	

16.5	使用了	文件系统	
	16.5.1	创建文件系统	
	16.5.2	装入和卸装文件系统	
	16.5.3	设置文件系统	
	16.5.4	文件系统的管理工具	
16.6	文件系	系统的组织	
	16.6.1	根文件系统	
	16.6.2	/etc 文件系统	
	16.6.3	/dev 文件系统	
	16.6.4	/usr 文件系统	
	16.6.5	/var 文件系统	
	16.6.6	/proc 文件系统	
16.7	文件机	又限	
	16.7.1	文件权限的概念	
	16.7.2	文件权限的构成	
	16.7.3	文件权限的依赖性	
	16.7.4	改变权限	
16.8	习题		
第 17 章	常用	月月户管理	254
17.1	用户管	管理简介	
	17.1.1	账号	
	17.1.2	账号口令	
	17.1.3	shadow 口令	
17.2	超级月	用户管理	
17.3	账号酉	記置文件	
	17.3.1	/etc/passwd 文件	
	17.3.2	- /etc/group 文件	
17.4	普通月		
	17.4.1	添加用户	
	17.4.2	修改用户属性	
	17.4.3	禁用用户	
	17.4.4	删除用户	
	17.4.5	改变用户组和所有权	
17.5	X Win	ndow 系统中的用户管理	
	17.5.1	添加用户	
	17.5.2	修改用户属性	
	17.5.3	禁用用户	
	17.5.3 17.5.4	禁用用户 删除用户	

	17.6	用户查询	
		17.6.1 who 命令	
		17.6.2 finger 命令	
		17.6.3 id 命令	
	17.7	维护用户账号安全注意事项	
	17.8	习题	
第	18章	常用网络管理	271
	18.1	ping 程序	
	18.2	ifconfig 命令	
		18.2.1 查看接口配置	
		18.2.2 配置接口	
		18.2.3 设置 IP 别名	
		18.2.4 ifconfig 命令选项列表	
	18.3	route 程序	
		18.3.1 显示路由信息	
		18.3.2 添加和删除路由信息	
	18.4	traceroute 程序	
	18.5	netstat 程序	
		18.5.1 使用 netstat	
		18.5.2 netstat 选项	
	18.6	tcpdump 程序	
	18.7	习题	
第	19章	shell 编程	
	19.1	shell 编程简介	
	19.2	创建和运行 shell 程序	
		19.2.1 创建 shell 程序	
		19.2.2 运行 shell 程序	
	19.3	shell 编程规则	
		19.3.1 变量	
		19.3.2 特殊字符	
		19.3.3 运算	
		19.3.4 条件语句	
		19.3.5 循环语句	
		19.3.6 函数	
	19.4	习题	
第	20 章	Perl 编程	
	20.1	Perl 编程简介	

20.2	Perl 简	单示例	304
20.3	Perl 编	程规则	305
	20.3.1	Perl 变量类型	
	20.3.2	Perl 运算符	
	20.3.3	条件语句	
	20.3.4	循环语句	
	20.3.5	文件操作	
	20.3.6	文本操作	
	20.3.7	过程	
	20.3.8	命令行选项	
20.4	→ 习题		
第 21 章	tcl/tk	编程	
21.1	tcl/tk 编	扁程简介	
21.2	tcl 编程	呈基础	
	21.2.1	交互使用方式	
	21.2.2	非交互使用方式	
21.3	tcl 编程	呈规则	
	21.3.1	命令	
	21.3.2	注释	
	21.3.3	变量	
	21.3.4	字符串操作	
	21.3.5	数字操作	324
	21.3.6	引用和置换	325
	21.3.7	条件语句	
	21.3.8	循环语句	
	21.3.9	文件操作	
	21.3.10	过程	
21.4	tk 入门]	
	21.4.1	tk 构件	
	21.4.2	创建构件	333
	21.4.3	构件的选项	333
	21.4.4	编程示例	
21.5	习题		

第1章 Red Hat Linux 简介

本章主要对 Linux 以及 Red Hat Linux 作一个简单的介绍。

本章内容包括 Linux 简史、Linux 发布版本的选择、Linux 以及 Red Hat Linux 的特点以及如何获取 Linux 帮助等。

学习本章后,可以对 Linux 和 Red Hat Linux 有一个提纲挈领的认识,以便进一步学习 安装和使用 Linux 系统。

1.1 Linux简史

Linux 最初的核心程序是由一名芬兰赫尔辛基大学的大学生 Linus Torvalds 创建的。 1990年, Linus Torvalds 还在读大学本科计算机专业的时候,因为学校的服务器不能满足他 的使用需要,于是自己买了一台 Intel 386 PC。然后, Linus 在自己的 PC 上利用 Tanenbaum 教授自行设计的微型 UNIX 操作系统 Minix 作为开发平台,开发出了属于他自己的第一个 程序。

Linus 用汇编语言编写了一个在 80386 保护模式下处理多任务切换的程序,"这个程序 包括两个进程,都是向屏幕上写字母,然后用一个定时器来切换这两个进程。"他回忆说, "一个进程写 A,另一个进程写 B,所以我就在屏幕上看到了 AAAA、BBBB 如此循环重 复的输出结果。"

后来 Linus 又编写了一些硬件的设备驱动程序和一个小的文件系统,由此诞生了 Linux 0.0.1 版本。但是它只具有操作系统内核的雏形,却不能运行,必须在有 Minix 的机器上编 译以后才能运行。

随后, Linus 决定抛开 Minix, 于是在 1991 年 10 月 5 日发布了 Linux 0.0.2 版本。在这个版本中可以运行 bash (the GNU Bourne Again Shell——一种用户与操作系统内核通信的软件)和 gcc (GNUC 编译器)。

这是 Linux 被首次公布于众。同年 11 月,该系统 0.10 版发行;12 月,0.11 版发行。 Linus 允许免费自由运用该系统源代码,并且鼓励其他人对其进行进一步开发。人们也的确 开始这样做了。

渐渐地, Linux 引起了黑客们(hacker)的注意。由于黑客们通过计算机网络加入了 Linux 的内核开发, Linux 倾向于成为一个黑客的系统。Linux 内核的开发被认为是真正的 编程。从此, Linux 发展迅猛, 到 1993 年底~1994 年初, Linux 1.0 终于诞生了。

Linux 1.0 已经是一个功能完备的操作系统,而且它的内核写得紧凑高效,可以充分发 挥硬件的性能,在 4MB 内存的 80386 机器上也表现得非常好,至今人们对此还在津津乐道。

因为硬件的发展实在太快,因此从2.1系列的内核出现以来,Linux开始走高端的路子。

但是 Linux 并没有失去它的本色。Linux 具有良好的兼容性和可移植性,大约在 1.3 版本出现之后,开始向其他硬件平台上移植,包括当前最快的 CPU。所以,不要总把 Linux 与低档硬件平台联系到一块,Linux 从开始发展到今天,这一直是一个误区。它只是将硬件的性能充分发挥出来而已。Linux 必将从低端应用横扫到高端应用。

在 Linus 带领下,通过 Internet,一个世界范围内的开发组正在对 Linux 进行坚持不懈的开发。世界各地的计算机高手把一个随时都有可能被遗弃的萌芽,扶植成了一个计算机领域中任何人都无法忽视的力量。现在,不少专业人员认为 Linux 是一个最安全、最稳定以及对硬件系统最不敏感的操作系统。

Linux 目前最稳定的版本是 2.2.16 版。Linux 没有利用来自 AT & T 或任何其他专有资源的代码。为 Linux 开发的许多软件是由自由软件基金会(Free Software Foundation)的 GNU (GNU is Not Unix)计划组开发的。因此, Linux 十分便宜;事实上它是免费的。

目前,装有 Linux 的机器遍布全世界。在当今金钱至上的商业社会,一个自由软件得 到如此多人的关心,不能不说是一个奇迹。

1.2 自由软件和GPL

当前流行的软件按其提供方式可以划分为3种模式:商业软件(Commercial software), 共享软件(Shareware)和自由软件(Freeware 或 Free software)。

商业软件由开发者出售拷贝并提供技术服务,用户只有使用权,不得进行非法拷贝、 扩散、修改或添加新功能;共享软件由开发者提供软件试用程序拷贝授权,用户在试用该 程序拷贝一段时间之后,必须向开发者交纳使用费用,开发者则提供相应的升级和技术服 务;而自由软件则由开发者提供软件全部源代码,任何用户都有权使用、拷贝、扩散以及 修改该软件,同时用户也有义务将自己修改过的程序源代码公开。

1984 年, Richard Stallman 组织开发了一个完全基于自由软件的软件体系计划——GNU,并拟定了一份公共版权许可证(GPL, General Public License)。

GNU 计划的宗旨是:消除对计算机程序拷贝、分发、理解和修改的限制。也就是说, 每一个人都可以在前人工作的基础上对程序加以利用、修改或添加新内容,但必须公开源 代码,允许其他人在此基础上继续工作。

在 Linux 的发展历程上,最重要的一件事就是 Linux 加入 GNU 并遵循公共版权许可证。

从一开始, Linus 就决定自由扩散 Linux, 包括源代码, 他在 comp.os.minix 新闻讨论组 里发布 Linux 0.0.2 时写到:

" Do you pine for nice days of Minix-1.1, when mem were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrasting when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.

"As I mentioned a month ago, I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable(though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is

just version 0.0.2 ... but I've successfully run bash ,gcc ,gnu-make ,gnu-sed ,compress ,etc.under it. "

Linux 从产生到发展一直遵循的是"自由软件"的思想。正因为如此, Linux 才发展得如此迅速和健康。1994 年 3 月 14 日, Linus 发布于 Linux 的第一个"产品"版 Linux 1.0 的时候,是按完全自由发布版权进行发布的。它要求所有的源代码必须公开,而且任何人均不得从 Linux 交易中获利。

然而,半年以后,Linus 开始意识到这种纯粹的自由软件的方式对于 Linux 的发布和发展实际上是一种障碍,因为它限制了 Linux 以磁盘拷贝或者 CD-ROM 等媒体形式进行发布的可能,也限制了一些商业公司参与 Linux 的进一步开发并提供技术支持的良好愿望。于是 Linus 决定转向 GPL 版权。这一版权除了规定有自由软件的各项许可权之外,还允许用户出售自己的程序拷贝,并从中赢利。

此举大大加强了 GNU 和 Linux,几乎所有 GNU 库/软件都可以移植到 Linux 上,完善并提高了 Linux 的实用性,同时 GNU 也有了一个根基,更重要的是遵循公共版权许可证, 在继承自由软件精神的前提下,不再排斥对自由软件的商业行为(如把自由软件打包以光 盘形式出售),不排斥商家对自由软件进一步开发,不排斥在 Linux 上开发商业软件。

从此以后,便有多家技术力量雄厚又善于市场运作的商业软件公司加入了原先完全由 业余爱好者和网络黑客所参与的这场自由软件运动,开发出了多种 Linux 的发布版本,增 加了更易于用户使用的图形界面和众多的软件开发工具,极大地拓展了 Linux 的全球用户 基础。还有一些公司在 Linux 上开发商业软件或把其他 UNIX 平台的软件移植到 Linux 上 来。如今很多 IT 业界的大腕,如 IBM、Intel、Oracle、Infomix、Sysbase、Corel、Netscape、 CA 和 Novell 等都宣布支持 Linux。商家的加盟,弥补了纯自由软件的不足和发展障碍。

从此 Linux 又开始了一次飞跃,出现了很多的 Linux 发行版。如 Slackware、Red Hat、 SuSE 和 OpenLinux 等诸多版本。其中,支持中文的有 TurboLinux、Xteam、Bluepoint 以及 红旗 Linux 等。Linux 主要是指操作系统内核,对所有发行版内核源代码都是一样的(但集 成的内核版本可能因发行时间不同而有所不同)。

Linus 本人也认为:"使 Linux 转向 GPL 是我一生中所做过的最漂亮的一件事。"

1.3 Linux 发布版本简介

Linux 有多种版本,它们都各有什么特点,哪一种才是正宗的Linux 呢?本节将对Linux 版本作一个总结比较,读者可以根据自己的喜好进行选择。

Linux 说到底只是个内核。一些公司和爱好者在不断地开发和推出新内核,而相应的应用工具和软件则多用 GNU 和其他的自由软件。由于 Linux 基本遵循 POSIX 标准,因此大多数 UNIX 软件的编译移植不会有太大的困难(UNIX 可移植性相当好,同一套源代码在小至微机大至巨型机上都可编译)。而将 Linux 内核和这些外围的程序整合起来的任务是由 Linux 发行商完成的,他们推出不同的发行版本(distribution)。

Linux 操作系统遵从 GNU 公共许可证 GPL 的规定。Linux 操作系统的大多数发行版本都使用标准的 Linux 内核,有时候也会稍微加以修改,它们都能够从因特网上的站点下载

获得。当然这个操作系统现在的规模已经变得相当庞大了,因此许多人更愿意购买它的光 盘发行版本。每一种发行版本彼此之间都稍微有一些差异,这些差异主要表现在它们各自 的安装程序包上,还表现在安全性与可用性等方面侧重点的不同。把系统内核、附带的工 具程序及应用软件包等等打包在一起,这样组成的一个 Linux 操作系统的集合体就是某一 种发行版本。

在最初的时候,有一些小公司把操作系统和一部分增值服务,如系统安装帮助及预配 置程序之类的东西打包到光盘上。现在,像 Red Hat 这样公司已经能够把大型的操作系统 打包为它的某个发行版本,其附加软件中包括了复杂的安装程序与商业软件演示版等;另 外还包括了使用手册和技术支持等方面的内容。

常见的 Linux 发行版本如下。

1 . Red Hat Linux

它基于 rpm 的包结构,核心代码约有 200MB。其好处是安装使用简单方便,并且 rpm 包的结构也使包的安装和卸载变得更方便。另外 Red Hat 的更新升级版出得也很及时,被 评为 Best Network OS。Red Hat Linux 同时也支持 Intel、Alpha 和 SPARC 平台,这也是 RedHat 公司最引以自豪的地方。

一般推荐入门时使用 Red Hat Linux,因为它安装容易。 若使用 Linux 已经有一段时间, 那就无所谓了,因为自己有了评价不同版本的能力,就可以根据自己的习惯、喜好和用途 来选择。

2 . Slackware Linux

Slackware 基于 tgz 的包结构,是历史很悠久的 Linux 发布版本,其核心部分有 120MB 左右。Slackware 比较公认的是做服务器时性能较好。Slackware Linux 支持大多数 Intel PC。 它支持对称多处理和 PCI,并为一些处理器进行了特别的编码优化。

3 . Debian Linux

Debian Linux 基于 deb 的包结构,这也是一个不错的发布版本,与 GNU 的关系紧密。 它收集的软件很全,代码约有 400MB,而且其 deb 包安装删除也很方便。

Debian Linux 由一群自愿者进行维护和升级。它的先进的管理工具包使得安装和维护 都异常简单。发布前全面的测试保证了系统的高度可靠性。一个公开的 bug 跟踪系统随时 监控用户的反馈。

4 . SuSE Linux

SuSE 公司是从 X Server 开始,和 XFree86 合作开发 x86 上的 X Server。SuSE 也有自己的发布版本。由于这是一家德国公司,该发行版在欧洲用得更为广泛。

SuSE Linux 有超过 800 个预设软件包以及 400 页的参考手册。其中的 YaST 实用工具 允许用户自己安装、设置和进一步配置系统。SuSE 支持 X Server 的高端图形卡。

5 . Linux Mandrake

Linux Mandrake 是 32 位多任务操作系统。它可以运行在所有 Intel 以及与其兼容的结构中。

6 . Caldera OpenLinux

OpenLinux 包括一些实用工具、图形界面、安装指南以及第三方的应用软件等。

7 . LinuxWare

LinuxWare 是一个十分灵活、易于安装的以及类似于 UNIX 的操作系统。主要面向对 UNIX 系统感兴趣的学生和家庭 PC 使用者。

8 . TurboLinux

TurboLinux 包括一系列的应用程序、一个 GUI 的桌面、文档和技术支持。

9. 红旗 Linux

由中国科学院软件所、北大方正电子有限公司和康柏电脑公司 3 家合作共同推出的国 产中文操作系统红旗 Linux,预装了炎黄中文平台和方正 TrueType 字库,实现了 Linux 上 的 TrueType 显示和打印功能,并且从安装到使用提供了全中文化的界面。红旗 Linux 支持 基于 Intel 芯片的各类 PC 机和服务器。

1.4 Linux的特点与适用场合

上节对各种 Linux 版本作了介绍和比较,那么,究竟选择何种 Linux 版本呢?这取决于各个版本的特性和用户的特定需求。下面介绍选择 Linux 发行版本时应考虑的一些考虑因素。

1.4.1 Linux的特点

Linux 操作系统有以下特点。

1. Linux 是免费的

Linux 像 UNIX - 样, 是一个非常强大的实时操作系统, 但同其他 UNIX 操作系统相比 又非常小。

2. 全面的多任务和真正的 32 位操作系统

Linux 和其他 UNIX 系统一样是真正的多任务系统, 它允许多个用户同时在一个系统上运行多个程序。Linux 还是真正的 32 位操作系统, 它工作在 Intel 处理器的保护模式下。

3. 支持多种硬件平台

从低级的 Intel386 到高级的超级并行计算机系统,都可以运行 Linux 系统。

4. 完全的多任务

Linux 同时能够完成多个任务并且能够访问多个设备。

5. 虚拟内存

Linux 能够利用硬盘的一部分作为虚拟内存,从而扩展可用内存空间。它通过把处于活动状态的进程保留在 RAM,而把使用频率较低或非活动状态的内存放置在硬盘,来提高系

统的效率。虚拟内存也利用系统的所有内存并且不允许内存分段出现。

6. 共享库技术

Linux 利用共享库技术,允许使用标准子过程的程序在运行时共享子过程。每个应用 程序共享一个公用的、运行时可被调用的子程序库,而不是保留各自的软件备份。这可为 系统节省大量空间。

7. X Window 系统

X Window 系统是用于 UNIX 机器的一种图形系统。这一强大的界面系统支持许多应用 程序并且是业界的标准界面。XFree86 则是 Linux 平台上的 X Window 系统。X Window 系统是功能强大的图形界面,支持多种应用程序。

8. 内置网络支持

Linux 使用标准的 TCP/IP 协议,包括网络文件系统(NFS, Network File System)和网络信息服务(NIS, Network Information Service)。在 Linux 系统中,通过 Ethernet 可以连接到 Internet 或当地的局域网。使用 SLIP(Serial Line Internet Protocol)或 PPP(Point to Point Protocol),通过电话线和调制解调器也可连到 Internet 上。

Linux 系统网络功能强大。不仅仅因为 Linux 系统内核中紧密地集成了网络功能和有大量的网络应用程序,更因为 Linux 系统在超强网络需求下表现出的令人惊奇的健壮性。

9. 同 IEEE POSIX.1 标准兼容

由于这一兼容性, Linux 支持许多为所有 UNIX 系统提出的标准。事实上, Linux 系统 上的一些功能是 UNIX 系统所不具备的。

10. 非专有资源代码

Linux 内核没有利用来自任何其他专有资源的代码。

Linux 内核中的代码均为自由代码。Linux 上的大部分程序是自由软件。这些软件是在 自由软件基金会的 GNU 计划下开发的。尽管如此,来自世界各地的黑客、程序员,甚至 商业公司也加入了 Linux 软件开发的行列。

11. GNU 软件支持

Linux 能够运行 GNU 计划的大量免费软件。这些软件包括诸如应用程序开发(GNUC 和 GNUC++等) 系统管理(gawk 和 groff 等)以及游戏(GNUChess、GnuGo 和 NetHack 等)等方面的所有内容。

12. Linux 比其他 UNIX 系统更便宜

通过 Internet 下载 Linux 是免费的。购买 Linux 的发布也很便宜。

1.4.2 Linux的适用场合

Linux 适用于以下任意场合。

1. 个人 Linux 工作站

无论在家中,还是在办公室里,Linux 与基于 Intel 芯片的 PC 的结合都会创造出一台

功能强大的 Linux 机器。Linux 对于那些负担不起 Sun 或 HP 工作站的公司,对于每一个工程师和所有在 X 终端上遇到困难的人,都是很好的产品,它同时也是希望保留一些旧格式文件和继续使用原有程序的 DOS 和 Windows 混合环境下用户的最佳选择。

2. Internet 服务器

Linux 是 Internet 的产物,而且擅长提供 Internet 服务。Linux 缺省提供 WWW、Usenet 新闻、电子邮件和 FTP 等许多功能,可以访问网络内部用户,也可以通过整个 Internet 发 布消息。如果与调制解调器相连,Linux 就变成强大的 Internet 访问接入点。大量的 ISP 因 为 Linux 的可靠性而选择了它。

3.X 终端客户

Linux 对 X11 的优秀支持简化了在应用服务器上远程运行应用的过程,并且可以在本地的 Linux 工作站上显示信息。而且,因为 Linux 提供了全方位的多媒体服务,包括声音、动画和高级图像支持,Linux 比传统 X 终端更占优势。如果用户需要将其 Linux 机器用于多台 X 终端,则可以非常快速而方便地实现。

4.终端服务器、传真服务器和 Modem 服务器

Linux 也能很好地支持串行设备和电话。定制的终端服务器通常支持 SLIP、PPP、Direct Connection、Dial up Apple Talk 和 The Internet Adapter 。Linux 不但能提供上述功能,还可以提供定制的安全性、身份验证和登录过程。一个 ISP 的中等系统可以连接 200 多个调制解调器,提供并维护可靠的拨号服务。

5.网络服务器

与商业组织类似,教育机构也热衷于将 Linux 用作企业服务器。用于文件及打印共享时,可将 Linux 配置为使用 NFS、Apple Talk 及 NetBIOS 协议。其性能优、价格廉的特点使它颇具吸引力。由 LAN 桥接而生成 WAN 也是 Linux 的一个很好的用途。当 Linux 运行于 RISC 和 SMP(对称多处理)硬件系统时,它可为后台处理带来便利及易用性。

6.X应用服务器

任何作为 X 终端的机器都能体会到 Linux 支持 X 应用的范围之大。使用 Linux 作为应 用服务器来加快 RISC 工作站与使用 Linux 作为工作站一样,都是非常合理的选择。如果把 Linux 配置成应用服务器,就能通过运行 DOSX 服务器把 486 等低档次的机器作为哑终端 重新利用起来。这也说明了 Linux 是低预算的 XWindow 网络的选择。

7. UNIX 开发平台

Linux 能够支持 UNIX 开发,它不但支持主流语言,也支持其他语言。其跨平台的环境还可以为其他 UNIX 操作系统平台产生二进制代码。GNU C、GNU C++、GNU Fortran77、ADA、Pascal、Modula2 和 3、Tcl/tk、Scheme 和 SmallTalk/X 均可免费使用,并附带详细的工作代码库。多种语言的编译器有助于节省开发时间。所有这些,再加上灵活的 shell 语言编辑器、源代码包和详细文档都给了编程者充分的可定制环境。另外,它也能作为一个理想的计算机学习系统。

1.4.3 选择Red Hat Linux的原因

使用 Red Hat Linux 有以下几种原因。

1. 基于 Linux 2.2x

Red Hat 的当前版本 7.0 以 Linux 内核的 2.2.16 版本为基础,这意味着它同 Linux 一样免费。任何人都可利用 FTP 从 Internet 下载,并把它安装在自己的系统上。这也意味着 Linux 的所有属性都适用于 Red Hat 版本。

2. 包含 Red Hat 软件包管理程序

Red Hat 软件包管理程序(RPM, Red Hat Package Manager)也包含在内,且同样免费。 这意味着在装载 Red Hat 之后,就不必再装入 RPM。RPM 是一个高级工具,它包含软件 包升级时智能的文件处理、共享文件处理、文档搜索支持以及通过 FTP 进行软件包安装的 功能。用户可以安装、卸载、查询、验证和升级 RPM 软件包。

3. 优良并原始的拷贝

Red Hat 承诺提供公司得到"原始程序"。RPM 源软件包包括原始的、未改变过的源程 序、修补程序以及一个控制文件,它用来定义创建与封装的过程。通过把 Linux 的开发的 代码与被 Red Hat 做过任何必要修改的代码进行清楚地区分和归档,可以使用户更容易、 更有效地同开发人员进行合作。

4. 安全性

Red Hat 在提供最新的安全性方面处于业界领先地位。此外,公司一直在努力提供最新的软件版本。

5. 文档

Red Hat 提供有 250 多页的安装与配置信息,这些信息可通过 FTP 下载或从 RedHat 站 点上查看。控制面板工具的整个覆盖范围,包括网络、用户/组以及打印机工具,都包含在 这些信息中。

6.标准

Red Hat 既遵循 UNIX 的标准,又遵循 Linux 的标准。Red Hat 符合 Linux 的文件系统标准 (FSSTND)。

7.测试

Red Hat 依赖于由 Linus 开创的开放式开发模型。今天,成千上万的工作在世界各地的 人们在对应用程序进行测试并为当今的商业和个人需求提供解决方案。

由此可以看出,通过提供工具、文档以及标准化,Red Hat 已超越了一般的 Linux 系统。

1.5 获取帮助的途径

Linux 是通过 Internet 发展壮大的,所以,如果有什么问题,尽管到 Internet 上去寻找

- 答案。下面是常用的涉及 Linux 的 Web 网址:
 - http://www.ssc.com/linux
 - http://www.caldera.com
 - http://www.redhat.com
 - http://sunsite.unc.edu/mdw
 - http://www.ssc.com/lg
 - http://www.linux.org
 - http://www.li.org
 - http://www.uk.linux.org
 - http://www.blackdown.org
 - http://www.linuxtoday.com
 - http://www.linuxjournal.com
 - http://www.linuxgazatte.com
 - http://www.linux.com
 - http://www.linuxhq.org
 - http://www.freshmeat.net
 - http://www.salshdot.org
 - http://www.gnu.org
 - http://sunsite.unc.edu/mdw/linux.html
 - http://linux.cis.nctu.edu.tw
 - http://linux.csie.nctu.edu.tw

下面是常见的 Linux 新闻讨论组:

comp.os.linux.announce	Linux 的发展情况
comp.os.linux.development.apps	Linux 的应用程序
comp.os.linux.development.system	Linux 的操作系统内核
comp.os.linux.hardware	Linux 硬件方面的问题
comp.os.linux.admin	Linux 系统管理方面的问题
comp.os.linux.misc	Linux 的一些特别的问题和回答
comp.os.linux.setup	Linux 安装和启动
comp.os.linux.answers	关于 Linux 命令的问题和解答
comp.os.linux.help	Linux 的帮助
comp.os.linux.networking	关于 Linux 网络的问题和解答

下面介绍有关 Linux 资料的来源。这些资料多是以电子信息形式给出的,所以需要用 户有相应的访问 Internet、USENET 或 Fidonet 的权限。

1. 在线文档

如果有访问 Internet 的权限,就可以通过匿名 FTP 访问世界各地的有关 Linux 的文档。 如果没有相应权限,一些 CD-ROM 上也会有相关文档。此外,这些文档也会在其他网络上 传播,如 Fidonet、Compu Serve 等。如果可以发送邮件,用户可以试着用 ftp mail 服务器 来获取所需资料。

2. 在 WWW 上

参见本节前面所列网站。

3. 书籍和其他出版物

目前,关于 Linux 的书籍很多,比较有价值的是 Linux 资料工程推荐的书。这些手册 从形式上看类似商品化的 UNIX 资料,内容包括 Linux 的安装到使用操作,非常全面。

市面上还有很多关于 UNIX 的书籍。这些书籍的内容同样适用于 Linux,例如,系统使用和编程等。在这些方面 UNIX 与 Linux 之间的差别很小。另外,有一份关于 Linux 的月 刊名为 Linux Journal,在世界范围内发行。在用户无法访问 USENET 新闻组时,阅读该杂志是保持对 Linux 世界了解的有效途径。

4. USENET 新闻组

参见本节前面所列的新闻组。

5. Internet 邮件列表

如果能访问 Internet 电子邮件,那么即使不能使用 USENET,也可以通过参加一些邮件列表而得到 USENET 的消息。"Linux - 参与者"邮件列表主要是为那些开发者和对开发 过程感兴趣的人提供一个相互联系的方式。可根据自己的爱好和兴趣参加一个或多个小组。 可用的小组有:

NORMAL	用于一般 Linux 问题
KERNEL	用于 Linux 核心程序的开发问题
GCC	有关编译器和程序库开发的讨论
NET	关于 TCP/IP 网络问题的讨论
DOC	有关 Linux 资料的编写和开发的讨论等等

可通过发送电子邮件的方式索取有关邮件列表的资料。

当然,还有其他的关于 Linux 的专用邮件列表。可以通过浏览 USENET 新闻组上的公告和定期在 USENET 的 news.answers 新闻组上发布的邮件列表获得相应资料。

第2章 安装 Red Hat Linux 7.0

在使用 Red Hat Linux 7.0 之前,用户必须安装 Red Hat Linux 7.0。本章主要介绍如何安装 Red Hat Linux 7.0。

本章内容包括安装 Red Hat Linux 7.0 前的准备工作、安装 Red Hat Linux 7.0 的 Work Station 版本的步骤、安装 Red Hat Linux 7.0 的 Server 版本的步骤以及定制安装 Red Hat Linux 7.0 的步骤等内容。通过对本章的学习,可以掌握安装 Red Hat Linux 7.0 的必备知识。

2.1 安装之前的准备

Red Hat Linux 与其他 Linux 版本之间的显著差别之一就是安装轻松容易。它的安装过 程十分简单明了,安装程序能够处理多种不同的系统配置,并且由 Red Hat Linux 安装程序 自动完成。

在安装这一操作系统之前,应该了解将要安装该操作系统的硬件系统。如果在安装之前填写表 2.1,那么,它不仅在安装时便于配置,还可以在出错的情况下解决问题。

表中的大多数信息在特定的设备手册上可以找到,其他信息可以通过系统管理员获取。 如果希望成功安装,一定要获得正确的信息。

待调查情况	调查结果
硬盘个数	
每个硬盘的容量	
主盘	
内存数量	
光盘驱动器类型和数量	
IDE 接口情况	
SCSI 接口情况	
每种光盘驱动器的制造商与型号	
SCSI 适配器的制造商与型号	
鼠标类型	
鼠标键数	
如果是串口鼠标,所连接的串口	
显示卡的制造商、型号及显存数量	
监视器制造商与型号	

表 2.1 安装前的准备工作

(续表)

待调查情况

调查结果

允许的水平刷新范围
允许的垂直刷新范围
网络 IP 地址
网络 IP 掩码
网关地址
域名服务器 IP 地址
域名
主机名
网卡制造商与型号
其他已安装的操作系统

如果使用 LILO, 在何处安装

如果用户对自己拥有什么和想要安装什么都已经很清楚了,那么下面就可以进行安装 了。

2.2 安装设置

下面介绍的安装过程是使用 Red Hat Linux 7.0 的第一张安装光盘进行安装的步骤。按照下面介绍的步骤,可以顺利地完成安装。

将光盘放入光驱中,重新启动计算机,设置 BIOS,使用光驱引导。

可以看到的第一个界面是一个欢迎界面,屏幕中间是一些说明信息,提示可以在屏幕 下面的 boot:提示符后输入<Enter>、text<Enter>、expert<Enter>、linux rescue、<Enter>或者 linux dd<Enter>。

初次安装 Red Hat Linux 7.0 时,输入<Enter>开始进行安装。

系统读入一些信息,然后进入安装界面。用户可以选择安装 Work station 版本、Server 版本或者自己定制系统,下面分别介绍不同版本的详细设置情况。

2.2.1 设置Red Hat Linux 7.0 Work station

(1) Language Selection 界面

这是安装程序的第一个图形界面。整个屏幕分成左右两个部分,左面显示的是在线帮助,单击左下角的 Hide Help 按钮,可以隐藏这个部分,这时右面的内容将扩充到整个屏幕。

在 Language Selection 这个步骤, 左面在线帮助的标题就是 Language Selection。标题下面是关于右面操作的说明。

右面是一个列表框,按照字母顺序列出了 Red Hat Linux 7.0 支持的语言种类,包括 Czech、Danish、English 和 French 等等,很遗憾,没有 Chinese。大多数用户会选择 English。

单击 English 选项,然后单击屏幕右下角的 Next 按钮,进入下一步 Keyboard Configuration 界面。

Next 按钮的左侧有一个 Back 按钮,可以使用它来退回安装。由于语言选择是第一步, 所以 Back 按钮此时无效。

(2) Keyboard Configuration 界面

这一步将配置系统的键盘。此时右面分成 3 个列表框,标题分别是 Model 和 Layout、 Dead Keys。

在 Model 列表框中,列出了支持的键盘模型。可选项有 Brailian ABNT2、Dell 101-Key PC、Everex STEPnot、Generic 101-key PC、Generic 102-key (Intl) PC、Generic 104-key PC、Generic 105-key (Intl) PC 和 Japanese 106-key 等等。应该根据自己拥有的键盘类型进行选择。如果键盘类型没有在上述列表中列出,或者不知道自己的键盘类型,可以选择 Generic 系列(例如 Generic 101-key PC)。事实上,简单地选择 Generic 系列并不会带来坏处。

在 Layout 列表框中,列出了键盘的分布。可选项根据字母顺序排列,有 Belgian、 Brazilian、Bulgarian、Canadian、……、U.S.English、……。一般选择 U.S.English,这也是 默认选择。

在 Dead Keys 列表框中,只有两个可选项, Disable dead keys 和 Enable dead keys。所 谓 Dead Keys,是指一些组合键序列,使用它们可以输入一些特殊的字符。如果需要使用 这些字符,选择 Enable dead keys。

在 Dead Keys 列表框下方,有一个文本框,用于测试键盘配置。可以在其中输入字符, 检查配置是否正确。

键盘配置完毕后,单击 Next 按钮进入到下一步 Mouse Configuraton 界面,或者单击 Back 按钮,重新进行上一步配置。

(3) Mouse Configuration 界面

这一步将配置系统使用的鼠标类型。首先要明确使用的是哪种鼠标,是 PS/2、总线或 者串行鼠标。如果鼠标插孔是圆的,那么是 PS/2 或总线鼠标;如果鼠标插孔是方的,那么 是串行鼠标。

明确了鼠标类型,就可以配置鼠标了。在右面的配置界面中,有两个列表框,分别给出了鼠标类型和串行鼠标所用的串口。在鼠标类型的列表框中,可选项有 ALPS GlidePoint (PS/2), MileMouse (PS/2), MileMouse (serial), ATI Bus Mouse、Generic 2 Button Mouse (PS/2), Generic 2 Button Mouse (USB), Generic 2 Button Mouse (serial), Generic 3 Button Mouse (PS/2), Generic 3 Button Mouse (USB)和Generic 3 Button Mouse (serial), 除此之外,还有诸多的鼠标类型,并且在鼠标类型后面都注明了 PS/2、USB 或者 serial。

根据自己的鼠标类型进行选择。如果不清楚,可以选择 Generic 系列。但是要区分是 2 键还是 3 键,是 PS/2、USB 还是 serial。

如果使用的鼠标是 serial 类型,那么要在下面的列表框中指定使用的 serial 端口。可选的端口有 ttyS0、ttyS1、ttyS2 和 ttyS3,分别对应 DOS 下的 COM1、COM2、COM3 和 COM4。

Linux 的 X Window 可以使用 3 键鼠标的中间键。如果使用的是 2 键鼠标,可以同时按下左右键来模拟中间建。如果在鼠标类型中选择了 2 键鼠标,请单击标题为 Emulate 3 Buttons 的按钮来模拟 3 键鼠标。

配置鼠标结束后,单击 Next 按钮进行到下一步准备安装 Red Hat Linux 7.0,或者单击 Back 按钮回到上一步。

(4) Welcome 界面

这一步什么也不必做,单击 Next 按钮开始安装 Red Hat Linux 7.0。但是应该先阅读左 面的帮助信息。

在 http://www.redhat.com 上有 HTML 和 PDF 形式的安装手册,在安装盘上也有 HTML 形式的帮助。

Red Hat Linux 的安装是无需分区的。如果有一个空间足够大的 FAT 分区,可以直接在 其上安装 Linux 系统。如果想最大限度地保护现有系统,这种方式是最合适的。

在下面的安装过程中,鼠标和键盘可以同时使用(在此之前,只能使用鼠标),可以使 用键盘的 TAB 和 Enter 键进行安装。在进行安装时,如果单击 Next 按钮,就会保存当前的 设置并进行后续安装;如果单击 Back 按钮,会放弃当前设置。

在进行到 Preparing to Install 步骤之前,都可以放弃安装。如果在上述步骤中单击了 Next 按钮,数据就会写到硬盘上。如果想放弃安装,可以按 Reset 按钮或者使用 Ctrl+Alt+Del 键重新启动系统。

(5) Install Type 界面

这一步将会决定所安装的 Red Hat Linux 的基本框架,应该选择是 Install 系统还是 Upgrade 系统。

如果 Install 系统,安装过程将会破坏所选分区的数据。

如果 Upgrade 系统,安装过程将保存原有的 Red Hat Linux 的系统数据。

如果希望 Install 系统,那么必须选择安装的类型。可供选择的安装类型有 Workstation、 Server System 和 Custom System。

如果选择安装 Workstation 版本的 Red Hat Linux 系统,那么将会安装 X Window 系统和所选择的桌面管理器。

如果选择安装 Workstation,安装过程将会破坏和 Linux 分区相关的一切数据,其他系统的分区保持原样。可以在安装 Linux 之后启动原来的系统。

如果希望 Linux 系统可以作为服务器使用,并且不对默认配置做大的改动,那么 Server System 是一个理想的选择。

但是,选择 Server System 意味着所有的现有数据不再被保护。安装过程将会破坏所有要安装的驱动器上的所有分区,包括其他系统使用的分区,所有的分区将被删除。

比较而言, Custom System 具有较大的灵活性。在 Custom System 的安装过程中,硬盘如何分区完全由用户决定。用户可以决定系统安装哪些安装包,同时,还可以决定是否使用 LILO 来启动系统。

一般不应该选择 Custom System 安装方式,除非具有丰富的 Linux 使用经验。

(6)选择 Workstation 版本

选中 Install 下的 Workstation,单击 Next 按钮,会弹出一个提示分区的界面。继续安装, 将会删除原来安装的 Linux 版本。

警告:这种安装方式将会删除所有 Linux 分区上的所有数据。如果不希望如此,可以选择手工分区,或者退回去选择 Custom 的安装方式。

在屏幕右面有 3 个单选按钮,上面是 Automatically partition and REMOVE DATA,下面

两个分别是 Manually partition with Disk Druid 和 Manually partition with fdsik[experts only]。 选择下面两个之一可以在选择下保护已有 Linux 分区的数据。

(7) Disk Druid 分区

选中 Manually partition with Disk Druid,单击 Next 按钮,会出现一个引导用户分区的 界面。屏幕右上是一个现有分区的列表,列出了现有分区的各种信息。屏幕右下是当前所 有分区所占的硬盘空间大小。

如果要执行无分区的安装,就需要指定一个 DOS/Windows 分区为根分区。具体操作如下:

在屏幕右下的分区列表中选择一个 FAT 分区。 单击分区列表下方的 Edit 按钮,出现一个对话框。 在对话框上的 Mount Point 的文本框中输入"/"。 单击对话框的 OK 按钮。

如果是用 fdisk 分区,那么必须为分区定义 Mount Point。如上就可以为分区定义 Mount Point。

如果是用 Disk Druid 分区,可以在此时创建或者删除分区。 下面介绍分区列表中的各字段的含义:

- Mount Point 该字段指出当 Red Hat Linux 安装和运行时,分区应该加载的地方。 如果 Mount Point 的值是 not set,那么可以使用 Edit 按钮来设置。
- · Device 该字段指出分区的设备名。例如,hda2 是主硬盘的第2个分区。
- · Requested 该字段指出分区有多大的空间。如果希望重新指定大小,就要删除该分区,然后用 Add 按钮重新创建。
- · Actual 该字段指出分区可以使用空间的大小。
- · Type 该字段指出分区的类型。

可能在某些分区之后看到一个 Unallocated Requested Partition 的红色提示。出现这一提示的原因一般是没有足够的自由空间分配给分区。解决这一问题的办法有:将该分区放在 其他硬盘上;重新指定分区的大小;删除该分区。

下面介绍分区列表下面的按钮的作用:

- · Add 用于申请新的分区。单击 Add 按钮,将会出现一个对话框,提示用户输入 必须的信息,如 Mount Point、Size、Partition Type 等。
- · Edit 用于改变选中分区的 Mount Point。
- · Delete 用于删除选中分区。
- · Reset 用于取消所作的修改。
- · Make RAID Device 只有在有使用 RAID 的经验下,才可以使用这个按钮。

下面介绍屏幕右下的信息:

- · Drive 显示硬盘设备名字。
- · Geom[C/H/S] 显示硬盘的柱面、磁头和扇区数。

· Free 显示硬盘的空闲空间。

· Used 显示硬盘上的已分配空间。以 M 为单位显示或以百分比显示。

当决定了如何分区后,单击 Next 按钮,将进行分区的格式化。

(8) Format Partition

刚才指定的 Mount Point 的分区现在再次出现。可以选择是否对它们进行格式化,是否 检查坏块。单击 Next 按钮,将进行网络配置。

(9) Network Configuration 界面

如果有网卡,屏幕的右上是以网卡的设备名为标题的选项卡。该选项卡上有两个可选项, Configure using DHCP 和 Activate on boot。

如果自己指定 IP 地址,那么保持 Configure using DHCP 不被选中,否则不能指定 IP 地址;如果希望在系统启动时激活网卡,那么选中 Activate on boot。

选项卡下面是 4 个文本框,分别提示输入 IP Address、Netmask、Network 和 Broadcast 信息。

其次是 Hostname 文本框,可以指定主机名,如果空缺,主机名将是 localhost。

最后是 4 个文本框,提示输入 Gateway、Primary DNS、Secondary DNS 和 Ternary DNS。可以有选择地填写,也可以不填。如果感兴趣,可以阅读本书介绍网络配置的章节。

单击 Next 按钮,将进行时区的选择。

(10) Time Zone Selection 界面

可以根据计算机的物理位置选择时区,也可以根据当地时间域与通用坐标时间(UTC, Universal Coordinated Time)的偏移量来选择时区。

在屏幕右面有两个选项卡,标题分别为 Location 和 UTC Offset。在 Location 选项卡中,可以根据计算机的物理位置选择时区。默认情况下,Location 选项卡中是一张世界地图,单击地图上方 World 按钮,可以选择局部地区。可选择的项有 World、North American、South American、Pacific Rim、Europe、Africa 和 Asia。例如选择 Asia, Asia 地图将代替 World 地图。

在选中的地图中,有许多黄色小点,代表可选的地名。单击某个地名,就选中了当地 时间。

很遗憾,这些地名中没有北京,所以可以选择上海(ShangHai)。

在 Location 选项卡的下部,是一个列表框。当选中一个地名后,该地名将显示在列表框中。

在 UTC Offset 选项卡中,可以直接指定时区。屏幕右面是一个列表框,列出了从 UTC-14 到 UTC+12 之间的各个时区。对于北京时间,应该选择 UTC+8。

选择时区后,单击 Next 按钮,将进行帐号设置。

(11) Account Configuration 界面

屏幕右面分成上、中、下 3 部分,上面部分是 root 帐号的设置,中间部分是普通帐号 的设置,下面部分是已设置的帐号的列表。

root 帐号设置部分有两个文本框,分别是 Root Password 和 Confirm。在 Root Password

中填写 root 口令,长度不少于6个字符。在 Confirm 中填入相同内容。只有 Root Password 和 Confirm 中的内容吻合, Next 按钮才可用。

普通帐号可以不设置。这里有 4 个文本框,分别是 Account Name、Password、Password (confirm)和 Full Name。Account Name 对应于用户名, Password和 Password(confirm) 对应于用户口令和确认。Full Name 对应于文字说明。当填写了 Account Name 项,并且 Password和 Password(confirm)内容相符,下面的 Add 按钮就有效了。单击 Add 按钮, 可以看到帐号列表框中添加了一项。

如果还想添加帐号,单击 New 按钮,然后重复上面的动作。

对于从帐号列表框中选定的帐号,单击 Edit 按钮可以编辑帐号,单击 Delete 按钮可以 删除帐号。

当 Root Password 和 Confirm 内容相符时,单击 Next 按钮,将进行安装包选择。

(12) Package Group Selection 界面

屏幕右面有 3 个可选项,分别是 GNOME、KDE 和 Games。单击任意一个可以安装该包。如果要进行单个安装包的选择,单击屏幕右下方的 Select individual packages 选项。单击 Next 按钮,将会出现一个和前面不同的界面,原来在屏幕左面的 Help 被自动隐藏了。

这时屏幕左面是一个目录树。一级目录有 Amusements、Applications、Development、 Documentation、System Environment、User Interface。单击一级目录左侧的+号,可以展开 二级目录。单击二级目录中的任意一项,都会选中该目录。右面是目录树被选中目录下的 内容图标,每个图标代表一个独立的安装包。

可以看到,有的图标上有一个红色的对勾,有的没有。红色对勾表示该安装包已被选中。单击任意一个图标,然后单击下面 Select Package For Installation 选项,可以在选中和不选中该安装包之间切换。

在屏幕下面是一个文字显示界面,对应于上面选中的安装包的说明。当不能确定是否 选择安装该安装包时,应该阅读其说明。

选择了安装包之后,单击 Next 按钮,将进行显示器配置。

(13) Monitor Configuration 界面

屏幕右面是一个列表框,列出了 Red Hat Linux 7.0 支持的显示器。当进行到这一步时, 一般会有一个显示器被选中。该显示器是系统检测到的显示器类型,一般这个配置是正确 的。如果有异议,可以自己从列表框中选择一个型号。

在列表框的下面有两个文本框, Horizontal Sync 和 Vertical Sync, 分别用来指定显示器的水平和垂直刷新频率。

指定显示器的水平和垂直刷新频率是有危险性的。如果指定的数值超出了显示器的承 受范围,将会造成损坏。显示器的水平和垂直刷新频率在说明书中有说明。

选定了显示器,单击 Next 按钮,将进行 X 配置。

(14) X Configuration 界面

这一步看起来比较复杂,但是不必担心,只要了解硬件类型,就可以顺利过关。

屏幕右上是一个列表框,给出了 Red Hat Linux 7.0 支持的显卡类型。当进行到这一步时,一般会有一个显卡被选定。这个显卡类型是系统检测到的显卡类型,一般这个选择是正确的。如果有异议,可以自己从列表框中选择一个型号。

列表框下方是若干个单选按钮,用于指定上面指定显卡所配备的显存容量。应根据自 己的情况进行选择,不可轻信系统的选择。

在指定了显卡类型和显存容量之后,应该测试所选是否正确。单击 Test this Configuration 按钮即可。如果正确,屏幕将在短暂的黑暗之后出现一个对话框,询问用户 是否可以看见该对话框。如果看到了这个对话框,说明上面的配置正确。

屏幕下部有一个 Default Desktop 按钮,用于指定系统默认的桌面环境。可选项有 GNOME 和 KDE。默认的桌面环境是 GNOME。

屏幕下部还有3个选项, Customize X Configuration、Use Graphical Login 和 Skip X Configuration。Customize X Configuration 选项用来指定是否想定制 X 配置,包括显示分辨 率和颜色深度等内容。建议选定该选型进行定制。Use Graphical Login 选项用来指定系统登录界面是文本界面还是图形界面。选中该选项则选择了图形登录界面。Skip X Configuration 用来指定是否跳过上述所有 X 配置。单击该选项,上述界面变成灰色无效。

选择 Customize X Configuration 选项之后,单击 Next 按钮,将进行 X 的定制。

(15) Customize X Configuration 界面

屏幕右面有许多的按钮,指定了不同的显示分辨率和颜色深度。可选的显示分辨率有 640×480、800×600、1024×768、1280×1024、1600×1200,可选的颜色深度有 8 Bits per Pixel、16 Bits per Pixel和 32 Bits per Pixel。

应该在允许的范围内指定这些参数,然后单击 Test this configuration 按钮来测试。

如果屏幕在短暂的黑暗之后出现了一个询问是否可见该对话框的对话框,那么所作的 选择是可接受的。如果对刚才所作的选择不满意,可以选择 No 来重复配置。

定制了 X 之后,单击 Next 按钮将进行安装阶段。

(16) About to Install 界面

现在只有屏幕右下有两个按钮, Back 和 Next 按钮。

如果确认刚才所有的配置没有问题,那么可以按下 Next 按钮进行实质安装。

注意:这一步是不可逆的,一旦按下 Next 按钮,安装程序将在硬盘上写入数据。 现在是放弃安装的最后机会。

如果想放弃安装,可以按下 Ctrl+Alt+Del 组合键,或者 Reset 按钮。

2.2.2 设置Red Hat Linux 7.0 Server System

安装 Server System 版本与安装 Word Startion 版本的第(1)~第(5)步,第(7)~第(11)步完全相同,下面只介绍步骤不同的情况。

(6)选择 Server System 版本

选中 Install 下的 Server System 选项,单击 Next 按钮,前进到分区界面。

(12) Package Group Selection 界面

屏幕右面有4个选项,分别是 News Server、NFS Server、Web Server和 DNS Name Server。 如果希望安装其中任何一个服务器,选中该选项即可。在屏幕右下有一个 Select individual packages 选项,选中该选项可以在这些服务器中进行单独安装包的选择。

选中 Select individual packages 选项,单击 Next 按钮,进行单独安装包的选择。该界面

和上面介绍的 Workstation 的界面相同。

(13) About to Install 界面。

选择了安装包之后,单击 Next 按钮,开始正式安装程序。这里的警告和上面 Workstation 的警告相同。

2.2.3 定制Red Hat Linux 7.0

定制安装的第(1)~第(5)和第(7)步与 2.2.1 节对应的步骤完全相同。下面只介绍 步骤不同的情况。

(6) 选中 Install 下的 Custom System 选项,单击 Next 按钮,前进到分区界面。

(8) Format Partition 界面

刚才指定的 Mount Point 分区现在再次出现。可以选择是否对它们进行格式化,是否检 查坏块。

单击 Next 按钮,下面进行 LILO 配置。

(9) LILO Configuration 界面

LILO(LInux LOader)是一个用于引导 Linux 系统的软件,它也可以用于引导 Windows 操作系统。有关 LILO 的详细介绍参见本书第3章。这一步用来指导用户配置 LILO。

屏幕右面垂直分成 3 部分,第1 部分是 LILO Configuration,它有两个选项, Create boot disk 和 Do not install LILO。

如果不想把 LILO 写在硬盘上,可以跳过安装 LILO。如果有两块硬盘,每个硬盘上有 一个系统,那么可能不把 LILO 装在硬盘上,而是装在软盘上。

如果选择 Do not install LILO 选项,那么以下内容无效。

第 2 部分是 Install LILO boot record on,它有两个复选框:/dev/hda Master Boot Record (MBR)和/dev/hda8 First sector of boot partition,还有两个文本框:Use linear mode(needed for some SCSI drives)和 Kernel parameters。

如果选择安装 LILO,那么要决定把 LILO 装在何处。如果计算机上只有 Linux 系统,应该把 LILO 装在 MBR 上。如果 Windows 9x 系统和 Linux 在同一个硬盘上,也可以把 LILO 装在 MBR 上。

如果使用 Windows NT 系统,并且想安装 LILO,那么推荐把 LILO 装在启动分区的第 一个扇区上。并且,应该制作启动盘,以防不能进入 Linux 系统。

对于 Use linear mode 选项,保持其默认状态。

Kernel parameters 信息将在系统启动时传递给内核,一般情况下请将该文本框保持空白。

第 3 部分默认引导系统。如果此时计算机上已经有了一个 Windows 操作系统,就会让选择默认的操作系统是 Windows 还是 Linux。

该部分下方的列表中列出了系统现有的操作系统,包括将要安装的 Red Hat Linux 7.0。 选中其中一项,Boot label 文本框中将显示一个字符串,一般是 linux 或者 dos。例如,如果 在 LILO 阶段想要启动 linux,那么在输出 LILO:提示符后输入 linux。

选中一个操作系统,单击 Default boot image 选项,在操作系统前将会出现一个绿色对

勾,表示该操作系统是未干预情况下的默认操作系统。

完成 LILO 配置后,单击 Next 按钮,进入网络配置。

第(10)步与2.2.1节的第(9)步完全相同。

第(11)步与2.2.1节的第(10)步完全相同。

第(12)步与 2.2.1 节的第(11)步基本相同,只是在最后当 Root Password 和 Confirm 内容相符时,单击 Next 按钮,将进行认证配置。

(13) Authentication Configuration 界面

如果不使用网络口令,可以跳过这一步,直接单击 Next 按钮。

在屏幕右面有 5 个选项,分别是 Enable MD5 passwords、Enable shadow passwords、 Enable NIS、Enable LDAP 和 Enable Kerberos。

在默认的状态下,前两个可选项是选中的。推荐激活 MD5 和给 shadow 口令加密。MD5 允许使用长达 256 个字符的口令, shadow 使口令保存在/etc/shadow 文件中,并且只对 root 可读。

如果想要使用 NIS,必须连接到 NIS 网络。如果不清楚,请询问系统管理员。使用 NIS,可以使用通用的口令在一个 NIS 域中共享信息。NIS Domain 文本框用于指定用户属于哪个 NIS 域, NIS Server 文本框用于指定特定的 NIS 服务器,而不是在一个局域网内广播查找 NIS 服务器。

LDAP 用于加强一个组织内的特定信息分类。LDAP Server 用于指定运行 LDAP 协议的服务器。LDAP Base DN 用于指定 DN (Distinguished Name),该 DN 用于查找用户信息。

Kerberos 用于提供网络认证。Realm 文本框用于指定使用 Kerbeos 的网络,该网络由一个或者多个服务器(KDC)和许多客户端组成。

KDC 文本框用于指定密钥发布中心(KDC, Key Distribution Center),该中心发布密钥。 Admin Server 文本框用于指定 kadmind 服务器。

如果对这一步骤的内容不明白,可以不予理会,但是推荐选中 Enable MD5 passwords 和 Enable shadow passwords 选项。

单击 Next 按钮,进行安装包选择。

(14) Package Group Selection 界面

屏幕右面是一个非常漂亮的列表,列出了可以选择安装的包。它们从上到下依次是: Printer Support、X Window System、GNOME、KDE、Mail/WWW/News Tools、DOS/Windows Connectivity、Graphics Manipulation、Games、Multimedia Support、Laptop Support、Networked Workstation、 Dialup Workstation、 News Server、 NFS Server、 SMB(Samba) Server、 IPX/Netware(tm) Connectivity、Anonymous FTP Server、Web Server、DNS Name Server、 SQL Server、Network Management Workstation、Authoring/Publishing、Emacs、Development、Kernel Development、 Utilities 和 Everything。

可以从名字判断出大部分安装包的内容,根据需要进行取舍。

屏幕下方有一个 Select individual packages 的选项,允许用户进行更加详细的选择。 单击 Next 按钮,进入正式的安装阶段,请参考安装 Workstation 部分的警告。

2.3 安 装

当用户选择了任意一种安装方式,并完成了安装配置选择后,安装程序自动开始运行, 系统开始复制和安装用户刚才选定的安装包,这个过程的时间长短取决于用户选择的安装 包的数量和用户计算机的配置。对于 PIII 级别的计算机,这个过程一般小于 30 分钟。

在正常情况下,安装过程不需要用户干预。

当安装完毕之后,系统会询问用户是否制作启动盘,推荐用户制作启动盘,以防不测。 如果用户不想制作启动盘,可以选择不制作启动盘。

安装过程结束后,系统顺序关机,然后又重新启动。用户应该适时把光驱中的安装盘 取出,并把 BIOS 设为硬盘启动。

第3章 使 用 LILO

LILO 是 Red Hat Linux 伴侣,是安装 Linux 不可缺少的部分。

本章详细介绍了 LILO (LInux LOader)的使用情况,包括多重引导程序的概念介绍, LILO 的安装位置的选择,如何安装 LILO 以及如何配置 LILO 等。还介绍了常见的一些错 误和错误诊断。

通过对本章学习,可以对 LILO 有一个透彻的了解,可以解决因为 LILO 引起的大多数问题。

3.1 多重引导加载程序简介

多重引导加载程序指一个允许在引导启动机器时对使用哪一种操作系统进行选择的程序。无论计算机中只有 Linux 操作系统还是安装了多个操作系统,都必须安装多重引导加载程序。这是因为计算机只有在知道系统内核(通常也就是这个操作系统本身)存放在什么位置的情况下才能够正确地引导启动。

表 3.1 是一些多重引导加载程序。

表 3.1	多重引	导加载程	序及来源

多重引导加载程序	来源
System Commander	www.v-com.com
NT OS Loader	NT 发行盘
OS/2 的多重引导加载程序	OS/2 发行盘
Partition Magic	www.powerquest.com

Linux 有自己的引导加载程序,用来把 Linux 内核装入计算机。使用哪个程序取决于计算机:对于 Intel 兼容的 PC 机,使用 LILO;对于 DEC 公司的 Alpha PC 机,使用 MILO;对于 SPARC 兼容的工作站,使用 SILO。下面将集中介绍 LILO。LILO,按照它的开发者 Werner Almesberger 的观点,代表 LInux LOader。

3.2 安 装 LILO

安装 LILO 涉及到几个方面的问题 ,主要的问题包括如何选择 LILO 程序的安装位置和 如何安装 LILO。下面将对这两个问题作详细介绍。
3.2.1 LILO程序安装位置

LILO 程序可以被安装在系统中的许多地方。每一种选择都有它各自的优点与缺点,下面分别介绍。

1. 主引导记录

最简单的选择就是把 LILO 程序安装在主硬盘的 MBR(主引导记录)处。对于只在 Linux 操作系统中使用的计算机这是最直接的方法。

如果 Linux 是计算机上仅有的操作系统,或者已经安装了 Windows 98 或 Windows NT, 并把 LILO 安装在引导驱动器的 MBR 处。如果还装有 OS/2,则要把 LILO 安装到硬盘的根 分区中,并使用 MBR 上 OS/2 的引导装载器。

如果在安装 Linux 的计算机上还打算再安装一些操作,如 Windows 98, 它将覆盖 MBR, 并把原来安装在 MBR 中的 LILO 程序删除掉。如果已经制作了 Linux 操作系统的应急引导 启动盘就可以解决这个问题。所以,如果希望将计算机用作双引导系统,则需要首先安装 Microsoft 类型的产品,然后使用 LILO 覆盖 MBR,或者使用其他类型的软件(如 Partition Magic)执行类似的工作。

2. Linux 操作系统的 root 根分区

在同一个硬盘上安装多个操作系统或者在多个硬盘上分别安装不同的操作系统,一般 会选择把 LILO 程序安装在 Linux 操作系统的 root 根分区上。

把 LILO 程序安装在这个位置可以避免在使用 MBR 的操作系统之间发生冲突。

如果在 Linux 操作系统的根分区安装 LILO 程序,就必须使用进行分区操作的工具程 序把它设置为可引导的。

3. 软盘

如果不打算把 Linux 操作系统当作主操作系统,或者对把 LILO 程序安装在硬盘上可 能会出现的问题非常重视,建议把 LILO 程序安装到一张软盘上。这样,只有在把这张软 盘插入到软驱中的时候才能运行多重引导启动程序。同时,一定要制作一份 LILO 程序的 备份,并把它保存到安全的地方。

如果在运行 Linux 的计算机上有多个操作系统,务必要把每一种的应急引导启动盘都 准备好。即使永远也许不会用到它们。准备好这些盘就意味着万一在安装 LILO 程序时出 现了问题,也不会出现不可收拾的情况。

3.2.2 安装LILO

下面将介绍如何安装 LILO。如果第一次安装 Red Hat Linux 7.0 时选择的是不安装 LILO,或者在使用某些内核选项方面有疑问,下面的内容可提供帮助。

通过使用 lilo 命令(位于/sbin 目录下)很容易安装 LILO,但还是应该先花些时间阅读 一下它的文档。这些文档可在/usr/doc 目录下找到。还可以找到一个称为 QuickInst 的 shell 脚本程序,它可以替换现有的 LILO 安装或进行第一次安装。LILO 的文档包含关于它的特 性的详细内容,并为特殊问题提供了一些重要的技巧和解决途径,例如在大容量的硬盘上 安装引导装载器或从其他操作系统引导。 在安装 LILO 之前,必须知道 Linux 分区的位置。如果机器上还有其他操作系统,也 必须知道其位置。在 Linux 下,硬盘驱动器被抽象为/dev 目录下的设备文件。如果有一个 或多个 IDE 驱动器,则第一个硬盘驱动器被当作/dev/hda,第二个硬盘驱动器被当作 /dev/hdb。SCSI 驱动器则分别被当作/dev/sda 和/dev/sdb。当安装 Linux 时,很可能要对硬 盘进行分区,则第一个硬盘上的第一个分区将被当作/dev/had1 或/dev/sda1。依此类推,第 二个分区将被当作/dev/hda2 或/dev/sda2。在配置 LILO 之前,应该知道哪个分区装有什么 操作系统,并且还应该知道要在哪里安装 LILO。

在试图做任何关于 LILO 的事情之前,都应该有一张紧急引导盘。具有一个不能引导 的系统不是一件好事情,而且,如果没有引导盘,是无法恢复与改变内容的。因此,花一 些时间为自己做一张引导盘可以避免许多麻烦。

安装 LILO 包括两个基本步骤:

(1) 配置/etc/lilo.conf。

(2)运行/sbin/lilo 来安装 LILO 并使它激活。

下面将介绍对现有的 lilo.conf 文件的修改。在对此文件做任何修改之前,先在同一目 录或单独的磁盘上为此文件创建一个备份。下面几个文件对 LILO 非常重要,它们是在初 始安装时生成的:

- · /sbin/lilo 映像安装程序,要获得更多信息请使用 manlilo 命令。
- · /boot/boot.b 引导装载器。
- · /boot/map 引导映像,它包含内核的位置。
- · /etc/lilo.conf LILO 的配置文件。

现在准备编辑 LILO 的配置文件/etc/lilo.conf。

编辑 lilo.conf 比较容易。首先,必须以 root 身份进入系统,然后把 lilo.conf 装入编辑器,最后确保对所做的修改进行了保存并把此文件保存为 ASCII 文本。

在以下几种情况下需要编辑 lilo.conf:

1. 正在测试新的内核,并且想要能够引导具有多个内核的同一 Linux 分区。通过在 lilo.conf 文件的 image = 部分使用多个项目可以解决。可能在 Linux 分区安装有多个内核, 通过输入其名字引导不同的内核在 label = 部分指定。

- 2. 想要对某个分区添加口令保护。
- 3. 具有一个需要指定特殊选项的硬件设置,例如引导一个远程文件系统。
- 4. 内核名称不是/vmlinuz 或者不在标准的地方,如/etc。

例如,如果使用的是 vi 文本编辑器程序,请按照下面的步骤进行:

- (1) 以根用户身份登录进入系统。
- (2) 输入 vi/etc/lilo.conf 命令编辑这个文件。
- (3)使用方向键把光标移动到需要添加内容的地方。
- (4) 按下 i 键进入 vi 的插入模式。
- (5)进行必要的修改。

(6) 按下 ESC 键退出插入模式。

(7)按下冒号(:)键进入编辑程序的命令模式。

(8) 输入 wq 再按下回车键保存并退出文件。

下面是一个 lilo.conf 的示例。

boot=/dev/hda2 map=/boot/map install=/boot/bootb prompt timeout=50 images=/boot/vmlinuz-2.2.16 label=linux root=dev/hda2 initrd=/boot/initrd-2.2.16.img read-only

other=/dev/hda1 label=dos table=/dev/hda

3.3 配 置 LILO

掌握 LILO 的难点之一在于正确地配置 LILO。通过配置 LILO,可以实现引导多个操 作系统、设定启动引导延时、设定缺省的操作系统等功能。本节将详细介绍这些内容。 本节还介绍了如何删除 LILO 等内容。

3.3.1 引导多个操作系统

可以配置 LILO 程序引导多个操作系统。按照下面的步骤在 LILO 程序中添加一个操作 系统的引导启动数据项:

(1) 打开/etc/lilo.conf 文件。

(2) 找到与下面的内容类似的部分:

image=/boot/vmlinuz label=linux root=/dev/hda2

对上面数据段中 label 项等号后面的内容改为自己喜欢的文字,下一次开机时 LILO 程序给出的操作系统选项就将是修改后的内容了。例如,把上面的 label=linux 修改为 label=RedHatLinux。

(3) 把光标移动到 image 数据段下面的行, 添加新的内容。

如果要添加的是一种非 UNIX 的操作系统,请不要使用 image 参数,要使用另外的参

数。例如,假定在硬盘 hda1 上已经有了一个 Windows 分区,在硬盘 hda2 上又刚刚安装了 Linux 操作系统。那么,新的数据段应该用这个语句开始:other=/dev/hda1。如果需要添加 的是另外一个 Linux 操作系统分区,就必须使用与第(2)步中同样的结构。

(4)现在要告诉 LILO 程序在分区表的什么位置可以找到这个操作系统。下一行可能 会是 tabel=/dev/hda。

(5) 最后给这个操作系统加上一个名称,如 label=Windows。

(6)保存并退出这个文件。

(7)运行 LILO 程序使修改内容生效。

3.3.2 启动引导延时

可以设置用来选择某个操作系统的启动引导延时。这需要在/etc/lilo.conf 文件开始部分的 boot 语句下加上一条新语句。计时单位是 1/10 秒。例如,如果打算设置引导延时为 5 秒,这个语句应该是"delay=50"。设置完成后需重新运行 LILO 程序。

3.3.3 缺省的操作系统

可以在 LILO 程序中设置一个缺省的操作系统。把某个操作系统的 image 与其他部分 挪到 LILO 配置文件的最开始即可。如果在引导启动延时中没有进行任何选择, LILO 程序 就把机器引导为排列在第一位的操作系统中去。请记住:在修改设置之后要重新运行 LILO 程序,否则修改不会生效。

3.3.4 配置选项

可以把下面表中所列的选项添加到 lilo.conf 文件中。这些选项也可以在引导提示处提供,但把它们放在/etc/lilo.conf 文件中会更简单一些。表 3.2 只列出了 LILO 的 23 个选项中的 13 个,详细资料请参见 LILO 的文档。

配置选项	作用
Boot=<引导设备>	把包含引导扇区的设备的名字告诉内核。 如果 boot 被省略,引导扇区将从当前
	被加载为 root 的设备上读取
Linear	生成线性的扇区地址来代替扇区/磁头/柱面地址。它可能比较麻烦,特别是当
	同 compact 选项一起使用时。其详细资料参见 LILO 的文档
Install=<引导扇区>	安装指定的文件作为新的引导扇区的内容。如果 install 被省略,/etc/lilo/boot.b
	作为缺省文件被使用
Message=<消息文件>	可使用它来显示文件的文本并设置引导提示,消息最长可达 65535 个字节。如
	果改变此文件,重新运行/sbin/lilo
Verbose=<级别>	打开进度报告。代表级别的号码越大,输出越多。号码的范围从1~52。它还有
	-v 和-q 选项,详细资料参见 LILO 文档
Backup=<备份文件>	把初始引导扇区的内容拷贝到<备份文件>(它也可以是一个设备,如/dev/null)
	而不是/etc/lilo/boot.x ,其中 x 为序列号

表 3.2 /etc/lilo.conf 的配置参数

	(续表)
配置选项	作用
Force-backup=<备份	类似 backup,此选项将覆盖当前的备份拷贝。如果使用 force-backup,backup
文件>	将被忽略
Prompt	要求输入引导提示条目
Timeout= <tsecs></tsecs>	为键盘输入设置等待时间(以 1/10s 为单位)。如果想立即引导或要求等待的
	时间比缺省的 5s 更长,使用它则会很方便。提示:想要 LILO 不等待键盘输入,
	用 - 1 这个值
Serial = <参数>	允许从指定的串行装置和 PC 键盘输入到 LILO , 在控制台上按下 Shift 键可以
	模拟串行装置上的中断。为了安全,当使用此选项时,用口令保护所有的引导
	映像。参数的语法为: <port> , <bps><parity> <bits> , 如/dev/ttyS1 , 8N1。参</bits></parity></bps></port>
	数 <bps>、<parity>和<bits>可以省略。如果这些参数中的一个被省略,则它后</bits></parity></bps>
	面的参数也必须省略。另外,如果只指定有端口号,则逗号必须被省略。详细
	内容参见 LILO 的文档
Ignore-table	忽略坏的分区表
Password=<口令>	使用此口令保护引导映像
Unsafe	这一关键词放在分区的定义之后,它通知 LILO 不要试图读取 MBR 或磁盘的分
	区表项。可以把自己系统的全部分区声明为所有现有分区系统的日志,然后放
	上关键词 unsafe 来阻止 LILO 对它进行读取

当对 lilo.conf 做过修改之后,一定要运行/sbin/lilo。当安装新的内核之后,也总是应该运行/sbin/lilo。

3.3.5 确定磁盘参数

任何时候,只要发现与硬盘参数有关的错误,那就可能是没有按照系统内核预期的方 式对它进行设置。

加上 linear 配置变量。如果使用命令行方式进行配置,加上"-1"参数起到同样的效果 (如输入"-124/56/512")。然后重新引导启动机器,看看能不能解决问题。

请按照下列步骤输入硬盘参数:

(1) 记下 BIOS 中硬盘的十六进制代码。这个代码将做为错误信息的组成部分显示出 来。错误信息的格式可能会是下面所示的两者之一:

geo_query_dev HDIO_GETGEO(dev0x50) Device 0x50 Got bad geometry 24/48/248

如果错误信息是第2种格式,其最后的3个数字就是扇区/磁头/柱面。记下这些数字直 接跳到第(5)步。

(2)引导启动到 MS-DOS。

(3)在 MS-DOS 下运行下面的程序:

DPARAM.COMHexErrorCode

如: DPARAM.COM0x50

(4) 记下 DPARAM.COM 程序的运行结果。

(5) 使用应急引导盘启动到 Linux 操作系统。

(6) 打开 LILO 程序的设置文件。

(7)对于需要定义的每一个硬盘,都要在 options(选项)部分有一个硬盘说明语句。 这个硬盘说明语句告诉 LILO 程序该硬盘是哪一种文件系统设备。例如,如果是在定义第 一个硬盘,它在分区程序中显示为 hda,这个语句将会是:

"disk=/dev/hda"

(8)现在使用从第(1)步和第(4)步收集到的数据在硬盘数据段中定义硬盘参数。 如果需要定义不止一个硬盘,就必须使用一个新的硬盘说明语句在第一个硬盘之后另起一 个硬盘数据段。但现在的主要问题是让 LILO 程序正常地引导启动,因此把注意力集中在 保存 LILO 程序的硬盘上。

Bios=

Sectors=

Heads=

Cylinders=

如果没有给出 cylinders(柱面)的数字, LILO 程序或者从系统内核获得它需要的信息, 或者就假设该硬盘有 1024 个柱面。因此最好是尽可能给出这个信息。例如,使用从第(1) 步记录的数字,可能需要输入下面的语句:

disk=dev/hda bios=0x50 Sectors=24 Heads=48 Cylinders=248

(9)保存这些设置,运行 LILO 程序,然后重新引导启动计算机。如果 LILO 仍然无 法引导启动,继续下一步操作。

(10)作为硬盘说明的一部分,还可以把每个分区都加以说明。从 cylinders 语句下面 开始,先定义要说明的是哪一个分区。例如,hda(IDE 驱动器)或者 sda(SCSI 驱动器) 硬盘的第一个分区应该是:

partition=/dev/hda1

(11) 另外一个可以使用的信息就是该分区是从哪个扇区开始存放的。这个信息可以 使用 FDISK 或者其他硬盘分区工具程序查到。例如:start=2048

(12)保存这些设置,运行 LILO 程序,然后重新引导启动计算机。

3.3.6 删除LILO和恢复MBR

1. 删除 LILO 程序

以根用户身份登录进入系统,再输入 lilo-u 命令就可以从系统中删除 LILO 程序。

2.恢复 MBR

输入 fdisk /mbr 命令恢复 MBR 原来的内容。

3.3.7 代替LILO的方案

即使 LILO 程序没有安装在 MBR(主引导记录)中,甚至在根本就没有安装它的情况下,也有许多引导启动 Linux 操作系统的其他方法。下面介绍用 Loadlin 程序引导 Linux 系统。如果选择这种方法,那么在安装过程结尾当回答把 LILO 程序安装到什么地方的问题时,请选择 Skip 选项。

LOADLIN.EXE 是一个使用 DOS 的 MBR 引导的 Linux 程序,它也可传递内核选项。 Loadlin 避免了修改 MBR,很多人喜欢使用 Loadlin 程序。使用 Loadlin 需要在硬盘上 有一个包含这个程序的 DOS 分区。请按照下列的步骤安装和使用 Loadlin 程序:

(1)把LOADLIN.EXE 文件拷贝到一个 MS-DOS 分区。

(2)在安装 Linux 操作系统之后,把内核映像/boot/vmlinuz 文件拷贝到同一个分区中的根目录(如C:)。

(3)在 MS-DOS 分区的提示符处输入与下面的内容类似的命令就可以引导 Linux 操 作系统启动。

loadlin c:\vmlinuz root=/dev/hda1 ro

在命令行中一定要插入 root 分区, ro 用来指定为只读。当第一次启动一个 Linux 分区时, 它应该被加载为只读, 否则可能会引起数据丢失。

如果具有 UMSDOS 文件系统,则可以输入:

loadlin c:\vmlinuz root=/dev/hda1 rw

rw 用来指定为可读可写。使用这一方式启动 UMSDOS 系统是安全的,此外,一定把 自己的分区替换进去。

3.4 引导Linux系统

在正确地安装和设置了 LILO 之后,就可以使用 LILO 来引导 Linux 系统了。除了正常 启动之外,还可能遇到引导故障。本节对引导故障进行诊断并给出相应的排除方法。

3.4.1 正常启动

请按照下面步骤引导进入 Linux 操作系统:

(1)重新启动计算机。如果把 LILO 程序安装到软盘上了,在重新启动计算机之前一 定要把这张盘放到软驱中。

(2) LILO 程序被激活运行。如果在这一步安装被挂起(将会看到 L、LI、LIL 而不是 LILO), 请参考 3.4.3 节的内容。

(3)当 LILO 完全调入之后,屏幕上将出现它的引导提示符:LILO boot: 如果是下面的任意一种情况,就不需要在引导提示符后输入任何内容:

- · Linux 操作系统是该计算机上唯一的操作系统。
- · Linux 操作系统是 LILO 程序缺省引导的操作系统。
- · 第一次引导 Linux 操作系统,还没有完全设置好 LILO 程序。

除了上面这几种情况外,请输入希望使用的操作系统名称并按下回车键。

3.4.2 LILO引导提示选项

表 3.3 列出的选项可以在引导提示时传递给 LILO,从而实现系统的某些特殊功能,或 者把这些选项传递给 Linux 内核来实现系统的正确引导。在 Red Hat Linux 的安装过程中, 了解系统的一些必要选项会带来很多方便。因为在安装时如果选择安装 LILO,便会要求输 入某些特殊选项。

虽然在 LILO:的提示下一般输入 Linux 或 dos,但也可以试用如表 3.3 所示的选项。 如果不记得在 lilo.conf 文件中为 LILO:提示符定义的确切标签,可以按下 Tab 键让 LILO 输出可用内核列表。如果无效,则可以在 LILO:提示符出现之前按下 Alt 或 Shift 键。

LILO 选项	作用
rescue	引导 Linux 进入单用户模式,以允许对系统进行修复
single	同 rescue 类似,但试图从硬盘引导
root = <设备>	同/etc/lilo.conf 中的该条目类似,此选项可以从光盘或其他存储设备上引导
vga=<模式>	改变控制台的分辨率

表 3.3 LILO 选项

3.4.3 诊断使用启动引导时挂起错误

使用 LILO 一般不会出现问题,但是如果有,可能会得到 70 种不同的警告或错误消息 之一。当 LILO 显示在屏幕上的时候,每一个字母都代表引导启动过程的一个组成部分。 如表 3.4 所示是 LILO 引导过程中可能发生的每一种错误,以及部分错误代码和引起它们的 原因。

LILO 程序可以从软盘移到 MBR,也可以从 MBR 移到软盘。在考虑这样做的时候,如果能够把安装 Linux 时建立的 root 分区设置为可引导(使用硬盘分区工具程序)的,那 么还可以把 LILO 程序移到 root 分区中去。

磁盘错误码能够指出如软盘驱动器门为开、驱动器超时、控制器故障、介质故障、BIOS 故障或者瞬时读错误。总之,LILO的一般问题包括:

- · 在内核更改之后没有重新运行/sbin/lilo。
- · 在创建新的引导映射时,/sbin/lilo 使用不正确。
- · 从一个非常大的分区安装和引导 Linux。
- · 在安装 Linux 和 LILO 之后安装了其他操作系统。

- · 手动编辑/etc/lilo.conf 文件有错误。
- · MBR 损坏。
- · 在 Linux 交换分区安装 LILO。
- · 找不到 Linux 内核映像。
- · 在 DOS 分区上安装并引导 Linux, 然后进行 DOS 分区的碎片整理。
- · 在 LILO:提示时输入错误的内核消息。

如果遇到了问题,一定要阅读/usr/doc/lilo 目录中 Almesberger 的 readme 文件。

表 3.4 使用 LILO 程序诊断启动引导时出现的挂起错误

LILO 阶段	诊断和建议
空白	LILO 无法运行加载。原因可能是它没有安装 LILO 或者安装在一个非引导分区中了。
	如果把 LILO 程序安装在 Linux 操作系统的 root 根分区 (/),请确认该分区被设置为
	可引导的
L 错误代码	LILO 程序的第一阶段加载正常,但是第二阶段无法加载。一般情况下,如果引导过程
	在这个阶段失败了,通常是因为磁盘(硬盘或者软盘)损坏,或者是因为硬盘参数不
	是 LILO 程序和初始化的 Linux 内核所预期的
LI	LILO 程序的第一阶段加载正常,第二阶段加载了却无法运行。有时候这个问题的原因
	是不正确的磁盘参数所致。另外,也有可能是移动了/boot/boot.b 文件(LILO 程序的真
	正引导加载部分) , 却没有再执行 LILO 命令更新系统内核的文件映射图
LIL	LILO 程序的第二阶段正常加载并开始运行,但却无法通过/boot/map 内核文件映射图
	加载其必需的文件。这个问题的原因可能是损坏的软盘或硬盘,或者非预期的磁盘参
	数所致
LIL?	LILO 程序的第二阶段加载到了一个错误的地址。通常这个问题的原因是磁盘参数的一
	个小错误,或者是移动/boot/boot.b 文件后没有重新执行 LILO 命令
LIL-	硬盘上的描述符表损坏。这个问题可能是因为磁盘参数不正确引起,也可能是移动
	/boot/map 文件后没有重新执行 LILO 命令
LILO	LILO 正确加载并运行

如表 3.5 所示是 LILO 程序的硬盘错误代码。

表 3.5 LILO 程序的硬盘错误代码

代码	代码含义	可能的解决方案
0x00	内部错误	运行 LILO 程序,重新建立映射文件。这个错误的原因可能是数据
		损坏所致
0x01	非法命令	请检查安装 LILO 程序的硬盘能否被 BIOS 正确识别。也可能是安装
		了太多的物理硬盘
0x02	地址标记没有找到	这个问题通常是因为驱动器故障。重新启动计算机。如果几次之后
		还是出现同样的错误,就可能是软盘或者硬盘上有了物理性损坏。
		使用应急引导启动盘开机,重新安装 LILO 程序

(续表)

代码	代码含义	可能的解决方案
0x03	磁盘写保护	检查 LILO 软盘 , 确认没有动过软盘的写保护块 , 没有找到 0x04 扇
		区,作 LILO 程序遇到非预期磁盘参数。如果是使用 rawrite 程序生
		成的引导盘,请检查该软盘的盘密度是否符合要求。如果硬盘出现
		这个问题,请检查系统内核取得的磁盘参数是否正确
0x06	无法选择操作系统	重新启动计算机。如果几次之后问题仍没有解决 , 请使用应急引导
		启动盘开机,重新安装 LILO 程序
0x08	DMA 过载	重新启动计算机。如果几次之后问题仍没有解决 , 请使用应急引导
		启动盘开机,重新安装 LILO 程序
0x09	DMA 试图跨越 64K 区	从 LILO 程序的配置文件中删除 compact 选项,使用新的设置值重
	域传送数据	新开机
0x0c	无效介质	这个错误通常是因为驱动器故障。重新启动计算机。如果几次之后
		问题仍没有解决,就可能是软盘或者硬盘发生了物理性损坏。请使
		用应急引导启动盘开机,重新安装 LILO 程序
0x10	CRC 错误	这个错误通常是因为驱动器故障。首先多启动几次计算机。如果问
		题没有解决 , 并且是从软盘上运行 LILO , 请使用应急引导启动盘开
		机 ,重新把 LILO 程序安装到一张新的软盘上去。 如果 LILO 程序是
		安装在硬盘上的,请重新安装它。如果还是不能解决问题,考虑把
		LILO 重新安装到软盘上去。可能在试图安装 LILO 程序的位置硬盘
		发生了物理损坏
0x20	控制器故障	这个错误的原因可能是硬盘的控制器发生了故障。多启动几次计算
		机。如果不解决问题,请考虑更换一个新的硬盘控制器。如果是硬
		盘故障,可以把 LILO 程序安装到软盘上;如果是软盘故障,可以
		把 LILO 程序安装到硬盘上
0x40	寻道失败	这个问题通常是因为驱动器故障。重新启动计算机。如果几次之后
		还是出现同样的错误,就可能是软盘或者硬盘有了物理性损坏。使
		用应急引导启动盘开机,重新安装 LILO 程序
0x80	硬盘操作超时	由于各种原因软盘或者硬盘没有准备好

第4章 配置X系统

本章主要介绍如何配置和使用 X 系统。包括 XFree86 系统配置简介、XF86Config 配置 文件的详细介绍、部分 XF86Config 配置文件的配置步骤、使用配置工具 Xconfigurator 和 xf86config 配置 X 系统等内容。

4.1 XFree86系统配置简介

安装 XFree86 系统时最困难、花费时间最多并且最容易受挫折的步骤之一是设置 XF86Config 文件。该文件最初在安装时生成。如果在 Red Hat Linux 7.0 安装时是选择安装 X11,安装成功时则出现一个 GUI,尽管可能分辨率较低。以后升级计算机显存或者安装 新的图形卡时都可以重新配置 XF86Config。

如果对 X11 很陌生,请首先阅读 X 和 XFree86 的手册页获得 X 的大致了解。在开始配置 X 之前,需要了解有关自己的计算机及其显卡和监视器的一些硬件细节。以下是一些有用的信息:

- · 计算机上安装的显卡的类型、制造商、名称或型号。
- · 显卡安装了多少显存。
- · 显卡芯片组使用的时钟芯片类型。
- · 使用的鼠标类型。
- · 该计算机连接的监视器的类型、制造商、名称或型号。
- · 监视器垂直和水平刷新频率。
- · 使用的键盘类型。

了解这些信息之后,接下来需要选择配置 XFree86 系统的方法和工具,并生成该系统 正确的 XF86config 文件。可以使用 Red Hat 的图形化的 X11 设置工具 Xconfigurator、XFree86 的文本模式工具 xf86config,也可以手动构造自己的 XF86Config 文件。Xconfigurator 和 xf86config 程序从控制台的命令行或者终端的命令行运行。

Xconfigurator 的优点是提供图形化的界面;而 xf86config 则是在文本模式的窗口中询问一系列的问题。如果比较幸运,所安装的计算机的硬件与这些程序产生的配置完全匹配。 但是如果这些设置无效、输入的信息不正确或者 XFree86 服务器不是完全支持该显卡芯片,则会产生问题。

4.2 XF86Config文件

配置 X 系统的核心部分是 XF86Config 文件。通过正确地编写该文件,可以配置 X 系统的各个部分。本节将对 XF86Config 文件进行详细介绍。

配置 X Windows 界面会涉及到配置文件 XF86Config。毫无疑问, XFree86 系统最重要 的配置文件是 XF86Config 文件。该文件用于正确地配置字体、键盘、鼠标、显卡芯片、显 示器功能以及颜色深度等 X11 显示服务器设置信息。当启动一个 X 会话时, X 服务器要搜 索这个文件。XF86Config 文件把字体、键盘、鼠标、显卡芯片、显示器技术能力指标和颜 色深度等设置信息适当地提供给 X11 显示服务器程序。

尽管 XF86Config 通常放置在/etc/X11 目录中,在许多系统中也可能在/usr/X11R6/lib/X11 目录中。如果作为 root 用户装载 X,则在/root 目录中可以找到。

应该清楚当前系统硬件的详细情况。对于显示器,必须知道它的水平和垂直同步扫描 频率范围以及带宽。对于图形卡,应该知道它的芯片型号,有时甚至还应知道时钟频率。 对于鼠标,应该知道它是 Microsoft 兼容系列还是其他品牌,如 Logitech 等。同时还应知道 鼠标所连接的端口号。

XF86Config 文件是一个文本文件,它包含好几个部分。可以直接创建这个文件,但最 好使用实用配置程序。用这些实用程序配置时,只需回答有关硬件的一些问题,然后程序 会生成相应的 XF86Config 文件。

可以使用 less 命令查看系统的 XF86Config 文件。在某个终端命令行上, 键入 less, 再 键入 XF86Config 文件的路径, 即:

#less /etc/XF86Config

然后,就可以使用光标方向键浏览这个文件。当阅读完毕时,按Q键退出。

文件/etc/XF86Config 分为如表 4.1 所示的几个主要组成部分。可以在 XF86Config 的 man 帮助文档中找到关于所有这些部分和对他们各项的详细讨论。例如,显示器配置屏生 成/etc/XF86Config 文件中的显示器部分,鼠标配置屏生成鼠标部分,其他类推。每一个部 分以关键词 Section 开头,紧接着是引号括起来的这一部分的名字。每一个部分以 EndSection 结束。注释行都以#开头。每一个部分的各项都以数据说明开头,然后是一系列取值。例如, 在列出 rgb 色彩数据的 File 部分,以数据说明 Rgb Path 开头,然后是 rgb 色彩数据文件的 路径名。

部分	用途
Files	告诉 X 服务器程序色彩、字体 , 或者指定的模块的存放位置
Module	告诉 X 服务器程序必须加载哪一个特殊的模块
ServerFlags	包括各种 on/off 开关标志,用来指出激活或者禁止哪种特殊的操作:如内存转储、键盘
	控制服务器程序终止、显示状态切换、显示调整以及鼠标和键盘的配置等

表 4.1 XF86Config 文件的主要组成部分及其用途

部分	用途
Keyboard	告诉 X 服务器程序检测哪一种键盘以及键盘使用的是哪一种设置
Pointer	告诉 X 服务器程序使用哪一种指针设备以及如何处理这个设备的按钮
Xinput	设备部分的特殊段,用来说明诸如图形板、输入笔等特殊的设备
Monitor	给出显示器的详细资料和设置值 ,如名称、水平同步范围、垂直同步范围以及状态行(即
	视频分辨率,如 640×480、800×600、1024×768
Device	给出视频卡芯片集的详细资料,如 RAM 大小和时钟芯片等
Screen	指定使用哪一种 X 服务器程序、颜色深度 (如 8 位、16 位、24 位或 32 位彩色)、屏幕
	分辨率(如 640×480、800×600、1024×768)以及虚拟屏幕的分辨率大小

不正确的设置可能会损坏显示器。不要使用超出显示器技术指标的显示器设置值。

作为 root 操作员,在试图使用新的 XF86Config 设置开始一个 X11 任务操作之前,请 在最喜欢的文本编辑器程序中打开这个文件,要禁止文本行打包编辑功能,然后认真检查 全部的设置值。检查刚才进行的设置值,激活或者禁止某些 X 服务器程序的选项,输入正 确的显存容量,细调显示器的各种设置,这样的检查是非常必要的。

下面将详细介绍 XF86Config 文件的主要组成部分。

4.2.1 Files部分

Files 部分列出了 XFree86 所需资源的所在目录。这些资源大多数是系统能够提供的字体。一个字体项以数据说明 FontPath 开头,然后紧接着是字体的字体文件路径名。下面是一个 Files 部分的示例。

Files section. This allows default font and rgb paths to be set

Section "Files"

The location of the RGB database. Note, this is the name of the

file minus the extension (like ".txt" or ".db"). There is normally

no need to change the default.

RgbPath "/usr/X11R6/lib/X11/rgb"

Multiple FontPath entries are allowed (they are concatenated together)

By default, Red Hat 6.0 and later now use a font server independent of

the X server to render fonts.

FontPath "/usr/X11R6/lib/X11/fonts/TrueType"

FontPath "unix/:7100"

EndSection

4.2.2 ServerFlags部分

ServerFlags 部分可以用来配置由 XFree86 X 服务器程序激活的某些操作。把某个指定的状态标志前的#字符去掉,就可以激活这个特殊的操作。大多数用户不会禁止 DontZap

(续表)

操作,因为它提供了一个快速退出某个X任务的方法。如果只在一种视频分辨率下使用X, 就可以禁止 DontZoom 操作。 下面是一个 ServerFlags 部分的示例。 # Server flags section. Section "ServerFlags" # Uncomment this to cause a core dump at the spot where a signal is # received. This may leave the console in an unusable state, but may # provide a better stack trace in the core dump to aid in debugging #NoTrapSignals # Uncomment this to disable the <Crtl><Alt><BS> server abort sequence # This allows clients to receive this key event. DontZap # Uncomment this to disable the <Crtl><Alt><KP_+>/<KP_-> mode switching # sequences. This allows clients to receive these key events. DontZoom EndSection Keyboard部分 4.2.3 Keyboard 部分告诉 X 服务器程序检测哪一种键盘及键盘使用的是哪一种设置。下面是 一个 Keyboard 部分的示例。

Keyboard section Section "Keyboard" Protocol "Standard" # when using XQUEUE, comment out the above line, and uncomment the # following line #Protocol "Xqueue" AutoRepeat 500 5 # Let the server do the NumLock processing. This should only be # required when using pre-R6 clients #ServerNumLock # Specify which keyboard LEDs can be user-controlled (eg, with xset(1)) #Xleds 123 #To set the LeftAlt to Meta, RightAlt key to ModeShift, #RightCtl key to Compose, and ScrollLock key to ModeLock: LeftAlt Meta RightAlt Meta ScrollLock Compose RightCtl Control

```
# To disable the XKEYBOARD extension, uncomment XkbDisable.
    #
          XkbDisable
    # To customise the XKB settings to suit your keyboard, modify the
    # lines below (which are the defaults). For example, for a non-U.S.
    # keyboard, you will probably want to use:
    #
          XkbModel
                         "pc102"
    # If you have a US Microsoft Natural keyboard, you can use:
    #
          XkbModel
                         "microsoft"
    # Then to change the language, change the Layout setting.
    # For example, a german layout can be obtained with:
    #
          XkbLayout
                         "de"
    # or:
    #
          XkbLayout
                         "de"
    #
          XkbVariant
                       "nodeadkeys"
    # If you'd like to switch the positions of your capslock and
    # control keys, use:
    #
          XkbOptions
                       "ctrl:swapcaps"
    # These are the default XKB settings for XFree86
    #
          XkbRules
                        "xfree86"
    #
          XkbModel
                         "pc101"
    #
                        "us"
          XkbLayout
                       ....
    #
          XkbVariant
    #
          XkbOptions
                        ....
        XkbKeycodes
                           "xfree86"
        XkbTypes
                           "default"
        XkbCompat
                           "default"
        XkbSymbols
                           "us(pc101)"
        XkbGeometry
                           "pc"
        XkbRules
                           "xfree86"
         XkbModel
                             "pc101"
         XkbLayout
                            "us"
    EndSection
4.2.4 Pointer部分
```

Pointer 部分告诉 X 服务器使用了哪一种鼠标以及这个设备按钮的处理方法。对串行口鼠标,有些 Protocol 的设置值可以设置为 Auto 对总线鼠标可以设置为 BusMouse。在 Device数据项处的/dec/mouse 值就是到实际设备的一个符号连接,比如对一个串行口鼠标就是/dev/ttys0。

Pointer 部分配置鼠标和所有其他定位设备。这一部分只有几个项,其中一些用于定制 某些特定类型的鼠标。Protocol 项指定鼠标所使用的协议,如 Microsoft 或 Logitech。Device 项指定鼠标设备文件的路径名。

以下列出了 Pointer 部分的各项:

- · Protocol 鼠标协议。
- · Device 设备文件路径,如/dev/mouse 或/dev/cua0。
- · BaudRate 串行鼠标的波特率。
- · Emulate3Buttons 允许两键鼠标通过同时按左右键来模拟 3 键鼠标
- ChordMiddle 配置某些 Logitech 3 键鼠标。
- · ClearDTR 和 ClearRTS 清除 TDR 和 RTS 行。只对 Mouse Systems 鼠标有效。
- · SampleRate 设置抽样率。

两键鼠标用户肯定想启用 3 键鼠标模拟。在该模式下,同时按下左右键模拟按下中键。 下面是一个 Pointer 部分的示例。

```
# Pointer section
```

Section "Pointer"

Protocol "PS/2"

Device "/dev/mouse"

When using XQUEUE, comment out the above two lines, and uncomment

the following line.

Protocol "Xqueue"

Baudrate and SampleRate are only for some Logitech mice

- # BaudRate 9600
- # SampleRate 150
- # Emulate3Buttons is an option for 2-button Microsoft mice

Emulate3Timeout is the timeout in milliseconds (default is 50ms)

Emulate3Buttons

Emulate3Timeout 50

ChordMiddle is an option for some 3-button Logitech mice

ChordMiddle

EndSection

4.2.5 Monitor 部分

如果最初的 XF86Config 配置文件无法工作,一定要特别注意 Monitor 部分、Device 部 分和 Screen 部分。Monitor 部分给出了显示器的详细资料和设置值,如显示器的名称、它 的水平同步范围和垂直同步范围以及关键的状态行。了解状态行是仔细调整 X11 显示效果 的关键。

一个状态行由 10 个不同的基本数值部分组成,分别表示下列意义:

- · 一个代表显示器分辨率的标签,如 800×600。
- · 以 MHz 为单位的视频频率。
- · 显示器每行上可见的光点数目。
- · 起始水平回扫值。
- · 结束水平回扫值。

- 显示器上所有可见和不可见的光点数目的总数。
- · 垂直显示结束值,也就是显示器上所有可见光点行的数目。
- · 起始垂直回扫值。
- · 结束垂直回扫值。
- · 总垂直值,也就是显示器上所有可见和不可见的光点行的总数。

系统使用的每一台显示器都应该有一个 Monitor 部分。垂直和水平扫描频率必须精确, 否则可能会损坏显示器。Monitor 部分开始的几项用来标识显示器,如销售商和型号等。 HorizSync 和 VerRefresh 项指定了垂直和水平扫描频率。大多数显示器支持多种分辨率。 Monitor 部分中的各 ModeLine 项指定了分辨率。每一项 ModeLine 对应一种分辨率。每项 有 5 个值,分别是分辨率名称,时钟频率,然后是两组值,一组是水平定时参数,一组是 垂直定时参数,以标志位结束。标志位指定显示模式的各种特征,如 Interlace 表明显示模 式是隔行扫描的,而 + hsync 和 + vsync 则是用来选择信号的极性。

ModeLine 通过指定时钟频率、水平定时参数和垂直定时参数来确定相应的分辨率。如 下是 ModeLine 的一部分。

Modeline "800 x 600" 50.0 800 8569761040600637643666 + hsync + vsync

最好不要改动 Monitor 部分,而完全由配置程序生成各项。 下面列出 Monitor 部分中通常使用的一些项。

- · Identifier 显示器标识名。
- VendorName 生产厂家。
- · ModelName 样式和型号。
- · HorizSync 水平扫描频率,可以是一个范围或一系列值。
- · VerRefresh 垂直刷新频率,可以是一个范围或一系列值。
- · Gamma Gamma 校正。

下面是一个 Monitor 部分的示例。鉴于该部分内容繁多,所以删除了注释语句。

Monitor section

Section "Monitor"

Identifier "SAM1035"

VendorName "Unknown"

ModelName "Unknown"

HorizSync 30 - 61

VertRefresh 50 - 120

 Modeline "512x384" 20.160 512 528 592 640
 384
 385
 388
 404 -HSync -VSync

 Modeline "512x384" 22
 512
 528
 592
 640
 384
 385
 388
 404 -HSync -VSync

 Modeline "640x400" 25.175
 640
 664
 760
 800
 400
 409
 411
 450

 Modeline "640x400" 31.5
 640
 672
 736
 832
 400
 401
 404
 445 -HSync +VSync

 Modeline "640x480" 45.8
 640
 672
 768
 864
 480
 488
 494
 530 -HSync -VSync

40

如果想手动配置 Monitor 部分,请按照下面的步骤进行:

(1) 搜索下面的文本内容:

Monitor section

如果是在 vi 编辑器中进行搜索,请输入/Monitor section,再按回车键。 (2)包含显示器有关设置项目的部分是:

Identifier " MAG DX700T" VendorName " Unknown " ModeName " Unknown"

这3项数据可以使用下面的方法安全地进行修改,不要删掉引号。

· 如果想赋给显示器本身一个新标签,请修改 Identifier 项中的文字。

- · 如果想设置制造商的名称,请修改 VendorName 项中的文字。
- · 如果想为显示器设置一个模式,请修改 ModeName 项中的文字。

(3)进行以下任意修改:

如果想修改显示器的水平同步频率范围,请修改 HorizSync 项的数值。这个数值可以是带有一位小数的某个数值范围,也可以是带有一位小数的一系列数值。在这种情况下,语句中没有引号,如下面所示:

HorizSync 40.5 - 85.3

请准确输入在显示器使用手册中或者制造商 Web 站点上列出的数值范围,或者按 照低限进行估计。如果把这个值设置得高过其处理能力,就可能对显示器造成损 坏。

- · 如果想修改显示器的垂直刷新速率,请修改 VerRefresh 项的数值。这个数值的格 式与 HorizSync 项完全一样,也适用于同样的警告。
- · 如果想设置在某个特定的图形模式下显示器的工作方式,需要设置 Mode 或者 ModeLine 语句。这两个变量定义的是同一个对象,但格式不相同。一个 Mode 语 句如下:

Mode "1024 x 768i" DotClock4 5 HTimings 1024 1048 1208 1264 VTimings 768 776 784 817 Flags "Interlace " EndMode

语句中的数值含义是:

Mode 本身的数值是这个模式的名称。这个名称通常就是按照格式(水平点阵×垂直点

阵)表示的显示器的分辨率。如果显示器是隔行扫描的,末尾再加上一个i字母。

DotClock 项的数值以 MHz 为单位,表示该显示器上单独一个光点的显示频率。

Htiming 项包括 4 个不同的数值。第 1 个是水平显示的点阵数量,第 2 个是水平图像开始处的光点位置,第 3 个是结束处的光点位置,最后一个数值是屏幕水平方向上一行的点 阵总数。这些数值一般都记录在显示器的使用手册中。

Vtiming 项和水平 Htiming 项的排列方式完全一样,只不过它定义的是垂直方向的数据。

Flags 项中包括所有用来通知 X 服务器程序关于显示器特殊问题的必要标志。隔行扫描 是其中最常用的一个。

另一方面, ModeLine 语句把全部的信息都包括在一个文本行中。因此可以把前面 Mode 语句示例中的数据编写到下面这样的 ModeLine 语句中:

ModeLine "1024 x 768i "45 1024 1048 1208 1264 768 776 784 817 Interlace

一般都会有一系列这样的语句,每一句用来定义某个特定模式类型的行为。

(4)保存并退出这个文件。

(5) 输入 startX 命令测试新的配置情况。

4.2.6 Graphics Device 部分

Graphics Device 部分指定系统显示卡。Graphics Device 部分的开头几项用来标识显示 卡,如 VendorName、BoardName 和 Chipset。VideoRam 项指明显存大小; Clocks 项列出了 一组系统的时钟值。这一部分中可以设定很多不同的项,比如,如果显示卡上有 Ramdac 芯片,可以用 Ramdac 项指定 Ramdac 芯片的类型;如果允许访问主缓冲区,MemBase 项 指出主缓冲区的基地址。修改 Clocks 项不太安全。如果设错了时钟值,很容易损坏显示器, 所以还是由配置程序生成时钟值为好。

下面是一个 Graphics Device 部分的示例。

```
# Graphics device section
# Any number of graphics device sections may be present
Section "Device"
   Identifier
               "Generic VGA"
   VendorName
                 "Unknown"
   BoardName "Unknown"
   Chipset
          "generic"
   VideoRam 256
#
          25.2 28.3
#
   Clocks
EndSection
# Device configured by Xconfigurator:
Section "Device"
   Identifier "ATI|3D Rage Pro AGP 1X/2X"
   VendorName "Unknown"
```

BoardName "Unknown" #VideoRam 16384

Insert Clocks lines here if appropriate

EndSection

如果想手动配置图形卡设置值,按照下面的步骤进行:

(1) 搜索下面的文本内容:

Graphics device section

如果是在 vi 编辑器中进行搜索,请输入/Graphics device section,再按回车键。 (2)包含图形卡有关设置项目的部分是:

Section "Device" Identifier "GenericVGA" VendorName "Unknown" BoardName "Unknown" Chipset "generic" VideoRam 1024

EndSection

这 5 项数据可以使用下面的方法安全地进行修改,不要删掉引号。

· 如果想赋给图形卡本身一个新标签,请修改 Identifier 项中的文字。

- · 如果想设置制造商的名称,请修改 VendorName 项中的文字。
- · 如果想为某种特殊的图形卡取一个名字,请修改 BoardName 项中的文字。
- · 如果想设置图形卡用的芯片集的类型,请修改 Chipset 项中的文字。
- · 如果想设置图形卡上拥有的显存 RAM 的大小,请修改 VideoRam 项中的文字。

(3)保存并退出这个文件。

(4) 输入 startX 命令测试新的配置情况。

4.2.7 Screen 部分

Screen 部分中首先是 Driver 项,它指定驱动程序的名字。驱动程序对应于 XFree86 服务器。SVGA 用于 XF86_SVGA 服务器。如果正在使用 XFree86_SVGA 服务器,Driver 项 将是 svga。如果正在使用任意一种加速型服务器,那么 XF86Setup 之类的配置程序将生成 Driver 部分。

在 Driver 项之后是 Device 和 Monitor 项,它们分别指定正在使用的显示器和显示卡。 在这些对应的部分中,Identifier 项给出的名字用于引用部件。具有 Identifier 项 Nec3v 的显 示器在 Screen 部分中将会有 Monitor Nec3v 项。

Screen 部分有 Display 子部分 指定所支持的颜色深度。前面各部分配置硬件,而 Display 子部分则配置一些显示特征,例如显示颜色数和虚拟屏幕尺寸。子部分中主要有 3 项:

Depth、Modes 以及 Virtual。Depth 项指明屏幕分辨率:8、16、24。不管当前的 X 服务器 支持什么显示卡,都可以加入 DefaultColorDepth 项为服务器设定缺省颜色深度:8 表示 256 色,16 表示 32K 色,24 表示 16M 色。Modes 项指定在给定分辨率下允许的显示模式。Virtual 项指定虚拟屏幕的尺寸。能够拥有一个比正常显示面积大的虚拟屏幕。当移动鼠标至显示 屏幕边缘时,它将向隐藏的屏幕移动。这样就能够拥有一个比显示器物理尺寸大得多的工 作屏幕。通常将虚拟屏幕设定为 1024 × 768,相当于 17 英寸显示器的尺寸。也可以设为 1152 × 864,相当于 21 英寸显示器的尺寸。如果想取消虚拟屏幕,可以把 Virtual 项设为 800 × 600,使虚拟显示尺寸和屏幕物理尺寸大小相等。

下面是一个 Screen 部分的示例。

```
# Screen sections
# The Colour SVGA server
Section "Screen"
   Driver
             "svga"
   Device
             "Generic VGA"
   #Device
              "ATI|3D Rage Pro AGP 1X/2X"
   Monitor
             "SAM1035"
   Subsection "Display"
       Depth
                 8
       #Modes
                   "800x600"
                 00
       ViewPort
   EndSubsection
EndSection
# The 16-color VGA server
Section "Screen"
   Driver
             "vga16"
   Device
             "Generic VGA"
   Monitor
             "SAM1035"
   Subsection "Display"
                  "640x480" "800x600"
       Modes
       ViewPort
                 0.0
   EndSubsection
EndSection
# The Mono server
Section "Screen"
   Driver
             "vga2"
   Device
             "Generic VGA"
   Monitor
             "SAM1035"
   Subsection "Display"
                  "640x480" "800x600"
       Modes
       ViewPort
                 0.0
```

EndSubsection EndSection # The accelerated servers (S3, Mach32, Mach8, 8514, P9000, AGX, W32, Mach64 # I128, and S3V) Section "Screen" Driver "accel" Device "ATI|3D Rage Pro AGP 1X/2X" Monitor "SAM1035" DefaultColorDepth 16 Subsection "Display" Depth 8 Modes "800x600" ViewPort 0.0 EndSubsection Subsection "Display" Depth 16 "800x600" Modes ViewPort 00 EndSubsection Subsection "Display" Depth 24 Modes "800x600" ViewPort 0.0 EndSubsection Subsection "Display" Depth 32 Modes "800x600" ViewPort 0.0 EndSubsection EndSection

如果想手动配置图形显示模式,按照下面的步骤进行:

(1) 搜索下面的文本内容:

Screen sections

如果是在 vi 编辑器中进行搜索,请输入/Screen sections,再按回车键。

(2) 一般在 X 图形界面的配置文件中包含不止一个屏幕定义部分,每一部分的基本格式是:

Section"Screen" settings EndSection

请选择某个屏幕定义部分开始操作。

(3)包含屏幕有关设置项目的部分如下所示:

Driver"svga"

Device"MyVideoCard"

Monitor"MAGDX700T"

这3项数据可以使用下面的方法安全地进行修改,不要删掉引号。

- Driver 项中保存着特殊的数据。其选项有 Accel(视频加速)、Mono(单色)、SVGA、 VGA2 和 VGA16。
- · 如果想改变设备本身的名称,请修改 Device 项中的文字。
- · 如果想改变显示器的品牌和型号,请修改 Monitor 项中的文字。

(4)在 Screen 部分中有专门的 Subsection 子结构对配置为使用 Driver 项定义的驱动 程序的特殊图形卡定义颜色和分辨率。这些子结构如下所示:

Subsection "Display" Depth 8 Modes "1024x768" "1152x864" ViewPort 00 EndSubsection

(5)颜色深度是最重要的项目,因为图形卡和显示器在特定的颜色深度下只能支持某 些特定的模式。请确定在图形卡上有多少显存。

(6) 根据选择设置颜色深度。

(7)在 Modes 语句上,在引号中列出在选定的颜色深度下希望使用的各种模式。

(8) ViewPort 项确定在虚拟桌面上 X 任务操作将从什么地方开始显示。一般比较好的方法是就把它保留为默认设置值。

(9) 编辑其他的 Subsection 结构。

- (10) 编辑其他的 Screen 部分。
- (11)保存并退出这个文件。

(12) 输入 startX 命令测试新的配置情况。

4.3 使用配置工具

通过上面的学习,读者可能觉得配置 X 系统是一个艰巨的工作。事实上,有几个工具 可以帮助用户配置 X 系统,下面介绍其中的两个工具。

4.3.1 使用Xconfigurator

Red Hat Linux Xconfigurator 程序将检测系统并询问几个问题,然后创建一个 XF86Config 文件。可以从控制台使用该程序产生一个新的 XF86Config 文件。必须以 root

用户运行该程序。从控制台或者 X11 终端窗口的命令行启动 Xconfigurator,如下所示:

#Xconfigurator

系统清屏并显示一个对话框,如图 4.1 所示。



图 4.1 Xconfigurator 界面

使用 Tab 键可以在屏幕上的各个按钮之间进行切换。当单击 OK 按钮后按 Enter 键, Xconfigurator 首先检查是否有/usr/X11/R6/lib/X11/XF86Config 到/etc/X11/XF86Config 的符 号链接。如果这个链接存在,Xconfigurator 将检测系统的显卡并弹出一个对话框,如图 4.2 所示。



图 4.2 Xconfigurator 报告显卡信息

在单击 OK 按钮之后,出现一个对话框询问与系统相连的监视器类型,如图 4.3 所示。 在 Xconfigurator 的数据库中列有许多种监视器,选中正确的型号,然后单击 OK 按钮。



图 4.3 Xconfigurator 选择监视器

询问安装在计算机或者显卡上的显存容量。通常拥有的显存越多,计算机支持的分辨 率越高,颜色深度越深。如图 4.4 所示。



图 4.4 Xconfigurator 显存配置

选择正确的显存安装量并单击 OK 按钮。

询问显示子系统的时钟芯片类型。如果不确定显卡是否使用时钟芯片设置,选择 No Clockchip Setting。如果知道显卡使用的时钟芯片类型,选择列出的时钟芯片之一并单击 OK 按钮。如图 4.5 所示。

然后,Xconfigurator 询问想要的显示分辨率和颜色深度,如图 4.6 所示。使用 Tab 键和光标键并按下空格键通过浏览选择不同的设置。不要选择超出监视器允许的显示分辨率。 完成设置以后,单击 OK 按钮。

Xconfigurator 试图启动 X,以测试设置。可以跳过测试或者按下 OK 按钮启动 X 服务器。如果跳过测试,Xconfigurator 创建并保存 XF86Config 文件。该文件存储在/etc/X11 目录下。



图 4.5 Xconfigurator 配置时钟芯片



图 4.6 Xconfigurator 配置显示分辨率和颜色深度

开始测试以后,系统清屏,然后出现一个小对话框,询问是否可以看见该消息。如果 单击"Yes"按钮,系统再询问是否在引导时自动启动 X。如果选择 Yes,则修改 Red Hat Linux 系统初始化表/etc/inittab 文件以便引导 Linux 目录到运行级 5。然后需要通过 GNOME 显示 管理器登录。如果单击 No,必须使用 startx 命令启动 X 会话。

在单击 Yes 或者 No 之后, Xconfigurator 在退出之前,创建并保存 XF86Config 文件, 该文件存储在/etc/X11 目录下。

4.3.2 使用xf86config

xf86config 是一个全屏幕文本配置界面,其作用和 Xconfigurator 相似。在 root 权限下的终端执行:

#xf86config

将会出现一个说明界面。如果确信要继续配置,按 Enter 键;如想要退出,按 Ctrl + C 键。如图 4.7 所示。



图 4.7 使用 xf86config

接下来配置鼠标、键盘、国家、显示器水平垂直刷新频率、显存数量和颜色深度。需 要注意的地方和使用 Xconfigurator 时相似,这里不再重复。如图 4.8~图 4.12 分别是上述配 置的使用界面。







图 4.9 使用 xf86config 配置键盘

	Termi	nal					
	File	Edit	Settings	Help			
Er Pr	1 U.S 2 U.S 3 Bel 4 Bul 5 Car 6 Cze 7 Ger 8 Swi 9 Dar 8 Swi 9 Dar 10 Spa 11 Fir 12 Fre 13 Swi 14 Uni 15 Hur 15 Hur 15 Hur 16 Ita 17 Jap	5. Engl Jgian Igian adian ach man siss Ger nish anish	ish ish w/ISOS man nch ngdom to choose r the next	3935-3 9 the country page	1.		4



Terminal
File Edit Settings Help
You must indicate the horizontal syme range of your monitor. You can either select one of the predefined ranges below that correspond to industry- standard monitor types, or give a specific range.
It is VERY IMPORTANT that you do not specify a monitor type with a horizontal sync range that is beyond the capabilities of your monitor. If in doubt, choose a conservative setting.
hsync in kHz; monitor type with characteristic modes 1 31,5: Standard VCA, 640x480 @ 60 Hz 3 31,5 - 35,1; Suger VCA, 100x60 @ 56 Hz 3 31,5, 35,15; 35,5: Suger VCA, 1024x758 @ 87 Hz interlaced (no 800x600) 4 31,5, 35,15; 35,5; Suger VCA, 1024x758 @ 87 Hz interlaced, 800x600 @ 56 Hz 5 31,5 - 42,5; Non-Interlaced Super VCA, 800x600 @ 50 Hz, 640x480 @ 72 Hz 6 31,5 - 42,5; Non-Interlaced Super VCA, 1024x758 @ 60 Hz, 800x600 @ 72 Hz 7 31,5 - 57,0; High Frequency SVGA, 1024x758 @ 70 Hz 3 31,5 - 64,3; Monitor that can do 1280x1024 @ 74 Hz 9 31,5 - 73,0; Monitor that can do 1280x1024 @ 74 Hz 10 31,5 - 62,0; Monitor that can do 1280x1024 @ 76 Hz 11 Enter gour own horizontal spure range
Enter your choice (1-11):

图 4.11 使用 xf86config 配置显示器刷新频率

 Terminal	
File Edit Settings Help	
New your must give information shout your video card. T	his will be used for
the "Device" section of your video card in XF86Config.	HIS WITT DE USEG FOI
You must indicate how much video memory you have, It i idea to use the same approximate amount as that detect intend to use. If you encounter problems that are due not supporting the amount memory you have (e.g. ATI Ms 1024K with the SVGA server), specify the maximum amoun server.	s probably a good ed by the server you to the used server ch64 is limited to it supported by the
How much video memory do you have on your video card:	
1 256K 2 512K 3 1024K	
5 4095K 6 Other	_



在进行完上述一系列配置后,可以选择保存配置到/etc/X11/XF86Config 文件或者保存 到临时文件。



图 4.13 使用 xf86config 配置颜色深度

- 4.4 习题
- 1. 查找 XF86Config 文件,并查看其内容。
- 2. 修改配置文件,进行显示分辨率的配置,查看配置效果。
- 3. 在熟悉系统配置的情况下, 手工修改配置文件。

第5章 轻松使用 GNOME 环境

本章主要介绍 Red Hat Linux 7.0 下的 GUI 界面 GNOME 环境。内容包括 GNOME 简介、 GNOME 桌面环境概述、使用 GNOME 客户程序和工具以及 GNOME 程序使用示例等。

5.1 GNOME简介

GNOME 是一组支持 X11 桌面环境的软件库和 X11 客户程序。GNOME 可以和任何它可以识别的窗口管理器,或者支持其面板组件和客户程序功能(如拖放动作)的窗口管理器一起工作。GNOME 在窗口管理器启动前初始化和运行。和 KDE 一样,GNOME 可以提供良好的用户环境,包括应用程序框架、文件管理器、面板、一组外观一致的应用程序以及会话管理,因此可在 X11 会话之间保存和恢复工作桌面。

GNOME 基于 GNU GPL 发布。它不像底层的图形软件库,如 KDE 的 Qt。除了许可限 制协议这一原因外,GNOME 能够成为 Linux 图形化的 X 桌面特性的重要部分,是基于以 下几点原因:

- · 该软件完全开放源代码,任何人都可以销售。基于该软件的商业软件无需购买软件; 件许可。
- 无需通过集中资源控制,允许赠送、改变和修改,对于更改和发布更改没有许可 限制。
- · 该软件支持多操作系统和外部编程语言。
- · 该软件可以和任何 GNOME 可以识别的 X11 窗口管理器一起工作。

由于许多 GNOME 客户程序包含在 Linux 文件系统的通常位置(如/bin、/usr/bin 或者/usr/X11R6/bin 目录),所以也可以从 X11 终端窗口的命令行启动它们。很容易辨认出GNOME 客户程序,因为大多数客户程序依照 GNOME 风格指南的约定。该指南规定每个程序有支持文档,每个程序有 File 和 Help 菜单。在 File 菜单有 Exit 菜单项,在 Help 菜单 有 About 菜单项。

许多 GNOME 客户程序共同的功能包括活动的菜单和工具条。要将菜单移走,单击并拖动菜单条左端斑点状的矩形即可。使用该方法可重新安排 GNOME 客户程序控件的顺序 (如将终端的菜单条放到终端窗口的底部),或者将控件作为浮动的工具窗口放在客户程序 窗口旁边。每个客户程序在启动时都保存有工具栏或者菜单栏上一次的设置。

5.2 GNOME桌面环境

最常见的 GNOME 客户程序是桌面。在桌面的左侧提供了几个图标,其中包括标题为 home directory、Trash、www.redhat.com、RHN Registration、Dialup Configuration 和 floppy 0 的图标。如图 5.1 所示。



图 5.1 GNOME 桌面

双击 home directory 图标,会打开一个文件管理器,并且当前的目录是用户的主目录。 双击 Trash 也会打开一个文件管理器,其中是 Trash 文件。

双击 www.redhat.com 或 RHN Registration 图标,将会连接到 Red Hat 公司,前者打开 Red Hat 公司的主页,后者注册 Red Hat Linux。

双击 Dialup Configuration 图标,将引导配置拨号网络。

双击 floppy 0 图标,如果软盘驱动器中有磁盘,将打开一个显示软盘内容的文件管理器。

GNOME 桌面的底部是一个任务栏。单击任务栏最左方的 GNOME 按钮,可以装载其他各种 GNOME 客户程序,后面将详细介绍。任务栏中还有几个快捷按钮,它们是集成帮助系统、终端仿真程序、GNOME 配置工具和 Netscape 按钮。

单击集成帮助系统按钮,将会出现一个图形化的帮助界面。可以从中获得各种帮助信 息。如图 5.2 所示。

单击终端仿真程序按钮,将会出现一个终端仿真窗口。可以像使用一个终端一样在窗口中执行命令。如图 5.3 所示。



图 5.2 集成帮助系统

root@Apollo: /root	
File Edit Settings Help	- 1
[root@Apollo /root]#	-

图 5.3 终端仿真程序

单击 GNOME 配置工具按钮,将会出现一个标题为 Control Center 的配置工具。可以在 其中配置桌面、多媒体系统和外设等内容。如图 5.4 所示。

单击 Netscape 按钮,将会出现一个 Netscape 的 Web 浏览器,可以用来浏览页面,收发邮件等。



图 5.4 GNOME 配置工具

5.3 使用GNOME客户程序和工具

单击任务栏最左边的 GNOME 按钮,弹出 GNOME 菜单。该菜单内共有 8 个菜单项。 如图 5.5 所示。

 Programs 菜单项 其级联菜单主要包括 Applications(应用程序)、Utilities(工具)、Games(游戏)、Graphics(图形工具)、Internet(Internet 应用程序)、Multimedia (多媒体应用)、Settings(系统设置)、System(系统工具)、File Manager(文件管理器)和 Help system(帮助)等菜单项。如图 5.6 所示。





图 5.6 Programs 的级联菜单

Applets 菜单项 其级联菜单主要包括 Amusements (娱乐程序)、Clocks (时钟程

序)、Monitors(监控程序)、Multimedia(多媒体应用)、Network(网络应用程序)和Utility(小应用工具)等菜单项。如图 5.7 所示。

🕥 Applets	
截 Amusements	5
🎯 Clocks	
Monitors	\succ
🥁 Multimedia	5
🎲 Network	5
🔓 Utility	1

图 5.7 Applets 的级联菜单

- KDE menus 菜单项 菜单的各菜单项主要是 KDE 环境下的程序和工具。如图 5.8 所示。
- · Run 菜单项 各菜单项提供一个运行终端命令的命令行。
- · Panel 菜单项 其级联菜单的各菜单项主要是管理桌面的工具。如图 5.9 所示。
- · Log out 菜单项 提供退出登录、停机和重启计算机的界面。

下面将举例说明 GNOME 环境中的客户程序和工具的使用。



图 5.8 KDE menus 的级联菜



图 5.9 Panel 的级联菜单

5.3.1 使用Calendar程序定制提醒本

使用 Calendar 可以定制一个提醒本,在特定的时间提示做某件事。操作步骤如下:

(1)在 Programs 的级联菜单中选择 Applications Calendar,出现 Calendar 应用程序界面,如图 5.10 所示。

图中有 4 个标签,分别为 Day View、Week View、Month View 和 Year View 标签,分 别代表现在所在的天、星期、月份和年份的日历。在 Day View 选项卡中,右上角是一个包 含当天的月历,蓝色显示的日期是当天的日期。左边是一个当天每隔半小时一项的提醒条 目,第一项是 00:00,最后一项是 23:30。

- (2) 右击其中的某一项, 弹出一个 New appointment 菜单项。
- (3) 单击这个菜单项, 弹出一个对话框。

root's calendar
<u>File E</u> dit <u>S</u> ettings <u>H</u> elp
Image: New Image: Grade of the section of the sect
Day View Week View Month View Year View
Tue Apr 24 2001
00:00 April > 4 2001 >
00:30 Sun Mon Tue Wed Thu Fri Sat
01:00 1 2 3 4 5 6 7
U3:30 To-do list
04:00 Summary
05:00
05:30
08:00
T_Add Edit Delete

图 5.10 Calendar 应用程序界面

(4)单击对话框中的 Geneval 标签,弹出 General 选项卡。在该选项卡中可以指定该 提醒从何时开始、何时结束和以何种方式提出提醒(可以选择显示、声音、执行程序或者 发送邮件)等内容,还可以指定提醒的私密级别,包括公开、个人和秘密 3 个不同级别, 如图 5.11 所示。

Create new appointment
General Recurrence
Owner: root
Summary:
Time
Start time: 4/24/2001 Calendar 7 09:00 💷 🔄 All day event
End time: 4/24/2001 Calendar / 09:30
Alarms
Display 15 🔆 Minutes 💷
□ Audio 15 7 Minutes □
Program 15 🔆 Minutes 🖃 Run program: 🛛 🛛 Browse
Mail 2 Minutes Mail to:
Classification
* Public -> Private -> Confidential
OK X Cancel

图 5.11 用 Calendar 创建一个提醒

(5)在 Recurrence 选项卡中,指定该提醒的重复性。
5.3.2 使用GNOME Control Center定制桌面

可以通过 GNOME Control Center 配置个性化的桌面。可以从工具条上的 The GNOME configuration tool 启动 GNOME Control Center,或者从 Settings 菜单的 GNOME Control Center 菜单项启动。GNOME Control Center 见图 5.4 所示。

在 GNOME Control Center 的使用界面中, 左侧是一个目录, 第一个目录项是 Desktop, 它包括 5 个项目。

· Background 项 用于定制桌面背景及其属性。单击 Background,出现的界面如图 5.12 所示。

	Control Center	
ana a	<u>File</u> <u>H</u> elp	
	Desktop	Vallpaper
	Screensaver Screensaver Window Manager Document Handlers Ocument Handlers Minime Types URL Handlers Multimedia Scient	↓ Tiled ↓ Centered ↓ Scaled (keep aspect) ↓ Scaled ★ Embossed Logo Color
	Peripherals OD Properties Strength Reyboard Mouse	Vertical Gradient
	Gawnish window manager Gawnish window manager Gawnish window manager Gawnish window behavior Gawnished Windows Matched Windows	T Use GNOME to set background
	infiguration of the desktop's backgrour	10

图 5.12 定制桌面背景及其属性

· Panel 项 用于定制面板及其属性。单击 Panel,出现的界面如图 5.13 所示。

Control Center	
Eile Help	
B Desktop	Animation Buttons Panel Objects Menu Miscellaneous
- Viceensaver - Window Manager	_ Constant speed animations Animation speed
Document Handlers	Auto hide
URL Handlers	Explicit hide
Gound G	Auto hide
Mouse Sawfish window manager	Delay (ms)
- W Focus behavior	
Global panel properties	Prevent OK Cancel

图 5.13 定制面板及其属性

Screensaver 项 用于定制屏幕保护及其属性。单击 Screensaver,出现的界面如图 5.14 所示。

Control Center	
<u>F</u> ile <u>H</u> elp	
Elle Help Elle Help Background Background Panel Control Center Background Panel Control Center Background Panel Control Center Background Descute Control Center Background Control Center Ce	Screen Saver Screen Saver No Screensaver 3D Clock Ant Attraction Bitspin Bouboule Braid Braid Random Settings Screen Saver Settings Start After 20 Minutes. I Require Password Priority: Low I Use power management. Shutdown monitor 20 minutes after screen saver has started.
Configure the settings of the screensaver	🔊 Try 💫 Revert 🥥 OK 🄀 Cancel
Configure and country of the serverbarrer.	

图 5.14 定制屏幕保护及其属性

· Theme Selector 项 用于定制桌面窗口及其属性。单击 Theme Selector,出现的界面如图 5.15 所示。

Control Center	
<u>F</u> ile <u>H</u> elp	
Desktop Background Panel Screensaver	Available Themes
- Content Selector - Decument Handlers - Default Editor Mine Tynes	Cheese Default Install new Ins
Willing Types Willing Types Willing Types Multimedia Willing Sound	Use custom font. Helvetica
Perpherais OP Properties Son Keyboard Mouse	Selected themes from above will be tested by previewing here. Sample Button Submenu One Two Sample Check Button Sample Check Button
Sawish window manager - Appearance - Focus behavior - Matched Windows - Meta	Sample Text Entry Field Radio Button 2 More More 7
Select which desktop theme to use.	

图 5.15 定制桌面窗口及其属性

•

• Window Manager 项 用于选择窗口管理器。

下面以 Background 为例介绍如何定制桌面。

Background 的使用界面分为左右两部分。在左面设置选项,在右面显示设置效果,这 样便于用户作决定。

Wallpaper 选项区用于设置桌面的墙纸。可以通过下拉选项或者文件打开对话框进行墙 纸图像文件的选择。选定墙纸之后,通过下面的单选按钮决定墙纸显示的属性,如居中、 拉伸等。

Color 选项区用于设置墙纸的颜色。通过下拉选项,可以选择墙纸颜色的渐变方向,包括不渐变、水平方向渐变或垂直方向渐变。下面的按钮可以选择墙纸的颜色基调。

5.3.3 使用Panel菜单定制面板

Panel 菜单用于定制面板及其属性。单击任务栏上的 GNOME 按钮,或者右击任务栏,从弹出的菜单中选择 Panel,弹出 Panel 级联菜单。它包括三部分,第一部分用于添加面板或者在面板中添加选项,它包括 Add to panel 和 Create panel 菜单项。第二部分用于定制面板的属性,包括 Properties 和 Global Preferences 菜单项。第三部分是使用帮助,包括 Panel Manual、About the panel 和 About GNOME 菜单项。见图 5.9 所示。

下面以在任务栏上添加一个按钮为例来说明 Panel 的用法。操作步骤如下:

(1) 选择 Add to panel Lancher, 弹出一个如图 5.16 所示的界面。

Create laur	Create launcher applet							
Basic Adva	anced	-						
Name:								
Comment:								
Command:								
Туре:	Application							
lcon:	No Icon 🔄 Run in Terminal							
	OK 🎗 Cancel 🤋 Help							

图 5.16 在任务栏上添加一个按钮

(2)在 Name 文本框中输入这个按钮的名字;在 Comment 文本框中输入一段说明, 这段说明将在鼠标移动到这个按钮上时显示出来;在 Command 文本框中输入当单击这个 按钮时系统执行的动作,通常是一个命令;保持 Type 的下拉菜单的默认值为 Application。 (3) 单击 No Icon 按钮选择按钮的图标。在此有非常丰富的图标资源供选择。 (4) 单击 OK 按钮,一个新的按钮图标就添加到了任务栏上。

在任务栏上添加一个抓图程序按钮的界面和结果,如图 5.17 和图 5.18 所示。

Create launcher applet							
Basic Adva	anced						
Name:	xv						
Comment:	a graphical snapshot						
Command:	xv &						
Туре:	Application						
lcon:	Run in Terminal						
	🥔 OK 🛛 🎗 Cancel 💡 Help						

图 5.17 在任务栏上添加一个抓图程序按钮

• 😳 🙆 🕉 • N 11:10 AM root@Apollo: /home/duwei/05 0

图 5.18 添加一个抓图程序按钮之后的任务栏

5.4 习题

- 1. 详细查看集成帮助环境提供的帮助,学习 GNOME 的使用。
- 2. 查看 GNOME 菜单,了解每个菜单项的作用。
- 3. 定制自己喜欢的桌面环境,包括背景、屏幕保护和桌面风格等。
- 4.在 GNOME 菜单中添加一个菜单项,执行一个程序。

第6章 Linux 常用命令

本章主要介绍 Linux 的常用命令。内容包括获取 Linux 命令的帮助、Linux 常用文件和 目录命令详解以及一些常用的非文件命令详解。

6.1 Linux常用命令简介

初次接触 Linux 时一定会感觉不适应 Linux 的命令,它们和 DOS 命令是不同的。可以 通过下面的内容了解这些不同,并且可以自己获得 Linux 命令的使用帮助。

Linux 的开发人员们做了大量的工作,使现在的 Linux 操作系统比它最初的版本更加倾 向图形界面。但是,还是需要在命令行的界面中进行许多的工作。事实上,是 Linux 操作 系统的强大功能照亮了命令行操作方式,使用户几乎可以对其系统中发生的任何事件都能 进行一定的控制。这一点对专注于提供网络或者因特网服务功能的服务器计算机,尤其正 确。因为 GUI 只会降低这些计算机的速度,所以学习掌握在这类计算机上使用命令行工具 的方法是非常必要的。但是不只是可以选择这两者之一进行操作,还可以在两者之间随时 切换。

许多人认为 Linux 操作系统和 DOS 差不多,因为这两种操作系统都提供了基于命令行的操作方式,但实际上二者有很大的差别。Linux 操作系统是一个多任务网络操作系统,要求用户按照各自的身份登录,而不是把他们一视同仁。分配给用户的安全级别是在建立他的帐户时就设置好的。而 DOS 是一个单任务系统,也就是说它一次只能完成一件事情,而且,无法管理多个用户的帐户,它把任何坐在计算机前面的人都当作是同一个人。

从较为初级的角度看,改用 Linux 操作系统后在涉及到目录的时候,Linux 使用正斜杠 (/)而不是 DOS 或者 Windows 中的反斜杠 (\)。例如,Linux 操作系统中的一个合法目录 的写法是/usr/bin。另外,Linux 操作系统在理解输入的字母时非常挑剔,也就是说输入一 个大写字母和输入一个小写字母的含义是完全不同的。因此输入字母 Vi 与输入字母 vi 是 不一样的。这与 Windows 形成了强烈对比,因为 Windows 对大小写不敏感。

在 Linux 操作系统下,可以非常容易获得帮助。像大多数 UNIX 运行版本一样, Linux 操作系统发行版本也为几乎每个程序、工具、命令或系统编程调用编制了使用手册页。从 中可以得到几乎所有命令的有关信息,包括 man 命令本身。例如,输入下列命令就可以阅 读 man 命令的使用手册页:

#man man

使用手册页就像一页 Linux 的命令汇总。每页使用手册页文件的名称以一个个位数字 作为文件后缀,存放在/usr/share/man 的一个子目录下。许多 Linux 操作系统命令的使用手 册页都在最初的安装过程中,或者在单独安装某个命令程序时拷贝到硬盘。如,man 命令 的使用手册页文件被命名为:

/usr/share/man/man1/man.1.gz

man 命令的使用手册页的名称是 man.1, 后缀.gz 表示 man.1 文件已经使用 gzip 命令压 缩过了。man 命令在显示使用手册页内容之前会自动解压缩这个文件。

在子目录/usr/share/man/man1 中存放有超过 1000 个使用手册页的文件。但是如果想知 道使用手册页的格式和存放的位置,在man.1 中是找不到的,它们存放在/usr/share/man/man7 的 man.7 文件中。

如表 6.1 所示是每个使用手册页组成部分的位置和内容。

目录	内容
/usr/share /man/man1	命令——在 shell 中执行的命令
/usr/share /man/man2	系统调用——关于核心函数的文档
/usr/share /man/man3	库调用——libc 函数的使用手册页
/usr/share /man/man4	特殊文件——关于/dev 目录中的文件的信息
/usr/share /man/man5	文件格式——/etc/passwd 和其他文件的详细格式
/usr/share /man/man6	游戏
/usr/share /man/man7	宏命令包——对 Linux 文件系统和使用手册页等的说明
/usr/share /man/man8	系统管理——根操作员操作的使用手册页
/usr/share /man/man9	核心例程——关于 Linux 操作系统内核源例程或者内核模块技术指标的文
	档

表 6.1 Linux 操作系统的使用手册页的组成部分

每页手册页都分成几个组成段。如表 6.2 所示是手册页的组织格式。

表 6.2 手册页的组织格式

组成段	说明
Name	命令的名称及简单说明
Synopsis	如何使用这个命令及命令行参数
Description	对这个程序命令及其参数的解释
Files	这个命令用到的文件清单和它们存放的位置
See Also	有相互联系的使用手册页的清单
Diagnostics	特殊输出情况的说明
Bugs	编程漏洞
Author	程序的主要编写者和维护人员

在子目录/usr/doc 下找到更多的关于 Linux 操作系统的命令和其他主题的详细资料。这个子目录中有常见问题答疑文件 FAQs 和操作指导文件 HOWTO。

man 命令对使用手册页进行检索的时候,主要依据在子目录/etc 中 man.conf 文件中的 详细说明的原则。这些原则规定了查找使用手册页的缺省子目录。可以查找这些使用手册

页的缺省位置有:

MANPATH /share/usr/man MANPATH /usr/man MANPATH /usr/local/man MANPATH /usr/X11R6/man MANPATH /usr/lib/perl5/man MANPATH /usr/kerberos/man

在 X 窗口系统中有一个 man 命令的图形化界面的版本,叫做 xman。不仅可以使用 xman 阅读使用手册页,还可以看到存放这些使用手册页的子目录的路径名。如果使用的是 X11 下的 KDE 桌面环境,将会发现 KDE 的使用手册页是超文本标记语言格式,即 HTML 格式。

6.2 常用文件、目录命令

对于普通用户,最常使用的 Linux 命令是针对文件和目录的命令。本节将对这些常用 的文件和目录命令进行介绍。

6.2.1 cd命令

切换子目录命令 cd 是在 Linux 操作系统的不同部分之间移动的基本工具。登录进入 Linux 操作系统之后,处在用户的主目录中。这个目录有一个名字,也就是"路径名",它 由/home/开头,后面跟着用户名。

输入 cd 命令,并在后面跟一个路径名,就可以直接进入到另外一个子目录中。如下面的命令将切换到/usr/bin 子目录中:

#cd /usr/bin

在/usr/bin 子目录中时,可以用下面的命令进入/usr 子目录:

#cd ..

在/usr/bin 子目录中时还可以使用下面的命令直接进入根目录,即"/"目录:

#cd /

最后,用下面的命令回到自己的用户主目录:

#cd

或者

#cd ~

如果使用 man 命令查看 cd 命令的使用手册页,将找不到任何有关信息。因为 cd 命令 是内建在 shell 中的,而 shell 在敲过回车键之后就直接解释执行命令行。 66

显示工作子目录 pwd 命令告诉用户所在的位置,并显示当前工作子目录。例如,如果 执行命令:

#cd /usr/bin

接着输入:

#pwd

会看到:

/usr/bin

pwd 命令虽然有一个使用手册页,但是实际上每次使用 pwd 命令的时候使用的都是一 个内建在 shell 中的 pwd 命令。怎样来区别它们呢?如果用下面的方式调用 pwd 命令,就 会看到它只给出当前的工作子目录:

#pwd --help

调用子目录/bin 中的 pwd 命令:

#/bin/pwd --help

它将执行子目录/bin 中的 pwd 命令而不是执行内建在 shell 中的 pwd 命令。将看到一个简短的 pwd 命令的帮助文件而不是当前工作子目录。

如果确实搞不清自己到底在什么位置,可以使用子目录/bin 中的 pwd 命令。

6.2.3 ls命令

列出子目录内容清单 ls 命令是最经常使用的程序命令之一。使用它最基本的格式,可 以列出当前子目录下几乎所有的文件。虽然这个命令本身只有两个字母,但是它的命令行 参数却比其他任何程序都多。

在/usr 下运行 ls 的结果如下:

X11R6 cgi-bin etc i386-glibe21-linux kerberos libexec sbin src bin dict games include lib local share tmp

它列出了/usr 子目录中的文件。 也可以使用-m 参数把文件用逗号分隔显示在一行上:

#ls -m

X11R6,cgi-bin,etc,i386-glibe21-linux,kerberos,libexec,sbin,src,bin,dict,games,include,

lib,local,share,tmp

如果不喜欢 ls 命令输出的这种排列方式,也可以用-x 参数按水平对齐的方式而不是缺 省的垂直对齐方式列出文件:

#ls -x

	X11R6 bin	cgi-bin dict	etc games	i386-glibe21-linux include	kerberos lib	libexec local	sbin share tmp	src
	所列出来	医可能是文	(件也可能	湜子目录。区分它	们的一个	办法就是	使用-F 参	数:
	#ls -F							
	X11R6	/cgi-bin/	etc/	i386-glibe21-linux/	kerberos/	libex	ec/ sbin/	src/
	bin/	dict/	games/	include/	lib/	local	/share/	tmp
	可以看出	Ⅎ,-F参数	(使得 ls 命	令在显示子目录的	时候可以	在它的文	件名之后	加上一个斜
线	(" / ") 字符	夺。						
	如果希望	看到所有	前的文件,	可以将-a参数和-F	参数一起	使用,如	下所示:	
	#ls –F -a							
	.X11R6	/cgi-bin/	etc/	i386-glibe21-linux/	kerberos/	libex	ec/ sbin/	
	src/	bin/	dict/	games/	inclu	de/	lib/	local/
	share/	tmp						

除非明确指出显示所有的文件,那些以句号(".")开头的文件或者子目录是无法通过 ls 命令显示出来的。

可以让 ls 命令以不同的颜色来显示文件、子目录或者可执行文件。可以像下面这样使用--color 参数:

#ls --color

可以使用 ls 命令的-l 参数来列出文件的全部信息:

#ls -l

total 108

drwxr-xr-x	8	root	root	4096	Apr 19 11:59	X11R6
drwxr-xr-x	6	root	root	32768	Apr 19 12:11	bin
drwxr-xr-x	2	root	root	4096	Apr 19 12:00	cgi-bin
drwxr-xr-x	2	root	root	4096	Feb 6 1996	dict
drwxr-xr-x	2	root	root	4096	Feb 6 1996	etc
drwxr-xr-x	2	root	root	4096	Apr 19 12:09	games
drwxr-xr-x	4	root	root	4096	Aug 1 2000	i386-glibc21-linux
drwxr-xr-x	66	root	root	8192	Apr 19 12:11	include
drwxr-xr-x	7	root	root	4096	Apr 19 12:02	kerberos
drwxr-xr-x	63	root	root	20480	Apr 19 12:11	lib
drwxr-xr-x	7	root	root	4096	Apr 19 12:07	libexec
drwxr-xr-x	11	root	root	4096	Apr 19 11:53	local
drwxr-xr-x	2	root	root	4096	Apr 19 12:11	sbin
drwxr-xr-x	94	root	root	4096	Apr 19 12:11	share
drwxr-xr-x	4	root	root	4096	Apr 19 12:06	src
lrwxrwxrwx	1	root	root	10	Apr 19 11:53	tmp ->/var/tmp

可以看出,总共有7个不同的信息栏。

第 1 栏是文件的权限标志。这些标志通常表明某个文件的类型以及其他用户是否能够 对这个文件进行读、写或者运行等操作。

第2栏给出这个文件的链接个数。

第3栏是文件所有者的用户名。

第4栏是这个用户所在的用户组名。

第5栏给出文件的长度。

第6栏是这个文件或者子目录创建或者最后一次被修改时的日期和时间。

第7栏是这个文件的文件名。

可以在 ls 命令的命令行上指定其他子目录或者路径名,这个方法可以用来查看其他子目录中的内容。例如,如果想查看子目录/usr/bin 中的全部文件,可以使用下面的方法:

#ls /usr/bin

最后,如果想看到计算机系统中的全部文件,可以使用 ls 命令的-R 参数,它将逐层进入所有子目录并把其中的内容全部显示出来。虽然可以使用这个方法检索和归类系统中的 文件,但显示这些文件需要好几分钟,而它的输出结果可能还包括那些不想列出的文件或 者是其他操作系统的文件,如 DOS 或 Windows 上的文件,特别是在使用了下面的命令的 时候:

#ls -R /

一个比较好的办法是把-d 参数和-R 参数一起使用 ,这样可以只列出有限的子目录层次 下的文件。例如 , 下面的命令将沿着根目录 (" / " 目录) 列出三层子目录中的内容:

#ls -Rd /*/*/*

6.2.4 touch命令

touch 命令使用起来很简单,一般会在两种情况下用到它。第1种情况是建立文件,第 2种情况是更新文件的修改日期。touch 命令是 GNU 文件工具包的一个组成部分,另外还 有几个命令参数。

如果想使用 touch 命令建立一个文件,请使用下面的方法:

#touch newfile

#ls -l newfile

-rw-r--r--1 duwei users 0 Jan 5 12 : 40 newfile

可以看到, touch 命令建立了一个长度或者大小为0的文件。也可以使用下面的命令:

#>newfile2

#ls -l new*

-rw-r--r--1 duwei users 0 Jan 5 12 : 40 newfile

-rw-r--r-1 duwei users 0 Jan 5 12 : 41 newfile2

这类似 touch 命令。上面的操作也建立了一个长度为 0 的文件。那么,为什么在通过

命令也可以建立文件的情况下还要使用 touch 命令呢?这是因为 touch 命令可以更新一个文件的建立日期或者时间,甚至可以使用 touch 命令把一个文件的建立日期或者时间改为过去或者将来时间,如下所示:

#touch newfile2
#ls -l newfile2
-rw-r--r--1 duwei users 0 Jan 5 12 : 44 newfile2

可以使用 touch 命令的-t 参数加上一个代表年、月、日、时的数字把某个文件的建立日 期或者时间设置为一个具体的时间,如下所示:

#touch -t 0125130001 newfile2.# ls -l --full-time new*

-rw-r--r--1 duwei users 0 Tue Jan 05 12 : 40 : 14 2001 newfile

-rw-r--r--1 duwei users 0 Mon Jan 25 13 : 00 : 00 2001 newfile2

touch 命令的一个用处是使用在进行备份操作的过程中。在备份一些文件和子目录之前 或者之后,使用 touch 命令可以更新这些文件的时间记录。这样,备份程序在进行下一次 备份工作的时候就有了一个可供参考的时间。

6.2.5 rm命令

rm 命令用来删除文件。这个命令只有几个简单的参数,但是在使用的时候千万要小心 谨慎。如果 rm 命令删除了某个文件,这个文件就无法恢复了。

以根操作员的身份登录进入系统并使用 rm 命令会造成许多灾难性的后果。因为一个简 单的命令,不仅可能会毁掉 Linux 系统,还可能毁掉包括 DOS 分区、RAM 卡、活动硬盘 等在内的任何已挂装的其他类型的文件系统。这个命令就是:

#rm -fr /*

这个带着-r 参数的命令会从根目录 (/) 开始递归地删除所有的文件和子目录。如果必须以根操作员的身份运行 Linux 操作系统 , 一定要先对系统进行备份。

```
rm 命令可以从命令行上一次删除一个或者几个文件。可以使用下面的任何一种方法:
#rm file
```

```
#rm file1 file2 file3
#rm file*
```

第 1 个命令行删除一个名称为 file 的文件;第 2 个命令行删除 3 个文件;第 3 个命令 行删除当前子目录中所有文件名以 file 开头的文件。使用 rm 命令比较安全的办法之一是使 用它的-i 交互操作参数。这样,在操作过程中会被问到是否真的想删除某个文件,如下所 示:

```
#rm -i new*
rm : remove 'newfile '? y
rm : remove 'newfile 2 '? y
```

还可以使用-f参数强行删除某个文件,如下所示:

#rm -f new*

如果使用了-f 参数但是没有文件能够匹配 new*格式的时候,rm 命令失效,但是不会 显示任何出错信息。而且当 rm 命令遇到子目录的时候,即使子目录是空的,也不进行删除 操作,并且还会显示出错信息,甚至在使用了-f 参数时也是如此。如下所示:

#rm -f temp*
rm : temp : is a directory

rm : temp2 : is a directory

但是当把-f 和-r 参数一起使用的时候,就可以删除这个子目录和这个子目录下面的所 有文件和子目录,条件是只要用户拥有它们或者拥有这个操作的权限。-f 和-r 参数可以像 下面这样联合使用:

#rm -fr temp*

-fr 参数使得 rm 命令的执行就像 rmdir 命令一样。

使用这个参数时千万要小心。如果在 Linux 操作系统中删除了一个文件,它就永远消 失了。

6.2.6 mkdir命令

mkdir 命令一次可以建立一个或者几个子目录。mkdir 命令还可以只使用一个命令行一次就建立起包括全部的父目录和子目录在内的一个完整的子目录继承结构。

这个命令加上 cp 命令和 mv 命令,是用来组织资料信息的基本工具之一。现在先来看 一些示例。如下所示,一条简单的命令建立起了一个子目录:

#mkdir temp

还可以使用下面的方法一次建立好几个子目录:

#mkdir temp2 temp3 temp4

还可以输入下面的内容在子目录 temp 下再建立一个名为 child 的子目录:

#mkdir temp/child

因为子目录 temp 已经存在,所以上面这条命令是有效的。但是如果输入下面的内容:

#mkdir temp5/child

mkdir : cannot make directory 'temp5/child ': No such file or directory

可以看到, mkdir 命令会提示子目录 temp5 不存在。如果想使用 mkdir 命令建立一系列 完整的子目录结构,就必须使用它的-p 参数,即父操作参数。如下所示:

#mkdir -p temp5/parent/child
#tree temp5

可以看到,mkdir 命令不仅建立了子目录 temp5,还在它的下一层建立了 parent 子目录,而子目录 parent 中又有一个 child 子目录。

6.2.7 rmdir命令

rmdir 命令用来删除子目录。如果希望删除某个子目录,只需要输入下面的内容即可: #rmdir temp

注意:这个子目录必须是空的。如果试图删除一个其中还有文件的子目录,就会得到出错信息:

#rmdir temp5

rmdir : temp5 : Directory not empty

在这个例子中,子目录 temp5 中还有其他的子目录。如果某个子目录的名称代表的其 实是一个文件而不是子目录,rmdir 命令也会显示出错信息。解决的方法是先用 rm 命令删 除这些文件,或者先把这些文件转移到其他的地方,或者使用 mv 命令给子目录改个名称。

与 mkdir 命令相同, rmdir 命令也有一个-p 参数。可以使用这个参数来删除某个子目录的全部继承结构, 如下所示:

#rmdir -p temp5/parent/child

使用-p 参数,必须指明某个子目录完整的结构才能够删除它。如果使用了与上面同样的命令但是没有加上-p 参数,那么就只有子目录 child 被删除。如果有两个或者更多个子目录,不加-p 参数,只能删除指定的子目录;使用-p 参数,效果相同,但会出现一个提示性信息,如下所示:

3 directories, 0 files

为了删除子目录 temp5 整个的子目录结构,必须使用下面的命令:

#rmdir temp5/parent/*

6.2.8 cp命令

cp 命令,即拷贝命令,是用来对文件或者子目录进行拷贝操作的。这个命令有几十个 命令行参数。本小节中将只介绍其中几个最常用的参数的使用方法。

第一次使用 cp 命令的时候,用到的可能是它最简单的形式,如下所示:

#cp file1 file2

这个命令把文件 file1 拷贝到文件 file2,同时 file1 还依然存在。但是在使用 cp 命令的 时候必须格外小心,因为在把一个文件拷贝到另外一个文件上的时候,可能会完全覆盖原 来的文件。

cp 命令还可以用来一次拷贝多个文件。类似于 rm 命令, cp 命令也有一个-r 参数。可 以使用这个参数把一个子目录拷贝到另外一个子目录中。例如,如果想把子目录 tempdir1 及其中的文件拷贝到子目录 tempdir2 中,请使用下面的方法:

#cp -r tempdir2 tempdir1

6.2.9 mv命令

mv 命令,也就是文件名更改命令,可以对文件或者子目录的名称进行更改。它也可以 被称为文件移动命令,在文件系统内移动文件或者子目录。事实上,那些文件或者子目录 并没有被真正移动。

mv 命令最普通的用法是更改文件名,如下所示:

#touch file1

#mv file1 file2

这个命令把文件 file1 改名为文件 file2。除了更改文件名之外, mv 命令还可以用来更 改子目录名而不管这个子目录是空的还是存有文件。即使使用 mkdir 命令建立了一个完整 的子目录结构,仍然可以使用 mv 命令更改新的顶层子目录的名字,如下所示:

#mkdir -p temp/temp2/temp3
#mv temp newtemp

6.2.10 cat 命令

合并文件命令 cat 用来把文件内容送到屏幕上显示。也可以使用这个命令通过输出重 定向把文件内容送到其他的文件中。下面只介绍这个命令的基本用法。

cat 命令在阅读短文件的内容的时候很有用,但它更经常地被用来进行文件的合并、建 立、覆盖或者添加内容等操作。如果想使用 cat 命令查看一个短文件,可以输入如下的命 令:

#cat test.txtToday is my birthday,I am very happy,because my girlfriend give me a lovely present..

cat 命令也有一些参数。如果想在阅读文件的时候加上行号 , 如想要注明某个特殊的段 落 , 可以使用-n 参数 , 如下所示 :

#cat -n test.txt

1 Today is my birthday,

2 I am very happy,

3 because my girlfriend give me

4 a lovely present ..

因为 cat 命令也接受通配符,所以,还可以使用 cat 命令一次查看多个文件,如下所示:

#cat -n test*

1 Today is my birthday,

2 I am very happy,

3 because my girlfriend give me

4 a lovely present ..

- 5 This is the first line of test2.txt.
- 6 This file was also created by cat .
- 7 This is the last line of test2.txt .

可以看到, cat 命令在它的输出中包含了另外一个文件,并为整个输出结果的每一行加 上了行号。可以使用下面的方法来查看这两个文件:

#cat test.txt test2.txt

这个命令的输出结果与使用通配符的时候看到的结果完全一样。

同时查看几个文件只是 cat 命令的一种用法。还可以使用 cat 命令和重定向操作符(>) 来合并文件。例如,如果想把文件 test.txt 和文件 test2.txt 合并到 test3.txt 文件中,可以使用 下面的方法:

#cat test* > test3.txt

可以用下面的方法检查操作结果:

ls -l test*

-rw-rr	1	duwei	users 190	Jan	4	16:06test.txt
-rw-rr	1	duwei	users 108	Jan	4	16:07test2.txt
-rw-rr	1	duwei	users 298	Jan	4	16:07test3.txt

如果只是想把 test.txt 和 test2.txt 文件合并,并不想再生成另外一个更大的文件。这时, 首先需要决定是把 test.txt 的内容加到文件 test2.txt 中,还是把 test2.txt 的内容加入 test.txt 中。然后,使用 cat 命令和重定向符>>,输入下面的内容:

#cat test.txt>>test2.txt

这个命令把文件 test.txt 的内容添加到文件 test2.txt 的后面。可以再使用 cat 命令检查操 作结果:

#cat test2.txt
This is the first line of test2.txt .
This file was also created by cat .
This is the last line of test2.txt .
Today is my birthday,
I am very happy,
because my girlfriend give me
a lovely present..

需要注意的是,如果使用了下面的命令:

#cat -n test.txt>>test2.txt

现在, test2.txt 文件的内容会是下面这样:

#cat test2.txt

This is the first line of test2.txt .

This file was also created by cat .

This is the last line of test2.txt .

1 Today is my birthday,

2 I am very happy,

3 because my girlfriend give me

4 a lovely present..

6.2.11 more命令

more 命令是 Linux 操作系统命令中的一个页命令。页命令使用户在浏览文件的时候可 以一次阅读一屏或者一行。这在阅读大量使用手册页的时候特别有用,因为 man 命令是使 用一个页命令来显示每一页的。

more 命令是一个传统意义上的页命令,可以在命令行上像下面这样使用 more 命令,

#more longfile.txt

阅读一个文本文件是相当容易的,因为可以按空格键阅读后一页,也可以按 B 键阅读 前一页。

more 命令也有一些命令行参数。还可以设置 more 命令显示正在阅读的文件的读取百分比,设置前后翻阅文本时显示行数,使用多个文件名或通配符及打开或关闭滚屏功能。

6.2.12 less命令

less 命令有点像 more 命令,但是 less 命令的功能比 more 命令更强大。less 命令和 more 命令一样都是页命令,但是它改进了 more 命令,又添加了许多其他特色。

less 命令提供了许多比 more 命令更优秀的功能:

- · 可以使用光标键在文本文件中前后滚屏。
- · 可以用行号或百分比作为书签来浏览文件。
- · 可以实现在多个文件中进行复杂的检索、格式匹配和高亮度显示等操作。
- · 键盘操作与字处理程序如 emacs 兼容。
- · 阅读到文件结束或者标准输入结束的时候 less 命令不会退出。
- 屏幕底部的信息提示更容易控制使用,而且提供了更多的信息。
- · 带有许多附件,包括一个独立的键定义程序 lesskey,这样就可以定义使用哪些按 键来控制 less 命令。

安装了 Linux 操作系统之后, less 页命令将是许多程序使用的缺省的页命令。

作为演示,首先使用 ls 命令建立一个内容为子目录/usr/bin 中所有命令的文本文件,如 下所示:

#ls /usr/bin > programs.txt

这个命令建立了一个名为 programs.txt 的文本文件。然后,使用 less 命令阅读这个文件, 如下所示: #less programs.txt

less 命令装入这个文件并显示这个文件开头的一些行。如果想向下翻一页,按空格键; 如果想向上翻一页,按 B 键。也可以用光标键向前、后、左、右移动。一般情况下,less 命令的命令提示符是显示在屏幕左下角的一个冒号。总之,可以使用命令行参数-M 看到更 多关于文件的信息,如下所示:

#less -M programs.txt

现在,当浏览文件的时候,less 命令将显示出这个文件的名字、当前页码及总的页码, 表示当前位置在整个文件中的位置的百分比数值。如下所示:

programs.txt line 91/1221 8%

这表明正在阅读的是文件 programs.txt,当前屏幕最顶端显示的是总数为 1221 行文本 的第 91 行。如果想运行其他程序,如 wc 字数统计程序,需要敲入一个惊叹号,后面再跟 上命令行,然后按回车键,如下所示:

!wc programs.txt

wc 程序将在屏幕上显示文件 programs.txt 的行数、字数和字符数。当这个命令执行完之后,显示 done 并等用户按回车键。

还可以使用 less 命令在一个文本文件中进行快速查找。方法如下:先按斜杠键(/), 再输入一个单词或者词组的一部分。less 命令会在文本文件中进行快速查找,并把找到的 第一个搜索目标高亮度显示。如果希望继续查找,请按斜杠键(/),再按回车键。如果想 退出阅读,请按Q键,就返回到 shell 命令行。

6.2.13 head和tail命令

有些时候,能够一次性地输出整个文件或者把文件中的多个空行压缩为一行是比较好的。但在很多情况下,需要对即将显示的文本行进行一些控制。head 和 tail 命令提供了对输出结果进行控制的能力。

head 命令可以显示它的标准输入的开始 10 行的内容。如果使用-n 选项,则显示前 n 行的内容。

tail 命令显示它的标准输入的最后 10 行的内容。如果使用-n 选项 , 则显示后 n 行的内 容。

6.2.14 In命令

Linux 操作系统同时支持硬链接和符号链接。下面将介绍这两种形式的链接的区别以及 如何使用链接。

ln 命令可以建立这两种形式的链接并把文件与其他的文件链接在一起。这两种形式的 链接的重要区别是:硬链接直接链接两个同时保存在相同文件系统上的文件;符号链接用 来在扩展到不同的文件系统上的子目录或者文件之间建立链接。

如果想使用 ln 命令建立一个硬链接,就必须在命令行中明确指定另外一个文件,然后

就可以使用这个文件来对应原始文件 , 如下所示 :

#cat > file1
This is file1
#ln file1 file2
#ls -l file*
-rw-r--r-- 2 duwei users 14 apr 5 10:00 file1
-rw-r--r-- 2 duwei users 14 apr 5 10:00 file2
#cat file2
This is file1

可以看到,文件 file2 和文件 file1 是完全相同的。如果删除了文件 file1,文件 file2 依 然存在。如果修改了文件 file1,所有修改都会反映到文件 file2 中;如果修改了文件 file2, 文件 file1 也会随之更新。虽然可以看到两个文件,每个文件的长度都是 14 个字符,但是 在硬盘驱动器上只有原始文件,它占用了 14 个字符的空间。

符号链接很有用,但是它也存在不足。下面的例子说明了原因。

首先,使用 ln 命令的-s 参数建立一个符号链接,如下所示:

#cat > file1
This is file1
#ln -s file1 file2
#ls -l file*
-rw-r--r-- 2 duwei users 14 apr 5 10:00 file1
-rw-r--r-- 2 duwei users 14 apr 5 10:00 file2->file1

从文件 file2 指向文件 file1 的箭头告知文件 file2 是文件 file1 的一个符号链接。符号链 接和硬链接的区别是:符号链接只不过是指向原始文件的一个假名而已。如果删除了符号 链接,原始文件不会发生任何变化。而一旦删除了原始文件,符号链接也就没用了:

#rm -f file1
#cat file2
cat : file2 : No such file or directory

因为原始文件 file1 不存在了,所以也就无法通过符号链接 file2 存取其中的内容。但 是符号链接比硬链接有一个优势:可以使用一个符号链接指向 Linux 文件系统中的某个子 目录。在下面的例子中,如果打算建立一个对子目录/usr/local/games 的硬链接, ln 命令会 显示出错信息并退出:

#ln /usr/local/games play

In : /usr/local/games : hard link not allowed for directory

但是可以使用一个符号链接,如下所示:

#ln -s /usr/local/games play
#ls -l play
lrwxrwxrwx 1 duwei users 16 Jan 5 14:12 play->/usr/local/games

现在,不必键入下面这么长的命令:

#cd /usr/local/games

而只需要使用命令:

#cd play

就可以进入子目录/usr/local/games。

6.2.15 find命令

find 命令是一个功能强大的操作命令。可以使用它在硬盘驱动器上查找文件。只用一条很简单的 find 命令就可以在硬盘驱动器上很容易地查找文件。例如,如果想查找子目录 /usr 中的 spell 命令,可以采用下面的方法:

#find /usr -name spell -print

还可以用 find 命令按照日期查找文件,或者指定一个日期范围。例如,如果想查找子 目录/usr/bin 中最近一百天内没有使用过的程序,可以使用下面的命令:

#find /usr/bin -type f -atime +100 -print

-atime 参数后面跟着代表日期的一个数字。如果想在子目录/usr/bin 中查找建立不到一 天的新文件或者 24 小时内修改过的文件,可以使用下面的参数:

#find /usr/bin -type f -mtime -1 -print

-mtime 参数后面跟一个表示天数的数字,用来查找修改过的文件。find 命令搜索字符 串时还接受统配符。例如,可以用 find 命令下面的格式来显示子目录/usr 中所有的 PostScript 格式的文件:

#find /usr -name *.ps - print

find 命令还有许多参数可以使用。

6.2.16 whereis命令

whereis 命令可以迅速找到文件,而且还可以提供这个文件的二进制可执行文件、源代 码文件和使用手册页存放的位置。例如,下面的命令给出 find 命令是放在子目录/usr/bin 中 的,而它的使用手册页是放在子目录/usr/man/man1 中的:

#whereis find find : /usr/bin/find /usr/man1/find.1.gz

可以使用下面的方法让 whereis 命令只查找这个程序的二进制可执行文件:

#whereis -b find
find : /usr/bin/find

这在只想查找某个程序的存储位置而不需要知道它的文档存储位置时特别方便。如果

whereis 找不到让它查找的项目,就会看见只返回一个空字符串,例如:

#whereis foo foo :

whereis 命令找不到某个文件的部分原因可能是这个文件没有存在于任何 whereis 命令 搜索的子目录中。whereis 命令检索的子目录是固定编写在它的程序中的。虽然这看起来是 个缺点,但把搜索限制在固定的子目录如/usr/man、/usr/bin 或/usr/sbin 中可以加快文件查找 的进度。whereis 命令在查找程序及其使用手册页时比使用 find 命令快。

6.2.17 locate命令

locate 命令可以加快文件搜索的速度。locate 命令使用的是一个文件名数据库,检索一个文件当然要比搜索整个硬盘要节省时间。使用 locate 命令查找文件要比使用 find 命令快得多,原因是 locate 命令直接检索这个数据库文件,从中找到匹配的文件名并显示它的查询结果。

locate 命令很容易使用。例如,如果想查找系统上的所有 PostScript 格式的文件,可以使用下面的命令:

#locate *.ps

文件查询结果几乎立刻就出现在屏幕上了。

locate 和 find 之间的区别是:find 命令在 Linux 操作系统安装完成后可以立刻开始进行 文件的查询;而 locate 命令则需要先为计算机上所有的文件建立一个数据库。

安装好 Linux 操作系统之后, locate 命令的数据库文件每天都会自动更新。如果想为 locate 命令的数据库文件生成一个新版本,可以使用 updatedb 命令,方法如下:

以根操作员的身份登录进入系统,在提示符下敲入:

#updatedb

6.2.18 gzip命令

gzip 命令用来压缩文件。这个程序不仅可以用来压缩大文件以节省磁盘空间,还可以和 tar 命令一起构成一种流行的压缩文件格式。Linux 软件经常是.tar.gz 格式,这可以节省很多空间。根据统计,gzip 命令对文本文件有 60% ~ 70%的压缩率。

gzip 命令很容易使用。如果想压缩某个文件,可以使用下面的命令:

#gzip mydir. tar

在缺省的状态下,gzip 命令将压缩文件,再加上一个.gz 扩展名,然后删除原来的文件。 如果想解压缩文件,可以使用 gzip 命令的对应程序命令 gunzip 或者 gzip 命令的-d 解 压缩参数。

必须保证这个用于解压缩的文件有.gz(或者.Z、-gz、.z、-z 和_z)扩展名,否则 gzip 命令和 gunzip 命令都会显示出错信息。如果想使用自己的扩展名,可以使用-S 参数,如下 所示: #gzip -S .gzipped mydir.tar

gzip 命令还可以处理用 zip 命令、compress 命令和 pack 命令压缩的文件包。如果想在 压缩或者解压缩的过程中看到更多的信息,可以使用-1 列清单参数看到文件在被压缩或解 压缩的时候的文件长度。在前一个例子中,压缩完子目录 mydir 之后,可以使用 gzip 命令 按照下面的方法得到有关的数据:

#gzip -l mydir. tar.gz compressed uncompr . ratio uncompressed_name 23 910 240 97.9% mydir. tar

最后, gzip 命令还有一个很有用的参数-t, 它可以用来测试压缩文件的完整性。如果文件正常, gzip 命令不会给出任何显示。

6.2.19 split命令

split 命令是用来传输大文件的最方便的命令之一。它最主要用途是分割被压缩的源文件。split 命令的基本语法是:

split [options] filename [output prefix]

这里的选项和输出前缀是可选的。如果没有提供输出前缀, split 将使用 x 作为输出前缀, 输出的文件将被标识为 xaa、xab 和 xac 等。在缺省情况下, split 在每个输出文件中放入 1000 行。然而,由于 1000 行的文件其大小是不固定的,因此通常使用-b 或--bytes 选项, 它的基本语法是:

-b bytes[bkm]

或

--bytes=bytes[bkm]

这里的 bytes 是文件大小的字节数,其中:

b 512 个字节

k 1KB(1024 个字节)

m 1MB(1 048 576 个字节)

如果将 8.5MB 的 package.tar.gz 文件分解为一些 1MB 的文件,使用如下的命令:

#split -b1000k package.tar.gz

#ls

package.tar.gz xaa xab xac xad xae xaf xag xah xai

可以看出,这个 split 命令一共生成了 9 个文件,文件名都以 x 开头。使用如下命令使 得输出文件的文件名都带有 package.tar.gz 字样:

#split -b1000k package.tar.gz package.tar.gz.
#ls

新概念 Linux 教程

package.tar.gz package.tar.gz.ac package.tar.gz.af package.tar.gz.ai

package.tar.gz.aa package.tar.gz.ad package.tar.gz.ag

package.tar.gz.ab package.tar.gz.ae package.tar.gz.ah

这将创建大小为 1MB 的文件,从而使它可以被拷贝到软盘中或者通过一个较慢的 modem 每次上载一个文件。

当这些文件到达它们的目的地后,可以利用 cat 命令把它们连接起来:

#cat package.tar.gz.* >package.tar.gz

6.2.20 chmod命令

使用 chmod 命令,可以把某个文件设置为私人专用的,这样其他非 root 的任何人就不能够再读或者写这个文件了。例如,第一次创建这个文件的时候,用户和其用户组可能都能够读和写这个文件,而其他的用户就只能读它。

如果使用 chmod 命令和八进制数 600 来改变权限,如下所示:

#chmod 600 afile#ls -l afile-rw----- 1 duwei users 0 Dec 26 14 : 50 afile

这个命令执行后会使这个文件只对用户是可读和可写的,因为它只对用户自己激活了 读写,其他人则都没有这个权限。如果想把文件的权限再改变为最初的存取权限,就需要 激活读写属性给用户自己和用户组,激活只读属性给其他所有的人,也就是说必须使用八 进制数 664,如下所示:

#chmod 664 afile
#ls -l afile
-rw-rw-r-- 1 duwei users 0 Dec 26 14 : 50 afile

也可以改变文件目录的权限让其他人可以列出子目录的文件清单,或者让其他人只能 存取这个子目录中的文件而不能列出这个子目录的清单内容。例如,可以使用下面的命令:

#Chmod 333 /home/dunei/temp

查看子目录内容的时候,会看到这样的信息:

#ls /home/duwei/temp

ls : /home/duwei/temp : Permission denied

下面的命令可以让其他人阅读这个子目录中的文件却又不能列出其中的内容清单:

#chmod 711 temp #ls -ld temp drwx--x--x 2 duwei users 1024 Dec 26 14 : 52 temp

6.2.21 chown命令

chown 命令是用来改变文件或者子目录的所有权的。用户可以被分配到不同的用户组,

使用 chown 命令,可以把所有权分配给不同的用户和用户组,允许他们对文件或者子目录 进行读、写或者删除操作。

例如,创建了一个文本文件,可以使用 chmod 命令把它与用户组成员或者系统中的其他用户共享。使用 chown 命令,可以特意告诉 Linux 操作系统让哪些用户或者用户组存取文件。可以使用 groups 命令查看自己属于哪一个用户组,如下所示:

#groups

users

这个例子显示出 duwei 用户属于 users 用户组。root 用户属于 root 组,如下所示:

#groups

root

如果想知道某个用户组都有谁,请查看一下/etc/group 文件,或者使用某个用户的用户 名,如下所示:

#groups mxn

mxn : users

这个例子显示这个 mxn 用户至少属于一个叫做 users 的用户组。如果想把某个文件分配给 users 用户组,使得 mxn 用户也可以对它进行存取,那么就可以使用 chown 命令的 user:group 语法,如下所示:

#chown :users myfile

也许会想把所有权直接分配给某个特定的用户,但这个工作只有 root 用户才可以完成, 使用下面的命令:

#chown mxn : mxn myfile

虽然文件 myfile 是由用户 duwei 做为系统管理员建立的,但是可以把它的所有权分配 给任何一个用户和任何一个用户组。如果只想改变一个文件或者子目录的用户组的所有权, 可以使用 chgrp 命令;如果想改变自己的用户或者自己的用户组,可以使用 newgrp 命令。

6.3 其他常用命令

除文件目录命令之外,还有一些其他的较为常用的命令,下面将有选择地介绍几个。

6.3.1 date命令

可以使用 date 命令的 date -d mm/dd/yyyy 格式设置日期。例如,输入命令 date -d 01/19/2010。

date 命令还有许多其他的格式参数,查询详细资料请输入 man date 命令。

6.3.2 time命令

使用 time 命令的 time -shh:mm:ss 格式设置时间。例如,输入命令 time -s 14:02:00。 time 命令还有许多其他的格式参数,查询详细资料请输入 man time 命令。

6.3.3 jobs/ps命令

有两种完全不同的方法可以用来查看当前计算机上都运行着哪些东西。一种方法是列 出当前正在运行的命令,另外一种方法是列出运行着的进程。

1. 确定有哪些命令正在运行

如果想查看后台中有哪些命令正在运行,请使用 jobs 命令。输入这个命令后将列出正 在后台运行的全部命令的清单——如果有。在这个清单中,最重要的信息通常是工作号, 就是最左边的数字。

2. 确定有哪些进程正在运行

如果想查看有哪些进程正在运行,请使用 ps 命令。它列出的清单中只包括在输入这个 命令的虚拟控制台上运行的进程。其中最重要的信息是最左边的数字,即 PID (Process ID , 进程号)。

6.3.4 bg命令

使用 bg 命令可以把当前正在屏幕上运行的命令调入到后台,这样,该终端就可以另做他用。请按照下面的步骤把某个程序调入后台:

(1) 输入 jobs 命令获取这个命令的工作号 job。

(2) 按下键组合 Alt+FX。其中 X 是某个空闲终端的功能键的编号。

(3) 登录进入刚才窗口中使用的同一个帐户。

(4) 输入 bg job, 把这个工作调入到后台。

这个操作通常有一个简单的完成方法:就是在显示该进程运行结果的终端上按下组合 键 Ctrl+Z。

6.3.5 fg命令

如果想把一个正在后台运行的命令调回前台以便对它进行操作,首先要按照前面的说 明获取工作号,然后使用 fg 命令的 fg job 格式即可。

6.3.6 kill命令

在 Linux 操作系统中可以使用好几种方法终止命令的执行。终止当前正在执行的命令 的最快方法是按下组合键 Ctrl+C。这个方法只有在能够从某个虚拟控制台上控制这个程序 的时候才奏效。

终止某个出错程序的另外一种办法是用 kill 命令。请按照下面的步骤进行操作:

(1) 输入 ps 命令获取进程的 PID,这个命令要求使用者是运行这个程序的人或者是 根用户。如果是根用户,不管是谁拥有它们,请输入 ps -aux 命令查看所有的进程。 (2)在清单中找到这个出错进程。程序的名称列在清单的最右边,通常很容易就可以 找到它。

(3) 记下 ps 命令输出清单中进程状态行最左边的 ID 数字。

(4) 输入 kill ID, 终止该进程。

(5) 使用 ps 命令查看该进程是否仍在运行中。

(6)如果它没有出现在清单中,就表明已经完成了操作。如果它还在运行中,请输入 kill9ID。这个命令强制终止该进程,并不进行善后工作,因此不到必要的时候最好不要使 用它。

6.3.7 df/du命令

GNU 中有关磁盘使用情况的实用程序 df 和 du 非常类似于相应的 UNIX 命令,不过它们还增加了几个出色的能够使它们的输出结果非常易读易理解的选项。

在缺省情况下,这两个程序都以块的形式显示输出结果,块的大小根据本地计算机的 不同而不同,但是它们的输出结果可以改变为按照 KB、MB 或 GB 的形式给出。这些输出 选项如下:

- · -b 或--bytes 用字节数显示结果。
- · -k 或--kilobytes 以每块 1KB(1024 字节)的形式显示结果。
- · -m 或--megabytes 以每块 1MB(1048576 字节)的形式显示结果。
- -h 或--human-readable 附加一个字母来表示大小(K代表千字节,M代表兆字节,G代表千兆字节)。

同其他 GNU 程序一样, du 命令也有很多选项。例如,要显示系统中所有文档所占的 空间,使用带--summarize 选项的 du 命令,如下所示:

#du -b --summarize /usr/doc 111 287 182 /usr/doc

6.4 习题

1. 查看帮助手册所在的目录,并查看一个命令的帮助手册。

2. 练习一般的文件操作,包括 ls、mkdir、cp、rm、find 和 chmod 等最常用的命令。

3. 在一个终端下使用 more 命令, 在另一个终端下使用 kill 命令将 more 命令的进程杀

死。

第7章 Linux 的系统工具

本章主要介绍 Linux 常用的系统工具。内容包括经典编辑器 vi、Linux 配置管理工具 Linuxconf、Red Hat Linux 包管理工具 RPM、备份以及备份工具等。

通过学习本章,读者可以对上述几个常用工具有一个较为全面的认识。当然,熟练使 用还有赖于实践。

7.1 经典编辑器vi

vi 是一种全屏幕编辑器,原意是 visual,它是一个立即反应的编辑程序,也就是说可 以立刻看到操作结果。vi 是 UNIX 和类 UNIX 操作系统中最为普遍的文本编辑器,是系统 自带的,几乎所有的类 UNIX 系统都捆绑了 vi。这意味着无论何时跨平台使用某种类 UNIX 操作系统,该操作系统上面都有一个强力的编辑器。

有很多 vi 的版本运行在不同的操作系统上, Linux 中也有 vi 的一个版本。学习 vi 的基本操作的好处在于其通用性,在不同的计算机上都得心应手。

熟悉 DOS 下的文本处理后 , 也许会感到 vi 不好用。事实上 vi 确实有些难学。刚开始 接触时 , 会感到 vi 是一个让人裹足不前的编辑器。一旦学会了 , 就会觉得非常方便。

另一方面, vi 是个强大的编辑器。一旦了解了 vi,就能很快编辑文件,因为它减少了 敲键次数。由于它在插入和命令两种情况下使用不同的模式,所以,它比大多数不基于这 种模式的编辑器速度快。vi 几乎能做任何事,只要知道如何去做。而且 vi 程序非常小。

7.1.1 vi的模式

vi 有两种模式,插入模式和命令模式。输入模式用来输入文字资料,而指令模式则用 来下达一些编排文件、存档、以及离开 vi 等的操作指令。vi 将命令模式和插入模式区分 开来,经常被认为是 vi 的一个大问题,但往往这也被认为是 vi 的优势所在。理解其中的区 别是掌握 vi 的关键。vi 启动时,开始处于命令模式,此时输入的任何字符都被视为指令。 在这种模式下,可以在文件中到处移动,改变文本的某个特定区域,剪切,复制和粘贴文 本,以及其他一些操作。插入模式是指可以真正插入文本。换句话说,命令模式用来在文 件中移动,而插入模式是用来在文件中键入文本。

有些命令,如 a, i, c, C, O, o 等可以从命令模式切换到插入模式。Esc 键或者^C (CRTL+C)可以脱离插入模式而返回到指令模式。要适应这种差别。这种差别使 vi 区别 于其他编辑器,也使标准键盘可以做更多事情。

7.1.2 打开和退出vi

可以直接在系统提示符下键入 vi filename 启动 vi 编辑器。vi 自动载入所要编辑的文件

或是开启一个新文档。进入 vi 后屏幕左方会出现波浪符号 , 凡是行首有该符号就代表此行 目前是空的。这就是标准的 vi 编辑窗口。例如:

进入 vi 并编辑名为 newfile 的文件的命令如下所示:

\$ vi newfile

屏幕出现:

- ~
- ~

--INSERT--

在~提示符后,可以输入文件。

要离开 vi,可以在指令模式下键入:q。要切换到指令模式下,可以用 ESC 键。如果不 知道现在处于什么模式,可以多按几次 ESC,系统会发出嘀嘀声以确定进入指令模式。 常用的退出 vi 的指令有:

- · :a! 离开 vi , 并放弃刚才在缓冲区内编辑的内容。
- · :wq 将缓冲区内的资料写入硬盘中,并离开 vi。
- · :ZZ 同 wq。
- ・:x 同wq。
- · :w 将缓冲区内的资料写入硬盘中,但不离开 vi。
- · :q 离开 vi ,若文件被修改过 ,则会被要求确认是否放弃修改的内容。此指令可与:w 配合使用。
- 7.1.3 vi的输入模式

vi 有以下 3 种输入模式分成:

- · 增加文字 (append)
- a 从光标所在位置后面开始新增资料,光标后的资料随新增资料向后移动。
- A 从光标所在列最后面的位置开始新增资料。
- 插入文字(insert)
- i 从光标所在位置前面开始插入资料,光标后的资料随新增资料向后移动。
- I 从光标所在列的第一个非空白字元前面开始插入资料。
- · 开始新行 (open)
- o 在光标所在列下方新增一列并进入输入模式。
- O 在光标所在列上方新增一列并进入输入模式。
- 7.1.4 vi命令模式
 - · 移动光标

由于许多编辑工作是靠光标来定位的,所以 vi 提供许多移动光标的方式。移动光标的指令如表 7.1 所示。

表 7.1 移	动光标的指令
---------	--------

	作用
0	移动到光标所在列的最前面
\$	移动到光标所在列的最后面
CTRL+d	向下半页
CTRL+f	向下一页
CTRL+u	向上半页
CTRL+b	向上一页
Н	移动到视窗的第一列
М	移动到视窗的中间列
L	移动到视窗的最后列
b	移动到上一个字的第一个字母
W	移动到下一个字的第一个字母
e	移动到下一个字的最后一个字母
^	移动到光标所在列的第一个非空白字符
n-	减号移动到上一列的第一个非空白字符,前面加上数字 n 指定移动到以上 n 列
n+	加号移动到下一列的第一个非空白字符,前面加上数字 n 指定移动到以下 n 列
NG	直接用数字 n 加上大写 G 移动到第 n 列
fxFx	往右移动到 x 字符上再往左移动到 x 字符上
txTx	往右移动到 x 字符前再往左移动到 x 字符前
; ,	配合 f&t 使用 , 重复一次配合 f&t 使用 , 反方向重复一次
/string?string	往右移动到有 string 的位置往左移动到有 string 的位置
nN	配合/&?使用,重复一次配合/&?使用,反方向重复一次
n(左括号移动到句子的最前面,前面加上数字 n 指定往前移动 n 个句子
n)	右括号移动到下一个句子的最前面,前面加上数字 n 指定往后移动 n 个句子
n{	左括弧移动到段落的最前面,前面加上数字 n 指定往前移动 n 个段落
n}	右括弧移动到下一个段落的最前面,前面加上数字 n 指定往后移动 n 个段落

删除与修改

vi 认为编辑是指文字的新增、修改以及删除,甚至包括文字区块的搬移、复制等等。 这里先介绍 vi 如何做删除与修改。修改和删除的指令如表 7.2 所示。

表 7.2	修改和删除的指令

指令	作用
х	删除光标所在字符
dd	删除光标所在的列

(续表)

指令	作用
r	修改光标所在字符 , r 后接要修正的字符
R	进入取代状态,新增资料会覆盖原先资料,直到按 ESC 键回到指令模式下为止
S	删除光标所在字符,并进入输入模式
S	删除光标所在的列 , 并进入输入模式
u	恢复被删除的资料
U	恢复光标所在列的所有改变

• 较灵活的编辑指令

这些编辑指令是由指令与范围所构成。例如,dw是由删除指令 d 与范围指令 w 所组成,代表删除一个字 d(elete)w(ord)。相关指令和范围如表 7.3 和表 7.4 所示。

表 7.3 指令列表

指令	作用
d	删除(delete)
у	复制 (yank)
р	放置(put)
с	修改 (change)

表 7.4 范围列表

范围	意义
e	光标所在位置到该字的最后一个字母
W	光标所在位置到下一个字的第一个字母
b	光标所在位置到上一个字的第一个字母
\$	光标所在位置到该列的最后一个字母
0	光标所在位置到该列的第一个字母
)	光标所在位置到下一个句子的第一个字母
(光标所在位置到该句子的第一个字母
}	光标所在位置到该段落的最后一个字母
{	光标所在位置到该段落的第一个字母

组合这些指令编辑文件可以提供更多编辑文字的能力。例如:

:10dd	删除 10 行。
·10vv	复制 10 行

.10уу	
	_

- :10j 往下移 10 行。
- :10,20d 删除第 10 行至第 20 行的资料。
- :10d 删除第 10 行的资料。
- :%d 删除整个编辑缓冲区。

:10,20co30 将第 10 行~第 20 行的资料复制至第 30 行之后。

:10,20mo30 将第 10 行~第 20 行的资料搬移至第 30 行之后。

对于整列操作, vi 另外提供了更方便的编辑指令。前面曾经提到过删除整列文字的指 令 dd 就是其中一个; cc 可以修改整列文字; yy 是复制整列文字; 指令 D 则可以删除光标 到该列结束为止所有的文字。

查找命令

在 vi 中可以查找某一字符, 使光标移至该处。

/字符串	在光标之后寻找该字符串。
?字符串	在光标之前寻找该字符串。
n	往下继续寻找下一个相同的字符串。
N	往上继续寻找下一个相同的字符串。

• 环境设定命令

在编辑文件或程式时,有时会遇到需要缩进的状况,:set ai 即提供自动缩进的功能,相关的命令如下:

:set nu	设定资料的行号。
:set nonu	取消行号设定。
:set ai	自动缩进。
:set noai	取消自动缩进。

例如:

.vi xteamnews

.(进入编辑窗口后)

this is the test for auto indent

在这里设自动缩进:set ai

start indent

data

data

data

在这里取消自动缩进:set noai

the end of auto indent.

7.1.5 文件指令

```
    退出编辑
```

```
:q 结束编辑(quit)
```

如果不想存档而要放弃编辑过的文件则用:q。强制离开。

```
:w 存档 (write)
```

其后可加所要存档的档名。

可以将文件指令合在一起,例如:wq 即可存档后离开。

zz 功能与:wq 相同。

另外值得一提的是 vi 的部分存档功能。可以用:n,mw filename 将第 n 行~第 m 行的文字存放到所指定的 filename 里。

例如:

:10,20w xteamnews 将第 10 行~第 20 行的资料写入 xteamnews 文件。

:10,20w>>test 将第 10 行~第 20 行的资料加在 xteamnews 文件之后。

• 恢复被中断的文件

在编辑过程中,如果系统意外中断,而缓冲区的资料还未被存入硬盘时,可重新启动 并回到系统中。执行下列命令即可恢复中断前的文件内容:

\$vi -r filename

· 编辑多个文件

vi 也提供同时编辑多个文件的功能。

例如: \$vi file1 file2...

当第一个文件编辑完成后,可利用:w 将该缓冲区存档,然后再利用:n 引入下一个文件。

7.1.6 vi命令及其作用列表

如表 7.5 所示是较全面的依字母顺序的 vi 的命令和其作用。

表 7.5 vi 命令及其作用列表

命令	作用	命令	作用
数字	执行 n 次后面的命令	Н	屏幕上的第一行
:	进入 ex 模式	^h	在输入模式下是退格 ,指令模式下是左移
)	下一句	i	在当前的位置前面插入
(上一句	Ι	在本行的第一个非空白字符前面插入
}	下一段	^i	输入模式下是制表键,指令模式下未用
{	上一段	j	下移
]]	下一节	J	把下一行合并到本行
[[上一节	^j	指令模式下是下移 , 输入模式下是建一个
			新行
0	行的开头	k	上移
\$	行的末尾	K	未用
٨	行的第一个非空白字符	^k	未用
+	下一行的开头	1	未用
-	上一行的开头	L	屏幕中的最后一行
空格	下一个字符	^1	重绘屏幕
回车	下一行	m	把当前位置标记到寄存器中

(续表)

命令	作用	命令	作用
/	向前搜索	М	屏幕的中间行
?	向后搜索	^m	回车
%	查找当前小 (中 , 大) 括号的匹配字	n	重复上次查找
	符		
,	逆向重复上一个 f、F、t 或 T 操作	Ν	逆向重复上次查找
;	重复上一个 f、F、t 或 T 操作	^n	指令模式下是下移
	重复上一个操作	0	在当前行的下面建一个新行
•	转到标记处	0	在当前行的上面建一个新行
•	转到标记所在行的开头	^0	未用
~~	查找前回到上一个做标记的位置	р	在当前位置的后面粘贴
••	查找前回到上一个做标记所在行的开	Р	在当前位置的前面粘贴
	头		
~	切换字符的大小写	^p	指令模式下是上移
"	保存到寄存器中	q	未用
!!	重复上一次 SHELL 命令	Q	退出并且启动 ex
!	把后面的文本送给命令 , 取代输出	^q	未用
>>	右移当前段落一个移动宽度	r	覆盖当前的字符
	shift width		
<<	左移当前段落一个移动宽度	R	在输入模式下一直覆盖字符
	shift width		
>%	从当前到匹配小(中、大)括号之间	^r	在指令模式下重绘屏幕
	的文本右移		
<%	从当前到匹配小(中、大)括号之间	S	替换
	的文本左移		
I	移动到某一列	S	替换整行
а	在当前位置之后添加	t	到
А	在当前行的末尾添加	Т	向后到
^a	没有使用	^t	移到下一个移动宽度 shift tab 处
b	上一个单词的开头	u	撤消上一次操作
В	上一个单词的开头,忽略标点符号	U	撤消对本行的所有修改
^b	上移一屏	^u	上移半屏
с	覆盖	v	未用
С	覆盖到行尾	V	未用
^ _C	终止输入模式 , 在指令模式下没有用	$^{\text{V}}$	未用
d	删除	W	下一个单词的开头
D	删除到行尾	W	下一个单词的开头 , 忽略标点符号

(续表)

命令	作用	命令	作用
^d	下移半屏,如果在输入模式下则移到	$^{\rm W}$	指令模式下未用 ,在输入模式下是到上一
	上一个移动宽度 Shift Tab 处		个单词的开头处
e	单词的末尾	х	删除当前字符
?E	单词的末尾,忽略标点符号	Х	删除前一个字符
^e	屏幕下卷一行	$^{\Lambda}X$	未用
f	查找	у	复制
F	向后查找	Y	复制整行
^f	下移一屏	^y	屏幕上卷一行
g	未用	Z	重新配置当前行周围的屏幕
G	跳至(缺省是到文件末尾)	ZZ	保存之后退出
^g	显示状态栏	^ _Z	未用
h	左移		

7.1.7 vi命令模式的输入选项(:命令)

如表 7.6 所示是 vi 的:命令列表。

表 7.6 vi 的:命令列表

命令	作用
:r < 文件 >	把<文件>读入到当前文档
:r!<命令>	把<命令>的输出插入到当前文本中
:nr <文件>	把<文件>插入到第 n 行
:!<命令>	运行<命令>,然后返回
:sh	转到 SHELL
:so <文件>	读取<文件> , 再执行文件里的 ex 命令
:x	保存之后退出
:wq	保存之后退出
:l1,l2w <文件>	把第11和第12行之间的文本写到<文件>中。如果没有指定<文件>, vi 就假定
	是当前文件,如果没有指定11,12,就假定是整个文件
:w>><文件>	添加到<文件>末尾。也可以使用行号
:w!	覆盖当前文件
:q	退出
:q!	不存盘就退出
:e <文件>	不离开 vi 编辑<文件>
:e!	重新编辑当前文件,忽略所有的修改
:n	编辑下一个文件
:e +n <文件>	从第 n 行开始编辑<文件> , 如果超过了末尾 , 就从末尾开始编辑
:e#	编辑替换文件(如果使用了命令:e<文件>,替换文件指的就是原始文件)

1	はキ	`	
	ノビスセ	•	
•	- デール		

命令	作用
:args	显示所有要编辑的文件
:rew	重新回到第一个编辑的文档
:map m n	创建一个宏 (使 m 做 n)
:map! m n	创建一个输入模式的宏(使 m 做 n)
:unmap m	删除宏 m
:unmap! m	删除输入模式的宏 m
:ab <1> <2>	缩写,敲<1>的时候用<2>取代
:unab <1>	取消缩写<1>
:set <选项>	设置<选项>

7.1.8 设置选项

设置选项的语法如下:

:set <选项> <参数> (如果需要参数)

也可以在一行中指定多个选项。

:set <选项>? 显示当前这个选项的参数值;:set all 则显示所有的选项值。

对无须参数的选项,使用:set no<选项>就可以将它关闭。

如表 7.7 所示是 set 命令后的选项及其含义。括号内的是这个选项的缩写形式。

选项	缺省值	含义
autoindent(ai)	noai	使新行自动缩进,和上(下)行的位置对齐
autoprint(ap)	ap	每条命令之后都显示修改之处
autowrite(aw)	noaw	在:n , :!命令之前都自动保存文件
beautify(bf)	nobf	在输入的时候忽略所有的控制字符(除制表键 tab,换行
		newline,进纸 formfeed)
directory= (dir=)	/tmp	存放缓冲区的目录名
edcompatible	noedcompatible	在替换的时候使用类 ed 的用法
errorbells (eb)	errorbells	出错的时候响铃
exrc (ex)	noexrc	允许在主目录(home)之外放.exrc 文件
hardtabs= (ht=)	8	设置硬制表的边界
ignorecase (ic)	noic	正则表达式里忽略大小写
lisp	nolisp	打开 lisp 模式
list	nolist	显示所有的制表键和行尾
magic	magic	可以使用更多的正则表达式
mesg	mesg	允许向终端发送消息
number (nu)	nonumber	显示行号

表 7.7 set 命令的选项及其含义

选项	缺省值	含义
open	open	允许开放和可视化
optimize (opt)	optimize	优化吞吐量,不打印时回车
paragraphs= (para=)	IPLPPPQPPLIbp	设置{&}的分界符
prompt	prompt	指令模式下的输入给出:的提示符
readonly (ro)	noro	除非用!号,否则不准保存文件
redraw	noredraw	编辑的时候重绘屏幕
remap	remap	允许宏指向其他的宏
report=	5	如果影响的行数大于这个数就报告
scroll	1/2window	下卷屏幕时滚动屏幕的数目,同样这也是 z 命令输出的行
		数(z 输出 2 倍滚屏的大小)
sections=	SHNHHHU	定义节尾(当使用命令[[和]]时)
shell= (sh=)	/bin/sh	缺省的 SHELL , 如果设置了环境变量 SHELL , 就使用变量
shiftwidth= (sw=)	8	当使用移动(shift)命令时移动的字符数
showmatch (sm)	nosm	显示{ , } , (,) , [, 或者]的匹配情况
showmode	noshowmode	显示处在什么模式下
slowopen (slow)		输入之后不要立刻更新显示
tabstop= (ts=)	8	设置制表停止位(tabstop)的长度
taglength= (tl=)	0	重要标记的字符个数(0表示所有的字符)
tags=	/usr/lib/tags	定义包含标记的文件路径
term=		设置终端类型
terse	noterse	显示简短的错误信息
timeout (to)	timeout	一秒钟后键盘映射超时
ttytype=		设置终端类型
warn	warn	显示"Nowritesincelastchange"信息
window= (w=)		可视模式下窗口的行数
wrapmargin= (wm=)	0	右边距。如果大于 0, 最右边的单词将折行, 留出 n 个空白
		位置
wrapscan (ws)	WS	查找到文件尾后再重头开始
writeany (wa)	nowa	可以保存到任意一个文件

7.2 多功能管理工具linuxconf

linuxconf 是 Red Hat Linux 中最常使用的配置管理工具,功能丰富强大。linuxconf 命令 是一个图形化配置和管理工具。linuxconf 命令也可以在没有 X11 的支持下使用。

如果 root 用户使用了 X Window, linuxconf 命令会出现在 GNOME 面板的弹出菜单的 System 级联菜单中。

(续表)

如果 root 用户没有使用 X Window,在终端窗口的命令行输入 su -c linuxconf,一个文本方式的图形界面就会显示出来。

下面将介绍 root 用户使用 X Window 环境的情况。

在系统中选择 Program System 菜单中选中 LinuxConf,出现的使用界面如图 7.1 所示。



图 7.1 linuxconf 的使用界面

这个使用界面有3个标签,分别是Config、Control和Status。

在 Config 页中, 可以对系统的网络、用户、文件系统和引导模式进行配置。

在 Contorl 页中,可以控制系统的一些特性。

在 Status 页中,可以获取系统的一些信息,如日志等。

下面将以用户管理和启动管理为例,介绍 linuxconf 的一般用法。

7.2.1 用户管理

在 Control 页中单击 User accounts 中的 Normal, Normal, 会打开折叠, 出现 User accounts、Group definitions 和 Change root password 项目。这些项目的含义是明显的, User accounts 用作一般的用户管理; Group definitions 用作组管理; Change root password 用于改变 root 用户的口令。本小节只介绍 User accounts 这一项。

单击 User accounts 项目,在 linuxconf 的用户界面右方出现 User accounts 选项卡。其中的列表中列有当前系统中的用户。列表下方有 Quit、Add 和 Help 按钮。如图 7.2 所示。

下面在系统中添加一个用户名为 mxn 的用户为例,介绍使用 LinuxConf 进行用户管理的步骤:
(1) 单击列表下方的 Add 按钮。这时,在 User accounts 选项卡旁边添加了一个 User account creation 选项卡,该选项卡有 3 个子选项卡,分别为 Base info、Params 和 Privileges 选项卡。

zgluo: Linuxconf 1.19 (subrev 2)		_ = >
Config Control Status	Users accounts	
Networking Users accounts Group definitions Change root password Special accounts Policies File systems Miscellaneous services boot mode	You can edit, add, or delete users Select [Add] to add a new definition	
	AccountNameUidapacheApache48ftpFTPUseigamesgames12gdmgopher12gophergopher13IpIp4mailmail8mailnull47namedNamed25newsnews9operatoroperator11rpcPortmapr32	
Quit Act/Changes Help	Quit Add Help	J

图 7.2 linuxconf 的 Users accounts (用户管理)选项卡

· 在 Base info 选项卡中,可以填写用户的登录名、用户全名、组别和主目录等内容。 此例中 mxn 的 Base info 信息如图 7.3 所示。

zgluo: Linuxconf 1.19 (subrev 2)		
zgluo: Linuxconf 1.19 (subrev 2) Config Control Status Networking ⊕ Cient tasks ⊕ Server tasks ⊕ Networking ⊕ Networking ⊕ Server tasks ⊕ Normal ⊕ Normal ☐ Group definitions Change root password ⊕ Special accounts	Users accounts User account creation You must specify at least the login name and the full name Base info Params Privileges Login name	T The account is enabled [m×n
de Policies ⊕ File systems ⊢ Miscellaneous services ⊕ boot mode	Full name group (opt)	maxingning
	Supplementary groups	
	Home directory(opt)	/home/mxn
	Command interpreter(opt)	/bin/bash
	User ID(opt)	
Quit Act/Changes Help	Accept Cancel	TasksHelp

图 7.3 填写的用户 Base Info (基本信息)选项卡

• 在 Params 的选项卡中,可以指定用户的参数,包括用户的有效期等信息。可以指 定这些参数,也可以不指定而使用系统默认值。如图 7.4 所示。

zgluo: Linuxconf 1.19 (subrev 2)		
Config Control Status	Users accounts User account creation	_
中 Networking 中 Client tasks 中 Server tasks 中 Misc	You must specify at least the login name and the full name	
다 Users accounts 다 Normal	Base info Params Privileges	
→ Group definitions → Group definitions → Change root password ⊕ Special accounts ⊕ Policies	Must keep # days 🔶 Ignored 🔶	
File systems Miscellaneous services Hont mode	Must change after # days 99999	
	Warn # days before expiration 🛷 Ignored 💠	
	Account expire after # days 🔶 Ignored 🕹	
	Expiration date (yyyy/mm/dd)	
Quit Act/Changes Help	Accept Cancel Del Tasks Help	

图 7.4 Params (参数)选项卡

· 在 Privileges 选项卡中,可以指定用户的权限,使用户可以执行某些特定的操作或 者存取特定的目录文件等。也可以不指定用户权限,使用系统默认值。如图 7.5 所 示。

zgluo: Linuxconf 1.19 (subrev 2)				_ = ×
Config Control Status	Users accounts User account crea	ation		
	You must specify at least the login and the full name	ı name		
⇔ Users accounts ⇔ Normal	Base info Params Privileges			
User accounts Group definitions	General system control User ac	counts		
Change root password Change root password Policies	May use linuxconf	🗢 Denied	҅ Granted	😞 Granted/silent
File systems Miscellaneous services boot mode	May activate config changes	🗢 Denied	҅ Granted	🗢 Granted/silent
	May shutdown	🗢 Denied	҅ Granted	🗢 Granted/silent
	May switch network mode	🗢 Denied	🕹 Granted	😞 Granted/silent
	May view system logs	I Denied	҅ Granted	🗢 Granted/silent
	SuperUser equivalence	* Denied	🗢 Granted	⇔ Granted/silent
Quit Act/Changes Help	Accept Ca	ncel Del	Tasks	Help

图 7.5 Privileges (权限)选项卡

(2) 单击右下方的 Accept 按钮,确认修改。

最后,系统会询问是否为用户设置口令,设置即可。

到此,用户 mxn 就添加完毕。可以从一个终端上使用 mxn 登录,检验 mxn 是否有效。 可以看出,linuxconf 工具是一个方便用户使用的图形化管理工具,它管理的内容在命 令行的方式下也可以进行管理,尤其在 Linux 系统作为服务器的情况下,人们更倾向于使 用命令行的方式。对于初次使用 Red Hat Linux 的用户,linuxconf 是一个容易掌握的工具。

7.2.2 启动管理

启动管理主要是指对LILO的管理。使用linuxconf进行启动管理,可以从各方面对LILO进行管理。管理的内容包括:LILO默认的配置、LILO的Linux配置、其他操作系统的LILO配置和默认引导配置等。

在 linuxconf 的使用界面下,单击 Lilo 项, Lilo 项展开为 Configure LILO defaults、 Configure LILO linux configurations、Configure LILO other OS configurations、Change default boot configuration、Add to LILO a new kernel 和 Add to LILO a kernel you have compiled。下 面将介绍前面 4 项。

· Configure LILO defaults

单击 Configure LILO defaults,右面出现 Lilo defaults选项卡。该选项卡包含 2 个子选项卡,分别为 Base options 和 Extra options选项卡。

Base options 选项卡如图 7.6 所示。

zgluo: Linuxconf 1.19 (subrev 2)		
Config Control Status	Lilo defaults	
Networking Users accounts HE systems	Base options Extra options	LILO is used to boot this system
- Miscellaneous services	Install hoot sector on	/dev/bda
Configure LILO defaults (Linux boot loader) Configure LILO linux configurations Configure LILO differ OS configurations		
Change default boot configuration Add to LILO a new kernel	Blos boot mode Boot table encoding	Linear
- Add to LLO a kerner you have complied ⊕ Mode	Boot delay in seconds	0
		■ Present the LILO boot: prompt
	Prompt timeout in seconds	5
	Message file(opt)	/boot/message
Quit Act/Changes Help	Accept	Cancel Help

图 7.6 Base options 选项卡

在 Base options 选项卡中, Install boot sector on 可以选择将 LILO 安装在下拉菜单中的 某一项中,如/dev/hda 或者/dev/hdaX中。选择/dev/hda,将把 LILO 安装在 MBR中,选择

/dev/hdaX,其中X代表安装 Linux 系统的分区号,将把 LILO 安装在 Linux 系统的分区中。 将 Bios boot mode 和 Boot table encoding 保留默认设置即可。 Boot delay in seconds 可以设置启动的延迟,

Prompt timeout in seconds 可以设置提示保留的时间,按照个人爱好设置即可。 Message file (opt) 可以设置消息文件,如果没有定制,保留默认设置即可。 Extra options 选项卡如图 7.7 所示。

zgluo: Linuxconf 1.19 (subrev 2)		
Config Control Status	Lilo defaults	
Networking Users accounts	Base options Extra option	s
File systems Miscellaneous services	root partition	
boot mode build Configure LILO defaults (Linux boot loader) Configure LILO linux configurations Configure LILO linux configurations Change default boot configuration Add to LILO a new kernel Add to LILO a kernel you have compiled Adde L LICO a kernel you have compiled Adde L LICO a kernel you have compiled	Ramdisk size (opt) boot mode VGA mode	r Read only
	Boot options	
	Initial ramdisk(opt)	
	Password (opt)	
		☐ Restricted access
Quit Act/Changes Help	Accept	Cancel Help

图 7.7 Extra options 的选项卡

在 Extra options 选项卡中,可以设置根分区、引导模式、引导选项和口令等内容。 root partition 可以从下拉菜单选择设置根分区。一般不要修改这个选项。 Password 可以设置引导时需要提供的口令。

• Configure LILO linux configurations

单击 Configure LILO linux configurations,右面出现 lilo linux configurations选项卡,其中的文本框中出现的是 Linux 的标签和引导区和内核映像。

双击列表框中的条目,出现 Linux boot configuration 选项卡,如图 7.8 所示。刚才在列 表框中出现的属性现在变为可以修改的条目,包括标签、引导区和内核映像,另外还包括 Extra options 中出现的一部分选项。

Configure LILO other OS configurations

单击 Configure LILO other OS configurations,右面出现 lilo other OSs configurations选项卡。其中的文本框中出现的是其他操作系统的标签和引导区。如图 7.9 所示。

双击列表框中的条目,出现 Other operating system setup 选项卡。见图 7.9 所示。刚才 在列表框中出现的属性现在变为可以修改的条目,包括标签和引导区。

zgluo: Linuxconf 1.19 (subrev 2)		
Config Control Status	Lilo defaults Lilo linux confi	igurations Linux boot configuration
	You control how to boot on configuration	linux
Configure LLO other OS configurations Configure LLO other OS configurations Change default boot configuration Add to LLO a new kernel Add to LLO a kernel you have compiled Add to LLO a kernel you have compiled Add to LLO a kernel you have compiled	Kernel image file root partition Bamdisk size (opt)	/boot/vmlinuz-2.2.16-22 /dev/hda5
	boot mode	r Read only
	VGA mode	
	Boot options	
	Initial ramdisk(opt)	
	Password (opf)	
		Restricted access
Quit Act/Changes Help	Accept	Cancel Del Help

图 7.8 Configure LILO linux configurations 的使用界面

zgluo: Linuxconf 1.19 (subrev 2)		
Config Control Status	Lilo other OSs config	urations Other operating system setup
Networking Users accounts File systems Miscellaneous services boot mode Configure LILO defaults (Linux boot loader); Configure LILO dirux configurations Configure LILO ther OS configurations Configure LILO anew kernel Add to LILO a new kernel Add to LILO a kernel you have compiled GMode Lefault boot mode	You control how to I	ooot another operating ion of your harddrive(\$)
	Label	dos
	partition to boot	/dev/hda1
Quit Act/Changes Help	Accept	Cancel Del Help

图 7.9 Configure LILO other OS configurations 的使用界面

Change default boot configuration

单击 Change default boot configuration,右面出现 Default boot configurations 选项卡。选项卡中的单选框是 Linux 和其他操作系统的分区。如图 7.10 所示。如果希望系统默认的操作系统是 Linux,那么单击 Linux 的单选框;反之单击另一个单选框。

zgluo: Linuxconf 1.19 (subrev 2)	
Config Control Status	Default boot configuration
Config Control Status	Pick the configuration which will become the default LILO configuration
Quit Act/Changes Help	Accept Cancel Help

图 7.10 Change default boot configuration 的使用界面

7.3 软件管理工具RPM

RPM 是 Redhat Package Manager 的缩写,也就是 Red Hat 的包管理工具,但实际上它不仅限于 Red Hat 系统使用。本节将详细介绍 RPM 及其工具的使用。

7.3.1 下载应用程序

当下载新的文件和程序的时候,或者当准备把文件或者程序发送给其他人的时候,软件包管理就成为一个必须认真对待的问题。掌握软件包和文件格式方面的知识可以减少麻烦与混乱,可以最快、最好、更容易地完成这些工作。

· 选择下载的软件包格式

在下载某个软件包的时候,会遇到不止一种文件格式。通常会遇到的选择有源代码、 二进制代码或者 RPM 包。具体选择哪一种完全取决于用户编程水平的高低、系统的安装是 否标准、需要对下载获得的软件进行大量定制还是就按照它们的缺省方式运行等几方面因 素。

下载源代码

获得源代码之后,就需要对它进行编译。采用这种方法主要有以下几点原因:

(1)熟悉编程,并且事先明确地知道在编译这个软件包之前还需要对源代码进行一些 改动调整。 (2) 已经尝试过其他各种方法,但是效果均不理想。

(3)只能通过源代码的形式获得这个程序。

选择下载某个软件包的源代码是最为复杂的一个解决方案,因为必须先对它进行编译 之后才能安装。

· 下载二进制代码

通常选择下载二进制代码的操作过程都比较简单。这是因为二进制代码是程序已经编译好的版本,不用自己编译。选择下载二进制代码,就意味着无法修改代码本身,并且正确安装好二进制代码的速度要取决于它附带的 README 文件的质量。下载二进制代码的一个问题是程序可能无法与安装的 Linux 操作系统完全兼容。

下载 RPM 包

目前 Red Hat 发行版本的 Linux 软件包的下载格式选用的是 RPM 包 (Red Hat Package Manager, Red Hat 软件包管理器)文件。但是,千万不要因为它是一个 RPM 包就相信它 是安全的,最值得信赖的 RPM 包的有以下几个来源:

(1) Red Hat 公司的 Web 和 FTP 站点。

(2) 编写该软件的公司的 Web 和 FTP 站点。

(3) 知名并且可信的 Linux 软件下载站点。

(4) 带有可信机构 PGP (Pretty Good Privacy, 一种加密编码程序) 签名的 RPM 包。

(5) 用户自己制作的 RPM 包。

一个值得信赖的站点会对它的下载软件进行精心维护。每个软件包的有效性和安全性 必须经过测试。即便如此,也应该尽最大可能检查每个软件包,而不是盲目地相信下载来 的东西,这才是极为明智的做法。

这些原则适用于各种下载情况,不仅仅局限于 RPM 包。

· 软件打包方法

在下载文件时,会遇到许多种文件扩展名。以下是几种常见扩展名的格式,其中一些 在相关章节中还会详细介绍。

• .tar 档案文件

使用 tar 命令对多个文件进行打包的同时,还保留了这些文件的目录结构。这个档案制 作程序不进行文件压缩,除非使用了-z 参数,也就是在完成对文件的打包之后调用 Gzip 程 序对生成的档案文件进行压缩。它的作用是把指定的文件或者目录和它们的关联文件与存 储位置一起打包。总之,tar 命令通常和一个数据压缩程序或者数据压缩参数一起使用来减 少档案文件占用的空间。

可以从文件带有.tar 扩展名识别出它是一个 tar 文件。

· .Gzip 压缩

在 Linux 操作系统的用户中比较流行的一个文件压缩方法是使用 Gzip 程序。它是由 GNU 组织提供的压缩程序。用户可以从文件带有.gz 扩展名识别出它是一个采用 Gzip 程序 进行压缩的文件。有时候也会使用.z 后缀。

・ .Z 压缩

文件压缩的另外一种方法是使用 compress 文件压缩程序。可以从文件带有.Z 扩展名识 别出它是用 compress 程序压缩过的文件。

.zip 压缩

zip 压缩在制作和打开一个 pkzip 兼容的压缩文件时很有用。可以从文件带有.zip 识别 出用它是 zip 程序压缩过的文件。

• Red Hat 软件包管理器

另外一个传输并安装各种程序或程序包的方法是采用 RPM。可以从文件带有.rpm 后缀 识别出它是 RPM 包。

7.3.2 RPM基础

RPM 是一个功能强大的系统,并且是最有新意的实用程序之一。它是一个使用源代码 格式发布的软件发布机制。它也压缩二进制发布,以便快速安装。RPM 可用来安装、卸装、 升级、查询、验证和构造软件包。它使得软件的安装和升级能够以最小的代价进行。

RPM 非常流行,其他几种 Linux 发布版本都认为它是首选的安装方式。并且,还有一些工具能够将软件包在 RPM 和其他包格式之间互相转换。

组成 Red Hat Linux 7.0 的所有 RPM 包都包含在 Linux 安装光盘的 RedHat/RPMS 目录中。利用 RPM 构造的软件包是一个档案文件,它包含文件和一些相关的信息,如名字、版本及说明等。例如,Samba 就是其中的一个软件包,它包含在 samba-2.0.7.21ssl.i386.rpm 文件中。RPM 文件的不同扩展名分别表示不同的处理器体系结构(alpha、sparc 或 i386)或源发布(src)。本章假设目标平台为 Intel386 体系结构。

具体的 RPM 命名规则如下:

name-version-release.arch.rpm

这里的 name 是软件包的名字; version 是软件包的版本; release 是软件包版次; arch 是软件包适用的硬件结构; rpm 是缺省的扩展名。这种命名方法十分方便, 因为单从软件 包的名字就可以确定该软件包的一些基本信息。

与发布软件的传统方法 tar.gz 相比, RPM 软件包具有如下优点:

- · 升级 能够在不丢失原有配置文件的情况下完成软件新版本的安装。
- · 卸装 在多个位置安装的文件的软件包能够被很干净地删除。
- · 验证 安装软件包后,可以验证它是否能够正常运行。
- · 查询 可以很容易地获得某一文件所属软件包的有关信息。

除这些特性外, RPM 还可用于许多不同风格的 Linux 和 UNIX。这些因素使得它成为发布软件包的杰出实用程序之一。Red Hat Linux 7.0 中包含 4.0 版的 rpm 命令。

下面简要介绍 RPM 通常使用的一些方式:

- · rpm -Uvh package.i386.rpm 让 RPM 安装或升级一个软件包文件。
- · rpm -e package 从系统中移去安装的软件包。注意,没有.386.rpm 扩展名。
- · rpm -q package 对安装软件包的查询。发行号可以忽略。
- · rpm -qa 产生系统上安装的所有软件包的一个列表。

- · rpm -ql package 产生某个软件包所安装的所有文件的一个列表。
- · rpm -qf file 产生特定文件所对应的软件包的名称。
- · rpm -qpl package.i386.rpm 产生一个包含在软件包文件中的所有文件的列表。
- · rpm -qi package 产生一个所安装软件包的详细描述信息。
- · rpm -qpi package.i386.rpm 产生一个软件包文件的详细描述信息。
- · rpm2cpio package.src.rpm 将所有文件打成一个软件包。
- rpm --rebuild package.src.rpm 在/vsr/src/redhat下解包并建立一个二进制的RPM发 布文件。

下面将详细介绍 RPM 的主要运行方式和常用选项。

7.3.3 RPM主要运行方式和常用选项

RPM 运行的主要方式包括:

- · 安装 rpm-i
- · 卸装 rpm -e
- · 查询 rpm -q
- · 验证 rpm-V

括号中的选项用来调用主要运行方式。这几种主要运行方式将在后面详细介绍。这些 主要运行方式都能够识别下面两个选项:

- · -vv 显示所有调试信息。它有利于确切地查看 RPM 所执行的工作。
- · --quiet 显示非常少的信息,只包含错误信息。

除这些主要运行方式之外,还有几个非常有用的次要运行方式:

- · 显示版本 rpm --version
- · 提供帮助 rpm --help
- · 显示资源 rpm --showrc
- · 重新创建数据库 rpm--rebuilddb
- 1. 安装软件包

RPM 的主要用途之一是安装软件包。rpm 安装命令的一般语法是:

rpm -i [options] [packages]

这里的 options 可以是前面给出的常用选项之一,或者是下面列表中包括的安装选项之一, packages 是一个或多个 RPM 软件包文件的名字。RPM 能够使用的安装选项及其作用 如表 7.8 所示。

表 7.8	RPM 安装选项和其作用

安装选项	作用
-V	显示 RPM 正在执行的工作

安装选项	作用
-h 或-hash	当安装软件包时,显示 50 个散列符(#)
percent	显示文件从软件包中解压出来的百分比
test	演示软件包的安装过程,但实际上并不安装任何内容。它主要用来捕捉冲突
excludedocs	阻止文档文件的安装,例如手册页
includedocs	强制文档文件的安装,这是缺省的设置
nodeps	在安装软件包之前 , 不执行软件包相关性的检查
replacefiles	允许正在安装的软件包中的文件代替已安装的文件
replacepkgs	允许正在安装的软件包代替已安装的软件包
oldpackage	允许老版本的软件包代替以前安装的较新版本的软件包
force	强制安装一个软件包

当为 RPM 提供选项时 , 不管是什么运行方式 , 所有单字母的选项都可以被结合在一起 作为一个整体使用。例如 , 以下两条命令等价 :

#rpm -i -v -h rpmpackage.rpm

#rpm -ivh rpmpackage.rpm

但是,以--开始的所有选项必须单独提供。

2. 升级软件包

RPM 的升级方式提供了把现有软件包升级为新版本的一个非常便捷的途径。升级命令 与安装命令类似:

rpm -U [options] [packages]

options 可以是任何一个安装选项或任何一个通用的选项。

升级命令实际上是两种操作的组合,这两种操作是卸装和安装。在升级软件时,RPM 首先卸装老版本的软件包,然后再安装新版本的软件包。如果老版本的软件包不存在,那 么 RPM 只须对所请求的软件包进行安装。

与手工安装和卸装的方法相比,利用升级方法所具有的另外一个优点是,它能够自动 保存配置文件。由于这些原因,有些人宁愿使用升级方式而不是安装方式来进行所有软件 包的安装。

3. 卸装软件包

RPM 的卸装方式提供了一个彻底卸装软件包的方法 , 它能够干净地删除属于软件包的存放在各个不同位置的所有文件。

许多软件包在/etc、/usr 和/lib 中都安装有文件,因此删除软件包会令人感到十分麻烦, 而利用 RPM 却可以很容易地删除整个软件包。删除软件包的命令如下:

rpm -e [options] [packages]

options 将在后面介绍: package 是想要删除的软件包的名字。例如,如果要删除

DOSEMU 软件包,所使用的命令是:

rpm -e dosemu

这里为 package 指定的名字是软件包的名字,而不是用来安装软件包的文件的名字。 如果像下面这样卸装软件包:

rpm -e samba-2.0.7.21ssl.i386.rpm

便会产生如下的错误信息:

package samba-2.0.7.21ssl.i386.rpm is not installed

在卸装软件包时另外一个经常碰到的错误是软件包相关性的错误。当某个要被卸装的 软件包具有其他软件包所需的文件时,这种错误就会产生。如果该软件包被删除,可能导 致其他软件包不能正常运行。如果坚持要删除该软件包,可利用为 RPM 提供的--nodeps 选 项,它将使 RPM 忽略软件包相关性的错误。

还有一个十分有用的卸装选项是--test 选项,它能够使 RPM 演示删除软件包的操作,但却不实际删除任何内容。在通常情况下,卸装过程不产生任何输出。可以使用-vv 选项和--test 选项查看在卸装过程中所发生的事情。

4. 查询软件包

RPM 中的查询方式可用来确定软件包的各种属性。对软件包进行查询的基本语法是:

rpm -q [options] [packages]

这里的 options 是后面所列出的一个或多个查询选项。

RPM 查询方式在有些方面类似于卸装。它使用的也是软件包的名字 , 而不是安装软件 包的文件的名字。

如果想得到内核软件包的所有文件的清单,可以使用-1选项,例如:

rpm -ql kernel

这一命令将输出如下文件清单:

kernel-2.2.16-22

除可以获得文件清单外,还可通过使用-s 选项来确定它们的状态,这一选项将提供关于内核软件包所包含的文件的状态信息。

除了这些查询方式之外,通过使用 info 选项查看有关软件包的综合信息。例如,下面的命令将提供有关内核的如下信息:

rpm -qi kernel

查询结果如下:

Name : kernel

Relocations: (not relocateable)

Version	: 2.2.16	Vendor: Red Hat, Inc.
Release	: 22	Build Date: Tue 22 Aug 2000 08:58:22 PM GMT
Install da	te: Thu 19 Apr 20	01 12:01:27 PM GMT Build Host: porky.devel.redhat.com
Group	: System Environm	ent/Kernel Source RPM : kernel-2.2.16-22.src.rpm
Size	: 15211490	License: GPL
Packager	: Red Hat, Inc. <htt< td=""><td>p://bugzilla.redhat.com/bugzilla></td></htt<>	p://bugzilla.redhat.com/bugzilla>
Summary	: The Linux kernel	(the core of the Linux operating system).
Descripti	on :	

The kernel package contains the Linux kernel (vmlinuz), the core of yourRed Hat Linux operating system. The kernel handles the basic functions of the operating system: memory allocation, process allocation, device input and output, etc.

如表 7.9 所示是查询选项及其作用的列表。

表 7.9 RPM 查询选项和其作用

查询选项	作用
-1	列出软件包中的所有文件
-S	列出软件包中文件的状态
-d	列出软件包中所有的文档文件
-с	列出软件包中所有的配置文件
-I	列出软件包的综合信息

除-i 选项外,如果把这些选项与-v 选项一起使用,则文件将以 ls-1 格式列出。例如:

#rpm -qlv samba

它的输出结果为:

-rw-rr	1	root	root	152	Aug 14 2000	/etc/X11/applnk/System/swat.desktop
-rw-rr	1	root	root	196	Aug 14 2000	/etc/logrotate.d/samba
-rw-rr	1	root	root	125	Aug 14 2000	/etc/pam.d/samba
-rwxr-xr-x 1	root	root	1502	Aug	14 2000 /etc/r	c.d/init.d/smb
-rw-rr	1	root	root	97	Aug 14 2000	/etc/samba/smbusers
-rw-rr	1	root	root	357	Aug 14 2000	/etc/xinetd.d/swat

除前面介绍过的查询选项外, RPM 还能识别如下几个查询选项:

- · -a 列出系统安装的所有软件包
- · -ffile 列出拥有指定文件的软件包
- · -ppackage 列出由指定的软件包文件安装的软件包的名字

5. 验证软件包

利用 RPM 的验证方式对软件包进行验证是确定软件包是否出现问题的一个非常简单

的方法。在验证方式中 , RPM 将把所安装的软件包的信息同软件包在初始安装时存储在软件包数据库中的信息进行对比。

对软件包进行验证的基本语法如下:

#rpm -v packetname

如果软件包验证正确, RPM 将不会输出任何内容。如果 RPM 探测到所安装的软件包 和数据库的记录之间有一定的差别, 它将输出一个包含 8 个字符的字符串。在这个字符串 中,检验失败的项由一个单独字符来表示,检验通过的项由一个点(.)来表示。检验失败 时所使用的字符如表 7.10 所示。

字符	失败的检验
5	MD5 校验和
S	文件大小
L	符号链接
Т	文件修改的时间
D	设备
U	用户
G	用户组
М	文件的模式(文件权限和文件的类型)

表 7.10 检验失败的字符和含义

除此之外,利用查询选项-f 来验证包含某一特定文件的软件包也是可行的,它有助于 诊断某些程序的问题。

7.3.4 在 X Window 中管理RPM包

在 Red Hat Linux 7.0 中,除了可以使用命令行的方式管理 RPM 包之外,在 X Window 中还有图形化的工具可以更加直观、更方便地管理 RPM 包。

在 Red Hat Linux 7.0 中的图形化 RPM 管理工具叫 gnorpm。

选择 Program System | GnoRPM,或者在命令行中运行:

#gnorpm

出现的界面如图 7.11 所示。

如果初次使用gnorpm工具,可以单击gnorpm | Help | Introduction,将会出现如图7.12所示的帮助内容。可以从中找到适当的帮助信息。

可以通过一个类似文件管理器的界面来安装、卸装、查询和验证软件包。

例如,想要对 Apache 进行查询和验证。操作步骤如下:

(1)首先,选中图中的 System Enviroment | Daemons,右面的窗口中出现一些系统的服务程序。

(2) 在右面的窗口中选中 apache 的 RPM 包, 如图 7.13 所示。







图 7.12 gnorpm 的帮助界面



图 7.13 查询和验证 Apache 服务程序

(3) 然后,在任务栏中单击 Query 按钮。一个提供软件包详细资料的对话框被显示出 来。如图 7.14 所示。

Package Info			
apache			
	apache-	-1.3.12-25	
Size: 11 Build Host: po Distribution: Blu	79253 rky.devel.redhat.com ueSky	Install Date: Th Build Date: W Vendor: Re	nu Apr 19 11:53:49 GMT 2001 ed Aug 23 07:45:27 GMT 2000 ed Hat, Inc.
Group: Sy	stem Environment/Daemons	Packager:	Red Hat, Inc.
Apache is a pov Web server. Ap- Internet. DCSPath /etc/htt C /etc/htt	werful, full-featured, efficient ache is also the most popular pd/conf pd/conf/access.conf pd/conf/httpd.conf	and freely- availab Web server on th Jninstall	e Z

图 7.14 Apache 的查询结果

(4) 单击 Verify 按钮。验证结果如图 7.15 所示。

Verifying Pa	ickages		_ = ×
	apache-1.3	3.12-25	
	1 problems	found.	
Package	File	Problem	
apache-1.3.1	/etc/httpd/conf/httpd.coi	md5, file size, modification	time
	X Clo	ise	

图 7.15 验证 Apache 的结果

可以看到,有一个和验证数据库不符的条目,提示/etc/httpd/conf/httpd.conf文件被修改 过。这是正常的,因为确实修改了配置文件中的 ServerName 项,曾经去掉了这一行的注释, 并修改了服务器的名字。

7.4 备 份

在计算机操作中,有时会发生数据丢失。数据可能丢失的原因基本上可归结为以下 4 个:

· 硬件故障

虽然现在硬件越来越可靠,但仍然可能存在例外的情况。保存数据的硬件中,最关键的部分是硬盘,除了会发出微小的噪声外,它几乎是最理想的存储设备。虽然较新的硬盘

要比旧硬盘可靠一些,但它们也会发生故障,数据也会因此而丢失。

软件错误

在使用一些工具执行任务时,有可能这些工具对数据进行了破坏而不是正确地处理它。 有很多程序都是在未成熟之前就已经发布了。如果程序有错误,开发者再发布一个补丁程 序。在这种情况下,数据可能会蒙受损失。

· 人为因素

人们有时会操作失误,有的人甚至有目的地破坏数据。有很多数据损失都是因为他们 在错误的目录中执行了 rm -rf 命令。

 ・ 自然灾害

是一些偶然因素造成的数据丢失,如地震、龙卷风、火山、飓风等。

总之,谁也不能绝对保证数据不遭破坏或丢失。

数据非常重要。如果某些数据的生成时间很长,或者这些数据是唯一的,那么它们将 更为珍贵。因此应该特别小心,以免数据丢失。一旦丢失,要想重新建立它,得花许多时 间和精力,有时甚至不可能重建。要避免数据损失,解决的办法就是备份。

备份是保护数据的一种手段。通过保留若干个备份的方式,能够在数据受损的情况下 尽快恢复它。适当地备份是非常重要的。备份过程中最重要的是要保证备份能够发挥作用。

7.4.1 优良备份的品质

对于备份,最理想的情况是同时具有绝对可靠、随时可以使用、操作简单和速度快等 品质。在实际情况中,这些特点却是同时具备的,往往要根据情况有所取舍。

- · 在这些品质中,最重要的是备份的可靠性。一种可靠的备份介质应该能够保存几 年时间。
- 备份的速度是否重要,要由系统来确定。如果系统长时间不被使用并且备份能够 被自动执行,那么速度就不成问题;而对于恢复可能就是一个问题,因为恢复数 据所花费的时间与恢复数据变得同样重要。
- 可用性是一个必不可少的品质。再好的备份,如果在需要使用它们时却拿不到手, 那么它们也就毫无意义。用来执行灾难性恢复的备份对于恢复一个偶然删除的文件显然不合适。一个优良的备份和恢复方案既应该包括一套放在本地用于日常恢复的备份,也应该包括一套远离现场用于灾难性恢复的备份。
- 即便备份的速度很快,随时可以拿来使用并且可靠性也很高,但如果它们的易用
 性差,仍然不会有太大的用处。用来备份和恢复的工具必须易于使用,特别是对
 于恢复更是如此。在紧急情况中,可能一时难以找到执行备份和恢复的专业人员,
 此时便不得不让非专业人员执行恢复工作,因此,备份应该易于操作。
- 7.4.2 备份或者恢复操作之前需要考虑的几个因素

进行备份操作的最佳时机就是安装好 Linux 操作系统并确信所有的设备都能够正常工作的时候。在对系统进行备份或者恢复操作之前,有几个因素必须先考虑清楚。

应该知道一个备份文件和一个档案文件之间是有区别的。备份是定期进行的操作,用

来保存重要的文档、文件或者整个系统。而对文件归档则是为了长期保存重要的文档、文件或者整个系统而进行的操作。这就意味着必须首先设计出一个进行备份操作的策略并考虑下面几个问题:

- 是否需要使用一个正规的备份操作策略?
- · 是否需要每次都对整个系统进行备份?
- · 是需要备份选定的文件,还是备份整个一个子目录?
- · 每隔多长时间需要进行一次备份?
- 需要把归档的拷贝保存多长时间?
- 是否需要关于备份操作过程的各种报告或者统计数据?
- 应该使用哪一种介质?
- 应该采用哪一种软件格式存放备份?
- · 是否需要使用一些专业的软件工具,是否可以手动进行备份操作?
- · 是否需要使用压缩技术、直接拷贝或者加密技术?

通过检查使用的 Linux 操作系统方式,可以回答出上面这些问题中的一部分。如果在 Linux 操作系统中只是进行文字处理或者运行电子表程序,可能只对某些特定的文件或者子 目录进行备份就已经足够了;如果通过 Linux 操作系统来学习编写计算机程序,那么就需 要保存程序的原始拷贝和这些程序的不同的改进版本;如果在这个计算机系统上还有其用 户,那么可能不仅需要保存他们的子目录与文件;还要保存/etc/passwd 文件、保存文件系 统中花费很长的时间才能建立的部分;甚至需要保存整个系统。

这样,当出现硬盘崩溃或者操作失误等事件的时候,就能够迅速地对系统进行恢复。

7.4.3 备份什么

一般都希望尽可能地保留备份。但是,可以轻松重新安装的软件一般不必备份,即使 备份配置文件非常重要,只需重新配置即可。另外,/proc 文件系统,特别是/proc/kcore 文 件最没有必要备份。

必须决定哪些是最重要的。很显然,需要备份的是用户文件(/home)和系统配置文件 (/etc),还有散布在文件系统各处的其他文件。

7.4.4 压缩备份

备份会占用大量的空间。为了减少备份空间,所以要对备份进行压缩。常见的压缩方式有许多。有的程序内置压缩支持,如 GNU tar 的-z 选项,它在把备份写入备份媒体之前,通过 gzip 压缩程序,把整个备份压缩在一起。

但是,压缩备份通常会带来许多不便。由于压缩的特性,如果其中有一点错误,其他的压缩数据都将不能使用。有的备份程序有内置的纠错功能,但面对大量的错误,也是无法解决的。这意味着,如果备份是像 GNU tar 那样压缩的,也就是说,把整个备份压缩成一个整体,如果其中有一个错误,将导致其他所有备份数据的丢失。由于备份必须可靠,因此,这种压缩方式不是一个好办法。

另一种方法是对每个文件进行单独压缩。这意味着,如果其中一个文件丢失,其他的

文件却不会受损。总之,丢失的文件还是受损,所以,这种方法只比没有使用压缩好一点。

7.4.5 备份媒体

选择备份媒体比较重要。在选择备份媒体时,需要考虑价格、可靠性、速度、可得性和可用性这几个因素。

- 价格 是比较重要的因素,因为可能有许多数据需要多次备份,选择相对便宜的 存储媒体是必须的。
- 可靠性 是相当重要的因素。因为一个被破坏的备份无异于没有备份。在理想情况下,备份媒体能够保存数据的时间长达数年。对备份媒体而言,使用媒体的方式也会对其可靠性产生一定的影响。一般来说,硬盘是非常可靠的,但如果备份媒体和正在备份的磁盘处于同一台机器上,备份媒体就不可靠了。
- · 速度通常情况下,如果备份时不需要交互操作,速度就不是十分重要。只要它不 需要人监管,几个小时的时间是可以接受的。
- · 可得性 是必须考虑的因素。例如,用户没有光盘刻录机,当然不能将自己的数 据备份在光盘上。
- · 可用性 在考虑备份频率时,可用性最为重要。制作备份的方式越简单越好。备 份媒体绝对不应该不好用。

目前,常用的备份媒体是软盘、磁带和硬盘、只读光盘和可读写光盘,如表 7.11 所示 是在可靠性、速度、可用性和易用性方面进行的评价。

媒体	可靠性	速度	可用性	易用性
软盘	好	慢	高	少量数据时好,大量数据时差
只读光盘	好	快	高	可以做存档
可读写光盘	好	快	中等	可读写介质 , 对中等规模的系统是很经济的
磁带	好	中等	高	取决于磁带的容量 , 易用性可以很好
活动硬盘	好	快	高	相对较贵,可以提供 2GB 或更大的存储空间
硬盘	好	快	高	较好

表 7.11 备份媒体比较

7.4.6 备份策略

最简单的备份策略就是把系统上的所有文件拷贝到磁带,这称为完全备份。完全备份 的缺点是耗费的时间比较多。如果只须恢复单个文件,使用它就得不偿失。一个良好的备 份和恢复方案,应该清楚什么时候需要完全备份,什么时候增量备份会更好。

如果 Red Hat Linux 系统用于商用,就应该有一个备份策略。使用一个正式的计划来定 期存储关键的信息,如客户帐目和工作计划,以防止财产的损失。在设计了备份计划之后, 应坚持使用。

增量备份的执行一般比较频繁。利用增量备份,只有从上次备份以后被修改过的文件 才被备份,因此每一次增量备份都是建立在前一次增量备份的基础之上。 Linux 利用备份级别来区分不同的备份种类。完全备份被指定为0级备份。其他级别表示的是自前一级别以来被修改的文件。例如,在星期天晚上可能会执行一次0级备份(完全备份);到了星期一晚上将执行1级备份,这次备份将针对自0级备份以来被修改过的所有文件进行备份;到了星期二晚上将执行2级备份,这次备份将针对自1级备份以来被修改过的所有文件进行备份,依此类推。下面将介绍两种基本的备份策略。

第一个策略是:

- 星期天 0级备份
- 星期一1级备份
- 星期二1级备份
- 星期三 1 级备份
- 星期四 1级备份
- 星期五 1级备份
- 星期六1级备份

这种备份方案的优点是:它只需要两套磁带。要想对系统执行一次完整的恢复,只须 通过恢复0级备份和前一个晚上的增量备份就可以实现。

它的缺点是:需要备份的内容会逐渐增长到在一周内被修改的文件。因此数量会很大, 可能会需要使用另外一盒磁带来执行备份。

第二个策略是:

- 星期天 0级备份
- 星期一1级备份
- 星期二 2级备份
- 星期三 3 级备份
- 星期四 4 级备份
- 星期五 5级备份
- 星期六 6级备份

这种备份方案的优点是:每次备份都比较迅速,并且备份的内容比较少从而易于管理。 它的缺点是:需要有7套磁带要对系统执行一次完整的恢复,必须使用所有7套磁带。 在决定使用何种备份方案时,需要了解系统的使用情况。经常改动的文件应该比改动 少的文件的备份更频繁,而有些目录根本不需要备份。

7.4.7 备份工具

用于制作备份的工具有许多。用于备份的 Linux 工具有 tar、cpio 和 dump。另外,还 有许多第三方工具包,包括自由软件和商业软件。备份媒体的选择直接对备份工具的选择 产生影响。

从备份角度来看, tar 和 cpio 极为相似。两者都能把数据保存在磁带上,并从磁带恢复数据;两者均适用于任何一种媒体,因为内核设备负责管理低级控制设备和越来越像用户 级程序的设备。tar 和 cpio 的 Linux 版本可以胜任所有文件的备份。

dump 不同于前两者的地方在于:它直接读取文件系统,而不是通过文件系统读取文件,

它也是专门针对备份而设计的。tar 和 cpio 其实是用于对文件进行归档的,但同样可用于备份。

直接读取文件系统有诸多好处。例如,可在不影响文件时间戳的情况下备份文件;如 果利用 tar 和 cpio,就必须先以只读方式装入文件系统;此外,如果所有文件都需要备份, 直接读取文件系统更为有效,因为这样可减少磁头的运动。直接读取文件系统的缺点在于 备份程序只能专用于:一种文件系统类型;Linux dump 只能理解 ext2 类型的文件系统。

dump 还支持备份级别。tar 和 cpio 必须借助其他工具才能实施备份级别。

下面将分别介绍各工具。

1. tar 备份工具

tar 命令会生成一个档案文件。可以把这个档案文件写到硬盘驱动器上、一台磁带机上 或者几乎其他任何 Linux 操作系统中的设备上。

如果想在 tar 命令中使用磁带驱动器 , 就必须先在/dev 子目录中找到这个磁带设备。磁带设备有许多不同的类型 , 这里使用软盘磁带驱动器/dev/ftape 做为示例。

可以通过指定/dev/ftape 设备的方法在磁带上写一个 tar 类型的档案文件,如下所示:

#cd /

#tar cf /dev/ftape /home

这个命令建立了一个包含/home 目录下所有文件和子目录的磁带档案文件。如果想恢 复磁带档案文件,必须使用 tar 命令的 x 释放参数,如下所示:

#cd /

#tar xf /dev/ftape

· 列出一个 tar 档案文件的内容清单。

在打开某个 tar 档案文件之前,如果需要查看其中都有些什么,请使用下面的命令查看 文件名清单:

#tar -tf filename.tar

或者,如果想看到一个"ls-l"命令结果的长格式输出,请使用下面的命令:

tar -tvf filename.tar

· 对文件进行解 tar 归档操作。

如果想对文件进行解 tar 归档操作,需要使用的基本命令参数是字母 x (解包)。对文件进行解 tar 归档操作最常用的参数组合是 xvf,如下所示:

#tar -xvf filename.tar

参数 v 表示显示信息,参数 f 告诉 tar 命令后面跟着一个文件名。

在什么位置打开一个 tar 档案文件取决于它最初是如何被建立的。查看其中的内容可以 帮助决定怎样完成这一工作。这个操作会让用户看到在 tar 档案文件中保存的目录结构。这 样就可以相应地选择把这个文件放到什么位置。

如果还是没有把握,可以先在一个临时目录中对文件进行解档案操作,然后看看它是

按照什么结构打开的。接下来就可以在正确的位置再一次对文件进行解档案操作。

· 从 tar 档案文件中只解包 README 文件。

从 tar 档案文件中只解包 README 文件通常是很有用的 , 因为可以看到应该在什么位 置对文件进行解档案操作 , 而不必做两次。请按照下面的步骤进行 :

(1) 进入保存 tar 文件的目录。

(2) 输入 tar -tvf archive.tar 命令。

(3) 记下档案文件中 README 文件的格式和位置。

(4) 完整输入下面的命令,用正确的文件名替换下面命令中的 dir/README 字段:

#tar xf achive.tar dir/README > README

例如,有一个名为 program.tar 的档案文件,并且在它的目录结构中看到列出了 home/mxn/Read.me 的内容,那么就应该输入下面的命令:

#tar xf program.tar home/mxn/Read.me > README

注意:在进行这个操作时也可以不使用重定向,但是这样就会按照档案文件中包 含的目录结构建立这个文件。因此,如果这个例子中在命令的结尾处没有使用重 定向,这个 Read.me 文件就会被解包到 home/mxn 目录中。

(5) 仔细阅读 README 文件,看看它要求在什么地方打开这个 tar 档案文件。

· 建立一个 tar 档案文件。

请按照下面的步骤建立一个 tar 档案文件:

(1) 找到想对它们进行归档操作的目录或文件。

(2)决定是否允许档案文件被恢复到一个相对路径或者决定路径。

(3)把路径切换到想对之进行归档操作的目录或者文件的上一级父目录中。例如,如 果想对/var/spool/mail 目录进行归档操作,就切换到/var/spool 目录中。

(4) 输入 tar cvf archiev.tardirectory 命令,建立档案文件。命令中的 archive.tar 是新档 案文件的名字, directory 是对之进行归档操作的目录的名称。

例如,如果打算把/var/spool/mail 目录归档到 mail-archive.tar 文件中,需要从/var/spool 目录中输入 tar cvf mail-archive.tarmail 命令。

· 向现有 tar 档案文件中添加文件。

如果打算向现有 tar 档案文件中添加文件,请按照建立一个新 tar 文件的步骤操作,但 是要用参数 rf 代替 cvf,如下所示:

#tar rf mail-archive.tar mail-to-add

2. cpio 备份工具

cpio 命令(拷贝输入输出命令)的使用方法与 tar 命令大致相同,但是也有几个区别, 特别是在建立和释放档案文件操作所使用的命令行参数方面。

例如,如果想建立一个 cpio 文档,必须使用-o 建档参数。cpio 命令在建立档案文件的

时候还需要有一个文件名清单,而且路径名必须完整。

例如,在一个名为 x11 的子目录中包含有一些想进行归档的文件,可以把 find 命令的 输出与 cpio 命令组合在一起来生成这个文档,如下所示:

#find -name xll /*| cpio -o -v

这个命令行显示 cpio 命令获得了 x11 子目录中的文件名。使用-o 参数生成了档案文件, 并使用-v 参数在对文件进行归档的同时显示了它们的文件名。最后得到的档案文件的名字 是 x11.cpio, 它是使用了重定向操作符建立的。

如果想恢复一个 cpio 文档,需要使用 cpio 命令的-i 释放命令行参数,如下所示:

#cpio -i < x11.cpio 1082 blocks

从上面的命令行可以看出,可以把小于号(<)重定向操作符与 cpio 命令的-i 释放参数 一起使用来释放一个 cpio 文档, cpio 程序会重新生成原来的子目录结构。当操作完成的时候, cpio 命令会显示一个数字,这个数字以 512 字节为单位表示它往硬盘上写了多少个字 符块的内容。

还可以把 find 命令和 cpio 命令组合使用来建立任何一个子目录的档案文件,甚至还可 以对系统中不同子目录中的文件也进行归档。如果想对子目录中所有属于用户并且在当天 建立的或者是在当天修改过的文件进行备份,可以使用 find 命令的-user 和-mtime 命令行参数,如下所示:

#find /home/duwei -user duwei -mtime -1 -print
#cpio -o > today. cpio

这个命令行将建立一个名为 today.cpio 的 cpio 档案文件,其中包括所有在过去 24 小时 内修改过的文件。

3. taper 备份

taper 命令脚本程序是一个用来对选定的文件或者子目录建立压缩或者非压缩档案文件 的备份与恢复的程序。这个程序提供了良好的建立磁带档案文件并对之进行管理的界面。 它也可以使用其他类型的存储介质进行备份。

taper 命令使用起来很容易。使用-T 介质参数就可以指定它使用的存储介质的类型。 taper 命令支持如表 7.12 所示的几种选项。

选项	意义
ftape	包括在系统中的软盘驱动器磁带驱动程序,适用于连接到软盘接口的磁带机
zftape	一个比较新的软盘驱动器磁带驱动程序,它可以处理附加的磁带格式
scsi	支持使用 SCSI 接口的磁带机驱动程序
ide	支持 ide 接口磁带机驱动程序
removable	支持软盘或者活动硬盘的驱动程序

表 7.12 taper 命令支持的选项

如果没有磁带机但是想试试使用 taper 程序并把一两个子目录备份到软盘上,请使用-T 命令行参数和 removable 选项,如下所示:

#taper -T removable

这个命令启动 taper 程序运行。taper 命令将使用缺省的活动设备/dev/fd0,也就是软盘 驱动器。

7.5 习题

1. 使用 vi 编辑一个文本文件。练习光标的移动、文字的插入、删除、拷贝、粘贴等 命令。

2. 查看 linuxconf 可以配置的内容。选择比较熟悉的内容进行配置。

3. 在 Red Hat Linux 7.0 的第2张光盘上选择一个 RPM 包进行安装练习。

第8章 Linux 网络基础

本章主要介绍 Linux 网络基础和 TCP/IP 协议网络。内容包括 TCP/IP 协议网络简介、 TCP/IP 协议和 TCP/IP 网络工作原理、IP 路由以及域名系统简介等。

学习了本章之后,应该对 TCP/IP 协议网络有了一定的了解。本章内容主要为后续的 Linux 网络配置和服务器安装配置作准备,其中的 IP 路由,域名系统等知识将在以后的章 节中用到,应该认真掌握。

8.1 TCP/IP协议网络简介

网络是一系列计算机的集合。在网络中,不同的计算机之间能够相互通信。通常,这 种通信要依赖某些专用计算机或服务器,它们处在参与通信的两台计算机之间,完成对数 据的转发或者中继功能。计算机之间也可以用协议实现通信,但是,这种协议和通常提到 的"协议"不同,它们是一些高度规范化的代码,严格约束着通信双方的行为。简单地说, 计算机网络中采用的协议就是一系列非常严格的规则,规定两个或更多的计算机之间如何 交换信息。

在局域网中,数量不多的计算机位于同一幢建筑物中,甚至位于同一层楼或者同一个 房间里。这些计算机相互连接在一起,营造一个能够协同工作的环境。在这种情况下,不 同的计算机间经常需要共享文件,或者运行分布式的应用程序。这时,待传输数据需要分 割为较小的数据包,转发至目标计算机。抵达目标计算机后,再重新"组装"回原来的样 子。这种类型的网络叫作"包交换"网络。它最基本的应用便是通过网络运行分布式应用 程序。

基于 TCP/IP 协议的网络就是典型的包交换网络。目前,大多数系统(包括非 Linux 站 点)选择的协议都是 TCP/IP 协议。

为便于下面介绍 TCP/IP 协议,这里以一个虚设的大学 univ 为例。在这所大学中,各 系都运行自己的局域网(LAN)。一些系和别的系共享一个网络,另一些系则同时运行着几 个网络。所有网络相互间都是连接的。整个校园网通过一条高速链路,接入互联网 Internet。

假定一台计算机属于数学系的 LAN, 名字叫做 math1。math1 可以通过一种由 TCP/IP 协议提供的实时交互式应用远程登录来访问物理系中名为 phy1 的一个计算机,具体命令如下:

#rlogin phy1.physics
Welcome to the physics Department at univ
(ttyq2) login:

在出现 login 的命令提示行中,输入用户的登录名以及正确的密码,就可以获得 phy1

计算机的一个使用界面。可以在其中键入任何命令,就像坐在 phy1 计算机的控制台前一样。 退出这个界面后,就返回到用户自己计算机的命令提示行。

登录进 phy1 的时候,可能需要运行一个以 X11 为基础的应用程序。将该程序的窗口 显示在自己计算机的屏幕上,这时必须设置 DISPLAY 环境变量,如下所示:

#export DISPLAY=math1.maths:0.0

如果现在启动这个应用程序,它就会同本机的 X 服务器联系,而不是同 phy1 的服务器联系,并将所有窗口都显示在本机的屏幕上。要做到这一点,要求 TCP/IP 协议在 phy1 和 math1 两部计算机之间来回传送 X11 数据包,营造一个"透明"的网络。

在 TCP/IP 协议网络中,另一个非常重要的应用是 NFS (Network File System)。NFS 主要用来"复制"其他计算机的目录结构,使其他计算机的目录结构看起来就像本机的文件系统。例如,所有用户的主目录(home directory)都可存放在一台中心服务器中,局域 网内的所有计算机都复制它的目录。这样,可登录网内的任何一台计算机,并发现自己处在一模一样的主目录中。

以上是使用 Linux 通过 TCP/IP 协议网络能够做的一些事情。当然, TCP/IP 协议网络能做到的事情远不止这些。

8.2 TCP/IP协议及网络工作原理

IP 路由是 TCP/IP 网络的重要内容,下面将介绍涉及 IP 路由的一些基本概念。

8.2.1 以太网

目前,在局域网中广泛采用的形式叫做 Ethernet, 即"以太网"。

以太网工作起来就像一个总线系统。一个计算机可将单位长度多达 1500 字节的数据包 发给同一以太网内的另一台计算机。每个计算机都分配有一个 6 字节的地址,这种地址需 要固化到网卡内部。通常,用一个十六进制数字的序列来表达这种地址,每对数字之间用 冒号分隔,例如,aa:bb:cc:dd:ee:ff。

由一台计算机发出的数据包可被网内其他所有计算机收到。但是,只有目标计算机才 能实际接收并处理它。假如两台计算机同时试图发送数据,便会发生冲突。解决这种冲突 的办法便是两台计算机都取消这一次发送,各自等待随机的一小段时间,再进行发送数据 包的尝试。

8.2.2 IP协议

当然,网络连接并不仅仅限于一个以太网。在最理想的情况下,希望能任意使用一个 网络,无论它运行于什么硬件形式之上,也无论它由多少个子网组成。例如,在像 univ 校 园网这样的网络中,通常包含了数量众多的独立以太网,它们需要以某种形式相互连接到 一起。在 univ 大学,假设数学系运行着两个以太网。其中一个网络连接的是速度较快的计 算机,由教授和研究生使用;另一个网络连接的计算机速度较慢,由学生平时上机使用。 两个网络都同时接入校园干线。 这种连接是由一个名为网关的专用计算机控制的。网关的作用是在两个以太网和校园 干线之间,通过复制数据包的方式,对数据包的接收和发送进行处理。例如,如果数学系 的计算机想访问物理系的局域网上的 phy1 计算机,但连网软件不能直接向 phy1 发送数据 包,因为它们不在同一个以太网上。因此必须依靠数学系的网关 gate1 来转发数据包。然 后,网关 gate1 再把数据包转发到物理系的同级网关 gate2,由 gate2 把数据包转发到目标 计算机。

直接将数据导向远程计算机的方法称作"路由"。数据包常常被称作"数据报"。为使 一切简单化,数据报交换由一个独立的与硬件无关的协议进行管理。这个协议就是著名的 IP 协议(本书介绍的是 IPv4。以后如果不加以特别说明,IP 协议均指 IPv4)。

IP 协议的主要好处是:从物理上把不同的网络变成了同一个网络。这就是连网技术。 当然,IP 协议同时还需要一个与硬件无关的定址方案,也就是每台计算机对应一个独一无 二的 32 位数,这个数便是该计算机的 IP 地址。IP 地址通常用"点分四段"的格式来表示, 每一段以十进制表示其 4 个字节中的一个字节,段间用句点分隔。例如,phy1 可能有 "0x954C0C04"这样的一个 IP 地址,它对应的是 159.226.12.4,前 3 个字节由 InterNIC 注 册服务中心分配,标识计算机接入的网络,剩下的字节用来标识该计算机本身。

大家还会注意到,现在有3种地址类型:其一是计算机名,例如 phy1;其二是 IP 地址; 最后是硬件地址,例如6字节的以太网地址。所有这些都必须彼此相符,只有这样,在输入 rlogin phy1时,联网软件才能给出 phy1的 IP 地址。而且,在 IP 向物理系投递数据时, 必须找出其 IP 地址对应的以太网地址。

把计算机名映射成 IP 地址的寻址过程叫做计算机名解析,把 IP 地址映射成硬件地址的 寻址过程叫做地址解析。

8.2.3 TCP协议

目前看来,从一台计算机向另一台计算机发送数据包只是任务之一。如果登录到 phy1,还打算在 math1 上的登录协议命令(rlogin)进程和 phy1 上的进程之间建立一条可靠的连接,那么,发送端必须把准备收发的信息分成若干个数据包,再由接收端把多个数据包重新组合成一个字符流。这样做会增加许多工作量,而且,IP 协议不能保证传输的正确性。如果网络出现资源不够用的现象,IP 协议就会将数据包丢弃,这时,检查数据的完整性,数据是否完全以及另行转发就是通信计算机的责任。

TCP 协议(传输控制协议)可以解决这个问题。TCP 在 IP 之上建立了一个可靠的服务。 其本质特征是利用 IP 建立本地计算机和远程计算机上的两个进程之间的一条连接。从本质 上来说, TCP 连接的原理其实就像一条 " 双向 " 管道,两个进程可以边读取边写入。

TCP 利用连接涉及两台计算机的 IP 地址和各计算机上的端口号来标识连接的端点。端 口可被视为网络连接的附着点。如果把 TCP 连接想像成"打电话",那么 IP 地址就是区号, 而端口号就是本地代号。

在 rlogin 这个例子中,客户机应用程序 rlogin 打开 math1 上的一个端口,并将它连接 到 phy1 上的 513 端口 rlogin 服务器将被告知对这个端口进行监听。这样便建立了一条 TCP 连接。这样,在自己计算机上输入的任何命令都会通过 TCP 数据流得以传递,并被当作标 准输入传给远端的计算机。 8.2.4 UDP协议

当然,在 TCP/IP 协议中,TCP 并不是唯一的用户协议。NFS 这样的应用程序使用的 是 TCP 协议的兄弟协议——UDP(用户数据报协议)协议。和 TCP 协议一样,UDP 协议 也允许应用程序和远程计算机某个特定端口上的服务取得联系,但它不能建立连接,而是 向目标计算机发送独立的数据包。相对于 TCP,UDP 要快得多。但是 UDP 协议不会处理 数据包的丢失或中断。如果出现这种情况,只能由应用程序进行处理。

8.2.5 端口

端口可以看作是网络连接的附着点。如果一个应用程序想提供一项特定的服务,它就 会把自己附着在一个端口上,在该端口上监听客户机。想利用该项服务的客户机便在其本 地计算机上分配一个端口,并连接到远程计算机的服务器端口。

端口有一个重要特性:客户机和服务器之间的连接一旦建立,服务器的另一个副本可 能就会附着在服务器端口,监听更多的客户机。也就是说,允许若干个并发登录利用同一 个端口登录到同一个计算机。TCP 能够把这些连接区分出来,因为它们来自不同的端口或 计算机。

例如,如果两次均从 math1 计算机登录进入 phy1,第一个 rlogin 客户机就会采用 1023 这个本地端口,第二个则采用 1022 的本地端口。然而,两者均是连接到 phy1 上同一个 513 端口上。这个例子说明了端口可用做集合点,客户机通过它连接到另一个特定的端口以获 得特定的服务。为了让客户机知道正确的端口号,两个系统必须在端口号的分配问题上达 成一致。对用得较广的服务,其端口号就必须进行集中管理。这是由 IETF(因特网工程任 务组)来完成的。IETF 定期发布一个标题为"已分配号"的 RFC。该 RFC 对分配给众所 周知的服务器的端口号进行说明。Linux 采用的是一个文件,名为/etc/services。该文件把服 务名映射为端口号。

注意:尽管 TCP 和 UDP 连接和端口有很大关系,但其端口号之间不会有冲突, 就是说,在前面的例子中,TCP 端口 513 有别于 UDP 端口 513。

8.3 IP 网络

IP 网络其实与邮件投递系统类似。当写信给某人时,会在信封上写明收信人的确切地 址,指定国家、地区、省份、城市以及详细到几栋几号。邮政服务将把信送达目的地:先 送到指定国家,再由这个国家的邮政服务将其分发到相应的省或地区。邮政服务的层次结 构非常清楚:不管在哪里寄信,本地的邮政主管都知道信件投递路径,并将进行信件转发, 并不注重信件在目标国内的投递方式。

与之相同, Internet 由无数个独立的网络系统构成。每个系统都会在各自的成员计算机 之间执行路由,所以投递数据包的任务实际上就是找出通往目标计算机所在特定网络的路 径。也就是说,只要数据包被传到这个特定网络中的任何一台计算机,以后怎样到达最终 的目标计算机,则由这个特定网络自行负责。 8.3.1 子网

在前面介绍 IP 地址的内容中,把 IP 地址分为网络部分和计算机部分。默认情况下,IP 网络编号相同的计算机处于同一个网络内。也就是说,一个 IP 网络编号代表一个独立系统。 但在一个独立系统中,也往往包含不止一个 IP 网络,IP 网络还可以进一步分成若干个子网。

子网主要负责从自己所处的 IP 网络,把数据包投到特定范围内的 IP 地址。A、B 或 C 类地址的识别,是通过 IP 地址的网络部分来完成的。但是现在的网络编号被扩展到包含计 算机部分的位数。这些被视作子网编号的位数是子网掩码所赋予的。它同样是一个 32 位的 编号,用于为 IP 地址的网络编号指定位掩码。

univ 大学校园网就是一个很好的例子。它采用的是 B 类地址,其网络编号是 159.226.0.0,所以其网络掩码是 255.255.0.0。

从其内部结构来说, univ 大学校园网由若干个小型网组成。例如各个系的局域网。所 以,这个 IP 网络被分为 254 个子网, IP 地址在 159.226.1.0~159.226.254.0 之间。例如,理 论物理系分配的 IP 地址是 159.226.12.0,校园主干网的地址是 159.226.1.0。这些子网共享 同一个 IP 编号,其中的第 3 个字节是用来区分它们的。所以,它们的子网掩码将是 255.255.255.0。

8.3.2 网关

网关是同时连接两个或两个以上物理网络的计算机,它用于执行网络间的包交换。不 同的子网经常采用不同的网络硬件。物理网络上的计算机,如以太网内的计算机只能与本 网络内的计算机直接交谈。要对其他的计算机进行访问,只有通过网关才能进行。

对于 IP 网络,不同的物理网络只能属于不同的 IP 网络。例如,网络编号 159.226.4.0 是为数学系局域网上的计算机保留的。在向 phy1 发送一个数据报时,math1 计算机上的网 络软件立即就能知道该数据报来自 159.226.12.4 这个 IP 地址,而且其目标计算机处于另一 个物理网络中。因此,这个数据报只能通过一个网关(默认设置是 gate1)抵达目的地。

gate1 本身连接了两个子网:数学系的局域网和校园主干网。它分别通过两个不同的接口(eth0、eth1)访问这两个子网。这个网关需要分配两个 IP 地址,一个根据 159.226.4.0子网进行分配,另一个根据 159.226.1.0子网进行分配。这两个地址和其相应的网络掩码都 绑在接口上。因此,接口和 gate1 地址之间的对应关系就会像下面列出的这样:

接口	地址	网络掩码
eth0	159.226.4.1	255.255.255.0
eth1	159.226.1.4	255.255.255.0

8.3.3 路由表

下面将介绍在把数据报投到远程网络时,IP 协议是如何选择网关的。

math1 网关的工作流程在前面介绍过。它收到发向 phy1 的数据包后,便对其目标地址 进行检查,并发现其目标计算机不在本地网络内。所以,math1 网关把该数据包发给默认 网关 gate1。gate1 识别出 phy1 不在它直接连接的任何一个网络内,它把该数据包发给网关 gate2,这是通往物理系局域网的网关。这样,gate1 就需要更多的信息把目标网络和一个适 当的网关关联起来。它采用的路由选择信息 IP 实际上就是一个表,把网络和准备抵达的网 关链接起来。

一般的,路由表中有一个默认路由,它是一个与 0.0.0.0 网络关联在一起的网关。发向 未知网络的所有包都会通过这个默认路由得以发送。

gate1 网关的路由表如表 8.1 所示。

网络	网关	接口
159.226.1.0	-	eth1
159.226.2.0	159.226.1.2	eth1
159.226.3.0	159.226.1.3	eth1
159.226.4.0	-	eth0
159.226.5.0	159.226.1.5	eth1
:	÷	÷
0.0.0.0	159.226.1.2	eth1

表 8.1 gate1 的路由信息表

对于直接与 gate1 连接的网络,通向它的路由不需要网关,所以显示的网关条目是"-"。

路由表的建立方式较多。对于小型局域网,在启动时,利用 route 命令,手工构建路由 表并把它们投入 IP 网络通常是最有效的。对于大型的网络,需要根据路由 daemon,在运 行时构建路由表并适时进行调整。对于后者,路由信息运行于网络内的中心计算机上,并 在其成员网络间实行路由信息交换,从而计算最佳路由。

根据网络的不同大小,可能会用不同的路由协议。对于独立系统(例如 univ 校园网) 内的路由选择,将采用内部路由协议如 RIP(路由信息协议)。对独立系统间的路由,则必 须使用如 EGP(外部网关协议)或 BGP(边界网关协议)之类的外部路由协议。

8.3.4 Internet控制消息协议

IP 有一个伴侣协议 ICMP (Internet 控制消息协议), 它主要用于错误控制。假设 math1 打算登录到 phy1 的 12345 端口, 但这个端口没有监听进程。所以,发向这个端口的第一个 TCP 包就会抵达 phy1,网络层将认出这个包并立即向 math1 返回一条 ICMP 消息,指出"不 能抵达指定端口"。

8.4 域 名 系 统

TCP/IP 协议的连网协议中的地址使用的是 32 位的编号,但是用数字编号的计算机地 址使用起来很不方便,因此,计算机一般都有一个计算机名。然后,由特定的应用程序负 责找出和这个计算机名对应的 IP 地址。这个过程就叫作"计算机名解析"。

现在,在小型网络上维护计算机名和 IP 地址的对应表很简单。维护信息通常保存在名为/etc/hosts 的文件中。在增添或删除计算机名或重新分配 IP 地址时,只须根据实际情况, 更新这个文件即可。

对于稍大型的网络,使用/etc/hosts 文件维护地址是一个不可取办法,必须采用域名系

统(DNS)来进行维护。

8.4.1 输入DNS

DNS 采用域分层结构来管理计算机名。一个域就是若干个站点的集合,这些站点可以 组成一个网络,或者它们都属于某个特定的组织,或者它们的地理位置相当接近。例如, 所有的大学可以归为一个.edu 域,而各个大学或学院又分为一个单独的子域。如果为 univ 大学指定的是 univ.edu 域,那么,为数学系局域网分配的域就是 maths.univ.edu。数学系局 域网内的各台计算机分到的域就是再在前面的域名前加上计算机名。例如 math1 的域就是 math1.maths.univ.edu。

根据在域名分层结构中的位置,域可以称作顶级域、二级域或三级域。下面是一些常 见的顶级域:

.edu 教育部门,例如大学

- .com 商业组织,公司
- .org 非商业组织
- .net 网络上的网关和其他管理性计算机
- .mil 美国军方
- .gov 美国政府部门

从技术上讲,前4个域属于 Internet 的美国部分。但这些域中同样包含有非美国站点, 特别是.net 域中。此类非美国站点比比皆是。但是,.mil 和.gov 是美国专用域。

除美国外的其他国家都可以使用 ISO-3166 中定义的两个字母作为其顶级域。例如,中 国使用的是.cn 域;法国使用的是.fr 域;德国使用的是.de 域;澳大利亚使用的则是.au 域。 在顶级域下,各个国家可按照自己的方式组织计算机名。

这些国家域并不能代表这些域下面的计算机位于该国,它只是表示这台计算机在这个 国家的 NIC (国家 Internet 中心)已经注册。

对域名分层结构内的域名空间进行组织,能很好地解决域名唯一性的问题。利用 DNS, 计算机名在自己所处的域内必须是独一无二的,这样能保证它的名字有别于全球各地的计 算机。此外,完整资格域名也非常容易记,正因为有了它们,才可将一个大型的域分成若 干个子域。

DNS 还可以选择子域内的授权者作为管理员。例如, univ 计算中心的维护人员为各系 创建一个子域,然后把子域的控制权移交给该网络的管理员。然后,该网络的管理员就可 以自由地使用自己喜欢的计算机名,并在不与网络外部发生冲突的前提下,按照自己的方 式为计算机分配 IP 地址。

8.4.2 利用DNS进行名字查找

DNS 是一个巨大的分布式数据库。它是通过一个所谓的"域名服务器"来实现的。这 些域名服务器将提供具体域或与集合相关的信息。对于每个区,至少有两个域名服务器中 保存有该区中的所有计算机的验证信息。要想获得 math1 的 IP 地址,只需和 univ.edu 区的 域名服务器进行沟通,域名服务器就会返回需要的数据。 那么,怎样才能抵达 univ 大学的域名服务器呢?如果本地计算机没有装地址解析,DNS 有。当应用程序想查找关于 math1 的相关信息时,它会与一个本地域名服务器取得联系, 该服务器将为此实施交互式查询。向根域的一个域名服务器发出查询之后,该服务器便要 求得到 math1.maths.univ.edu 的 IP 地址。根域名服务器认出这个名字不属于自己的辖区, 而是属于.edu 域下面的一个辖区,它就会要求应用程序与.edu 区域名服务器取得联系,并 将列有所有.edu 域名服务器及其地址的清单封装起来。然后,本地域名服务器开始对清单 中的域名服务器——进行查找。它采用的方式类似于根域名服务器的方式,它知道 univ.edu 处的人们在运行自己的区,就把查询引向他们的服务器。最后,本地域名服务器再向其中 一个服务器提出查找 math1 的请求,math1 所在的域名服务器认出它是属于自己这一区之 后,便返回其对应的 IP 地址。

8.4.3 域名服务器

容纳某一区内计算机信息的域名服务器叫做这个区的验证服务器,有时也被称为主域 名服务器。任何对区内计算机的查询最终都会涉及到主域名服务器。

为了保证整个区的连贯性,主域名服务器必须能够得到同步更新。通过令其中一个主 域名服务器成为首要服务器可以做到这一点。这个首要服务器定期令传输区数据的其他域 名服务器作为其从属服务器,从而从数据文件中载入该区信息。

多个域名服务器可以分担工作负荷,也可以作为备用服务器。如果一个域名服务器失效,例如系统崩溃或丢失网络链接时,所有的查询都会转向其他服务器。

第9章 配置 Linux 网络

本章主要介绍在 Red Hat Linux 7.0 环境下如何配置网络。内容包括 Linux 网络配置文件介绍、手工配置 Linux 网络、使用网络配置工具 netcfg 等配置 Linux 网络。

9.1 网络配置文件

系统中重要的网络配置文件有:

- /etc/sysconfig/network
- /etc/hosts
- /etc/services
- /etc/host.conf
- /etc/nsswitch.conf
- /etc/resolv.conf
- /etc/init.d/network

下面将对这些文件逐一介绍。这些文件都可以在系统运行时进行修改。不用启动或停 止任何守护程序,更改会立刻生效。

1. 网络设置文件/etc/sysconfig/network

这个网络文件中包含控制和网络有关的文件和守护程序的行为的参数。下面是一个示例文件:

NETWORKING=yes FORWARD_IPV4=false HOSTNAME=apollo.ict.ac.cn DOMAINNAME=apollo.ict.ac.cn GATEWAY=159.226.43.190 GATEWAYDEV=eth0

最好使用 GUI 工具 netcfg 来编辑此文件。这个工具需要使用 X 服务器。

2. 映射 IP 地址和主机名的文件/etc/hosts

/etc/hosts 文件中包含 IP 地址和主机名之间的映射,还包括主机名的别名。IP 地址设计 得使计算机容易读,而人就很难记住它们。这就是创建/etc/hosts 文件的原因。/etc/hosts 文 件通常含有主机名 localhost 和系统管理员经常使用的系统别名。其他主机名通常使用 DNS 来解决。DNS 客户部分被配置在文件/etc/resolv.conf 中。 3.映射端口号和服务器名之间的文件/etc/services

/etc/services 文件中包含端口号和服务器名之间的映射。不少系统程序要使用这个文件。

4. 配置名字解析器的文件/etc/host.conf和/etc/nsswitch.conf

有两个文件声明系统到哪里寻找名字信息来配置名字解析器的库。

/etc/host.conf 文件声明名字系统的顺序(/etc/hosts 文件, DNS, NIS)。这些名字系统 用来解析主机名。/etc/hosts.conf 文件的每一行应包括下面指令中的一个,并带有参数。

/etc/nsswitch.conf 文件用于管理系统中多个配置文件查找的顺序。因此,它比 /etc/host.conf 文件提供更多的功能。

/etc/nsswitch.conf 文件中的每一行或者是注释(以#号开头)或者是一个关键字后跟冒 号和一系列要试用的有顺序的方法。每一个关键字是在/etc/目录可以被/etc/nsswitch.conf 控 制的/etc 文件的名字。如表 9.1 所示是/etc/nsswitch.confi 中的关键字及其功能。

关键字	功能
aliases	邮件别名
passwd	系统用户
group	用户组
shadow	隐蔽口令
hosts	主机名和 IP 地址
networks	网络名和号
protocols	网络协议
services	端口号和服务器名称
ethers	以太网号
rpc	远程进程调用的名称和号
netgroup	网内组

表 9.1 /etc/nsswitch.conf 中的关键字及其功能

5. 配置 DNS 客户的文件/etc/resolv.conf

文件/etc/resolv.conf 配置 DNS 客户。它包含主机的域名搜索顺序和 DNS 服务器的地址。

Red Hat Linux 中没有提供缺省的/etc/resolv.conf 文件。它的内容是根据在安装时给出的选项动态创建的。

6. 配置主机地址、掩码和缺省路由器的文件/etc/init.d/network

与其他的 UNIX 和 Linux 操作系统不同, Red Hat 并不能自动地通过/etc/hostname 和 /etc/hosts 文件来配置网络。为了改变主机缺省的 IP 地址,必须直接编辑/etc/init.d/network 脚本使其反映正确的网络配置。这个文件包括声明 IP 地址、掩码、网络、广播地址和缺省路由器的变量。

9.2 手工配置网络

手工配置网络是 Linux 用户的基本功,下面示例性地介绍如何手工配置主机名、IP 地址和网络接口。

9.2.1 设置主机名

多数情况下,网络应用程序依赖于本地主机名,这是在系统启动过程期间,利用 hostname 命令来完成的。

要设置主机名,应该使用下面的方法:

#hostname name

通常用没有域名的不合格主机名来进行试验。但是,由于本地主机名常用于查找主机的 IP 地址,所以必须确保解析器库能够查找主机的 IP 地址。这通常意味着必须在/etc/hosts 内输入主机名。

9.2.2 设置IP地址

在自己的主机上进行单机操作时可忽略这一步,因为只需要回送接口的 IP 地址,而这 个接口的地址始终都是 127.0.0.1。

如果想连接到一个现有网络中,必须要求该网络的管理员分配给一个 IP 地址。在自行 设置网络时,必须给自己分配一个地址。

如果网络没有接入因特网,便可以根据自己的喜好自由选择网络地址,只要它是合法的。同时,必须保证从 A、B 或 C 类地址中选择。但是,如果打算不久将接入因特网,现 在就应该想办法获得正式的 IP 地址。

要想同时操作若干个以太网,就必须将自己的网络分成若干个子网。

9.2.3 主机和网络文件

在将自己的网络分成若干个子网后,应该利用/etc/hosts 文件进行简单的主机名解析。 如果不用 DNS 和 NIS 进行地址解析,就必须将所有的主机信息放在一个 hosts 文件内。

即使运行 DNS 或 NIS,也应该先有/etc/hosts 文件内的某些子网的所有主机名。例如, 在无网络接口运行的情况下想有某类主机名解析,那么,这样不仅很方便,还可以用 rc.inet 脚本内的主机名。因此,在更改 IP 地址时,只需将一份更新过的主机文件复制到所有的机 器中,再重新启动即可,而不是重新逐个编辑数量庞大的 rc 文件。通常情况下,可把所有 的本地主机名和地址放入主机文件内。如果需要,还可将网关和 NIS 服务器添加在内。

查询主机 IP 地址时,如果要所有应用程序专用/etc/hosts 文件,必须对/etc/host.conf 文件进行编辑。在以关键字 order 开头的所有代码行前加上一个"#"号,就可以批注出所有的行,并插入下面这行:

order hosts

主机文件中每行都有一个条目,由 IP 地址、主机名和供选择的主机名别名清单组成。 这些字段用空格或标号隔开,而且地址字段必须在第一列内。跟在"#"号后面的被视作批 注,可忽略。

正如主机的 IP 地址一样,应该用一个好记的名字来代表某个网络。所以还有一个 /etc/networks 文件,该文件指明网络名和网络编号之间的对应关系。

9.2.4 配置网络接口

网络接口配置并初始化路由表的命令有两条。采用的基本工具是 ifconfig 和 route。配置任务通常是在系统启动时,从 rc.inet1 脚本开始进行的。

ifconfig 用于令接口能够被内核联网层访问。这涉及到 IP 地址和其他参数的分配以及接口的激活。这里的激活意思是内核将通过该接口收发 IP 数据报。要想启用接口,使用下面的命令:

ifconfig interface ip-address

上面的代码将 ip-address 分给接口,就激活了它。其他所有参数都采用默认值。例如, 默认的子网掩码取决于 IP 地址的网络类,例如代表 B 类地址的 255.255.0.0

第二个工具 route 用于在内核路由表内增加或删除路由。使用下面的命令调用它:

route [add|del] target

add 和 del 这两个参数用于判断是增加还是删除路由。

9.2.5 配置回送接口

首先激活的接口是回送接口,如下所示:

#ifconfig lo 127.0.0.1

有时,还可以看到本地主机用伪主机名代替了 IP 地址。ifconfig 将查找主机文件内的 主机名,其中一个条目将该主机声明为 127.0.0.1 的主机名:

localhost 127.0.0.1

如果要查看接口的配置情况,将接口名作为 ifconfig 的参数,使用下面的方法调用它 即可:

#ifconfig lo

由此可知,回送接口已经分到一个网络掩码255.0.0.0,这是因为127.0.0.1 是 A 类地址。 现在可以运行网络了,但还缺一项,即路由表中的一条,它告诉 IP 可以使用这个接口 作为去往目标127.0.0.1 的路由。可通过使用下面的命令完成:

#route add 127.0.0.1

通过上面的步骤,完全可以在一台独立主机上使用连网应用程序了。在 rc.inet1 增加上面的代码行,并保证从/etc.rc 执行这两个 rc.inet 脚本之后,可能需要重新启动机器。试着

运行各种应用程序。例如, telnet localhost 将建立一条 telnet 链接,并给出一个登录提示。

9.2.6 配置以太网接口

以太网接口的配置和回送接口的配置大致相同,唯一例外的是使用子网技术时,它要 求的参数比后者要多一些。

使用下面的命令调用 if config:

#ifconfig eth0 191.72.1.2 netmask 255.255.255.255.0

这样,便为 eth0 接口分配了一个 IP 地址 191.72.1.2。如果省略网络掩码, ifconfig 就会 根据 IP 网络的类别推断出网络掩码是什么,而且这个网络掩码将是 255.255.0.0。

接下来的情形和回送接口类似。同样必须安装一个路由条目,向内核报告通过 eth0 接口能够抵达的网络有哪些。应将 route 当作下面的代码行进行调用:

#route add -net 191.72.1.0

-net 选项被选用,因为 route 既可以处理通向网络又可处理通向单个主机的路由。当给 出点分十进制地址后,它通过观察主机部分知道它是一个网络还是一个主机。如果地址的 主机部分是 0, route 假定它代表网络,否则它被视为主机地址。因此 route 认为 191.72.1.0 是主机地址而非网络地址,因为它不知道使用了子网。所以必须使用-net 标记显式告诉它 此地址代表网络。

现在,基本的配置步骤告一段落。

9.3 网络配置工具

Red Hat Linu 7.0 中有几个配置网络的工具,使用它们可以简便地完成配置工作。下面 将介绍其中主要的几个。

9.3.1 网络配置工具netcfg

网络配置工具 netcfg 可以简便地配置网络参数,如 IP 地址、网关地址、网络地址和域 名服务器等。还可以增加、删除、配置、激活和关闭网络设备,以及取别名等。netcfg 的 使用界面如图 9.1 所示。

当使用 netcfg 网络配置工具时,单击 Save 按钮可以将所做修改保存到硬盘上;若不想修改而直接退出的话,直接单击 Quit 按钮即可。

下面将分别介绍 netcfg 使用界面中的各个选项的功能。

配置名字

网络配置工具的 Names 选项卡有两个主要的功用:设置主机名和计算机域名以及设定 哪一个域名服务器用于网络上计算机的查找。网络配置工具并不能将一台计算机配置为域 名服务器。编辑一个域或增添新的信息只需单击某个域然后输入所需的信息即可。
Network Confi	gurator				
Names Ho	sts Inte	rfaces	Routing		
Hostname:					
Domain:					
Search for host	tnames in a	dditional	domains:		
					A
Nameservers	1			 	
	1				
∥I	1				
	Save			Quit	

图 9.1 netcfg 的使用界面

配置主机

Hosts 选项卡可以增加、编辑和删除/etc/hosts 中的记录。增加和编辑一条记录的方法是相同的,都将弹出一个编辑对话框,只要在其中输入新的信息然后单击 Done 即可。

如图 9.2 所示是 Hosts 选项卡的使用界面。

	Network (Configurat	tor			
	Names	Hosts	Interfaces	Routing		
I	IP		Na	ume		Nicknames 🔼
I	127.0.0.1		Apollo			localhost.localdomain localhost
			Edit /etc/hos IP: Name: Nicknames:	ts	Icel	
			Add	Edit	Ren	nove
		8	ave			Quit

图 9.2 Hosts 选项卡的使用界面

• 配置网络接口设备

如果在安装了 Red Hat Linux 之后又安装了一块网络设备,或者没有在安装时配置以太 网卡,可以使用 Interface 选项卡添加设备及其选项。如图 9.3 所示是 Interface 选项卡的使 用界面。要增加一个设备,应该首先单击 Add 按钮,然后选择要增加的接口类型即可。

配置路由

使用 netcfg 的 Routing 选项卡可以配置默认网关和网关设备等。

Routing 选项卡配置界面如图 9.4 所示。如果想添加网关,单击 Add 按钮,再配置相应的参数即可。

Net	vork Configurat	or					_ 🗆 ×
Nam	es Hosts	Interfaces	Routing				
	Interface	[I	Р	proto	atboot	active	
lo		127.0.0.1		none	yes	active	
ethO		159.226.43.4		none	yes	active	
		[- [-	. [((.		V
	Add Edit		lias Rem	iove Act	ivate [Jeacuvate	
	S	ave			Quit		

图 9.3 Interface 选项卡的使用界面

Network Config	urator				
Names Hos	ts Inte	rfaces R	outing		
🔲 Network Pac	ket Forwa	rding (IPv4)			
Default Gateway	<i>r</i> :	159.226.43.4	Ļ		
Default Gateway	/ Device:				
Interface	Netwo	rk Address	Netmask	gateway	
					$\overline{\nabla}$
		Add	Edit Remove		
	Save			Quit	

图 9.4 Routing 选项卡的使用界面

9.3.2 网络配置工具netconf

网络配置工具 netconf 是一个功能强大的配置工具,使用它可以配置客户端的参数,包括主机名、DNS、网关和路由等;可以配置服务器端的参数,包括 NFS 和 IP 别名;还有一些杂项配置。

网络配置工具 netconf 的使用界面如图 9.5 所示。

这个使用界面共有3个标签,下面将分别介绍单击这些标签后弹出的各选项卡。

· Client tasks 选项卡

Client tasks 选项卡,如图 9.5 所示。该选项卡中有 6 个长条按钮,下面只介绍前 4 个 按钮的作用。

单击 Basic host information 按钮,出现的界面如图 9.6 所示。界面中有 4 个标签。单击 Host name 标签,在弹出 Host name 选项卡中,只有一个用于填写本机名的文本框。在 Adaptor 1 选项卡中(见图 9.6),可以填写网络接口 1 的相关参数,如 IP 地址和网络掩码等内容。





This host basic configur	ration
You are allowed to contri which are specific to this to its main connection to	ol the parameters host and related the local network
Host name Adaptor 1 2	T Enabled
Config mode	🗢 Manual 😞 Dhop 🕹 Bootp
Primary name + domain	
Aliases (opt)	
IP address	159.226.43.4
Netmask (opt)	255.255.255.0
Net device	eth0 7
Kernel module	ne2k-pci
I/O port (opt)	
Irq (opt)	7
Accept	Cancel Help

图 9.6 Basic host information 的配置界面

单击Name server specification(DNS)按钮,出现的界面如图9.7所示。在这个界面中,可以指定用于解析主机名到IP地址的DNS服务器的IP地址。可以指定多个DNS服务器的IP地

址,图中只用了本地回送地址127.0.0.1。

Resolver configuration	
You can specify which no to resolv host ip number, to handle this on a TCP/II are the local /etc/hosts fil (see "information about or the NIS system	ame server will be used Using the DNS is P network. The others e other hosts" menu
DNS usage	TDNS is required for normal operation
default domain	
IP of name server 1	127.0.0.1
IP of name server 2 (opt)	
IP of name server 3 (opt)	
search domain 1 (opt)	
search domain 2 (opt)	
search domain 3 (opt)	
search domain 4 (opt)	
search domain 5 (opt)	
search domain 6 (opt)	
Accept	Cancel Help

图 9.7 Name server specification(DNS) 配置界面

单击 Routing and gateways 按钮,出现的界面如图 9.8 所示。这个界面中有许多按钮,可以配置有关路由的各种信息。

Routes to othe	r networks	_ = ×
If your network you must tell ho For many setup is good enough If you only talk t then all this is n	has access to other networks w to reach those networks , simply setting the default ro to machines on the local net ot necessary.	s ute work
Set	Defaults	
	other routes to network	(S
	other routes to hosts	
	routes to alternate local r	nets
Configure	the routed daemon	
	Quit Help	

图 9.8 Routing and gateways 配置界面

单击 Host name search path 按钮,出现的界面如图 9.9 所示。这个界面有许多单选按钮,用于指定名字解析所用的顺序。本图中的顺序是 hosts、dns,这也是最常用的选择。在系

统启动时,用 hosts 文件查找主机名对应的 IP 地址,然后用 DNS 继续这种查找。

_		
 Name service access		
You must tell the system the various name servio	n in which order ces must be probed	
hosts mean /etc/hosts in NIS stand for Network In dns stands for Domain I	s probed nformation System Name Service	
☐ Multiple IPs for one	e host	
🗢 hosts, NIS, dns	🗢 hosts, NIS	🗢 hosts, dns, NIS
🗢 hosts, dns	🗢 hosts	🗢 NIS, hosts, dns
🗢 NIS, hosts	💠 NIS, dns, hosts	🔷 NIS, dns
⇔ NIS	🔷 dns, hosts, NIS	🔷 dns, hosts
🗢 dns, NIS, hosts	🧼 dns, NIS	🧼 dns
Acce	pt Cancel H	Help

图 9.9 Host name search path 配置界面

· Server tasks 选项卡

Server tasks选项卡可以配置服务器选项,如图9.10所示。

Network configurator
This package allows you to configure a TCP/IP network from scratch using ethernet and modem (or other serial connection)
Client tasks Server tasks Misc
Exported file systems (NFS)
IP aliases for virtual hosts
Quit Help

图 9.10 Server tasks 选项卡

• Misc 选项

Misc选项卡可以配置一些杂项配置,如图9.11所示。

Network configurator
This package allows you to configure a TCP/IP network from scratch using ethernet and modem (or other serial connection)
Client tasks Server tasks Misc
Information about other hosts
Information about other networks
Linuxconf network access
Quit Help

图 9.11 Misc 选项卡

9.3.3 网络配置工具netconfig

配置工具netconfig是一个文本配置工具,主要用于配置网络接口和IP地址。在终端输入如下命令:

netconfig

将会出现一个询问是否配置网络的界面。使用 Tab 键将光标移动到 Yes 按钮上,用回车键确认,进入配置界面,如图 9.12 所示。

Terminal	- 🗆 ×
File Edit Settings Help	
netconfig 0.8.3 (C) 1999 Red Hat, Inc.	A
Configure TCP/IP Please enter the IP configuration for this machine. Each item should be entered as an IP address in dotted-decimal notation (for example, 1.2.3.4)	
IP address: Netmask: Default gateway (IP): Primary pameserver:	
(Tab)/(Alt-Tab) between elements 「 〈Space〉 selects 「 〈F12〉 next scree	n 🔽

图 9.12 网络配置工具 netconfig 的界面

9.4 习题

- 1. 练习手工配置网络。
- 2. 使用 netcfg 工具配置网络。
- 3. 使用 netconf 工具配置网络。

第10章 配置域名服务

本章主要介绍如何在 Linux 环境下配置域名服务。内容包括域名服务简介、使用 hosts 文件、DNS 简介、配置 DNS 服务器和解析器以及配置 Cache-Only 服务器示例等。

配置域名服务是一项复杂的工作,通过学习本章,可以对配置域名服务有一个比较全 面的认识。

10.1 域名服务简介

连接到 TCP / IP 网络上的计算机都是由它的 IP 地址标识的。IP 地址是一套由 4 组数 字组成的代码,用以指定网络地址以及该网络内部主机的地址。IP 地址不容易记忆,因此 每个 IP 地址的域名也可以用于标识一台主机。在第 9 章已经指出,域名由两部分组成,主 机名和域。主机名是主机的专有名称,域用于标识该主机属于的网络。美国使用的域扩展 名通常说明了主机的类型。例如,.edu 用于教育部分,而.com 用于商业。国际域通常的扩 展名用于指定所处国家。例如,.du 代表德国,.au 代表澳大利亚。主机名、域以及扩展名 共同组成一个名称,通过这个名称可以找到一台计算机。而域又可以再分成子域。

网络上的计算机即使有自己的域名也仅能够通过 IP 地址标识。可以通过域名访问网络 上的一台计算机,但要先在一个数据库中查询以将域名转化为对应的 IP 地址。然后,网络 使用这个 IP 地址而不是域名来访问对应的计算机。大型 TCP / IP 网络(例如 Internet)出 现之前,在每台机器上维护一个 IP 地址和对应域名的信息是可行而且方便的。每当访问一 个域名时,就查找这个文件,从而找到对应的 IP 地址。

随着网络的增大,上述方法变得不太实际,尤其是在 Internet 上。为了提供将域名转换为 IP 地址的服务,将产生出域名的数据库存放到服务器上。为了找到一个域名的 IP 地址,向域名服务器发送一个请求,然后,域名服务器会查询对应这个域名的 IP 地址,可将其返回。在大型的网络中,可能有多个域名服务器覆盖网络的不同部分。如果一个域名服务器找不到对应域名的 IP 地址,它会将服务请求发送给另一个域名服务器。

域名服务器是由解析器查询的。解析器是专门用于从域名服务器中获取IP地址的程序。 要想在系统上使用域名服务器,必须配置解析器。本地的解析器是通过文件/etc/host.conf 和/etc/resolv.conf 来配置的。

10.2 使用hosts文件

尽管 DNS 是域名解析的首选方法,但在大多数计算机中仍然可以找到/etc/hosts 文件。 它有助于提高经常请求的 IP 地址的查找速度,如本机的 IP 地址。在引导期间,在 DNS 可 以引用之前,计算机需要知道一些主机名到 IP 地址的映射(例如 NIS 服务器)。这些主机的 IP 地址到主机名的映射保存在/etc/hosts 文件中。

10.2.1 配置host.conf文件

host.conf 文件列出了解析器的选项。文件内的选项必须以单行的形式出现。每个选项 可以有多个域,中间用空格或制表符分隔。在一行的行首如果是#,则表示该行的内容是注 释。选项告诉解析器使用何种服务。该文件中各行的顺序是很重要的,解析器会由上至下 读取该文件的内容。

• order

该选项决定解析服务的使用顺序。有效选项有:

bind 用于查询域名服务器。

hosts 用于/etc/hosts 内的查找。

nis 用于 NIS 查找。

可以同时指定一个选项或所有的选项。各选项在命令行出现的顺序将决定各自的服务 顺序。

• multi

其选项有 on 和 off。它用于判断/etc/hosts 内的主机是否获得允许使用多个 IP 地址。如 果可以,这个主机就被称为"多宿主主机"。这个标记对 DNS 和 NIS 查询是没有用的。

nospoof

DNS 允许查找属于某一台主机的主机名。域名服务器提供假主机名的作法称为"电子欺骗"。为防止这种情况的发生,可将解析器配置为查看原始 IP 地址是否真的与获得的主机名相符。如果不相符,这个主机名就会被丢弃并返回一个错误。将"电子欺骗"设为 on,就可以启用这个行为。

spoofalert

该选项将 on 和 off 作为其参数。如果设置为 on , 任何 " 电子欺骗 " 尝试将令解析器把 消息记录到系统日志中。

• trim

该选项将域名作为其参数,在实施主机名查找之前,这里的域名会从主机名文件中被删除。这个特性对主机条目是非常有用的,因为只想指定不带域名的主机名。对绑定本地域名的主机所进行的查找将删除这个特性,因此,允许/etc/hosts 文件内的查找继续下去。 trim 选项累积起来,使其很容易将本地主机视为处于本地域内。

10.2.2 解析器环境变量

利用大量的环境变量,可以改写 host.conf 文件中的设置。它们是:

1. RESOLV HOST CONF

该变量指定准备读取的文件,并用该文件来代替/etc/host.conf。

2 . RESOLV SERV ORDER

改写 host.conf 文件内指定的 order 选项。指定的服务分别为 hosts、bind 和 nis,并分别 用空格、逗号、冒号和分号隔开。

3 . RESOLV SPOOF CHECK

判断针对电子欺骗的尺度。如果设置为 off,则表示彻底取消。warn 和 warn off 这两个 值都会启用电子欺骗校验,但将分别打开和关闭记录。* 值将打开电子欺骗校验,但会像 host.conf 文件内定义的那样,保留记录设备。

4 . RESOLV MULTI

可选项有 on 和 off,用于改写 tthost.conf 文件内的 multi 选项。

5 . RESOLV OVERRIDE TRIM DOMAINS

该环境变量指定一份 trim 域清单,它们将用于改写 host.conf 内的域。

6 . RESOLV ADD TRIM DOMAINS

该环境变量指定一份 trim 域清单,这些域将增加到 host.config 中指定的域内。

10.3 DNS 简 介

DNS 是一个复杂的系统。本节将对 DNS 的一些概念进行介绍和澄清。

10.3.1 配置工具

要为站点配置 DNS,需要如下工具:

1. named 程序

大多数 DNS 的实现,包括 Red Hat Linux 附带的版本,都使用 BIND。BIND 随 Red Hat Linux 一起发售,它代表 Berkeley Internet Name Domain。在 Linux 系统上,域名服务是通过 named 的程序来提供的。

named 是在 DNS 服务器上运行的守护程序,它用来处理查询。如果它不能对某一查询做出回答,它将负责把这一请求转发给可以解答的服务器。除了查询,named 还负责执行区间传输。通过区间传输,可以在 Internet 上传播修改过的 DNS 信息。如果没有在 Red Hat Linux 7.0 安装时安装 named 守护程序,需要利用 BIND 软件包进行安装,BIND 软件包可以从安装光盘上得到,文件名是 bind-8.2.2_P5-25.i386.rpm。

named 守护程序通常在引导时启动。以下是手动启动、停止和重新启动 named 的命令:

#/etc/rc.d/init.d/named start
#/etc/rc.d/init.d/named stop
#/etc/rc.d/init.d/named restart

在对/etc/named.conf 文件或者任何由该文件引用的文件进行更改以后,在更改生效之前,必须重新启动 named。

2.解析程序库

解析程序库使客户程序能够执行 DNS 查询,这个库被集成在 Linux 的标准库中。解析 程序库从/etc/resolv.conf 中获得配置信息。

3.nslookup 程序

nslookup 是一个从命令行调用的实用程序,它用来确保解析程序和 DNS 服务器正确配置。这一任务是通过把一个主机名解析为 IP 地址或者把一个 IP 地址解析为域名来完成的。 要使用 nslookup,只需将要解析的地址作为它的参数即可。

4. traceroute 程序

traceroute 实用程序用来确定数据包从当前网络传输到其他网络所采用的路径。它对调 试网络连接非常有用,特别是当怀疑故障发生在其他网络中时。

使用 ICMP 协议, traceroute 将查找到达目标主机所需经历的每台计算机并显示出相应 的名字和 IP 地址。同每个名字在一起的还有以毫秒为单位的 3 组数字, 这些数字用来表示 到达目标所需花费的时间。

每个名字前面都有一个数值,表示要到达目的主机需要经过的中继次数。中继次数表 明处理该数据包需要的中间计算机个数。可以推测,相隔一个或者两个中继的计算机比相 隔 30 个中继的计算机要近得多。

要使用 traceroute,只需要以目的主机名或 IP 地址作为命令行参数即可。

10.3.2 DNS术语

如表 10.1 所示, 是基本 DNS 术语一览表。

表 10.1 基本 DNS 术语一览表

术语	定义
DNS client	所有联网的计算机上的软件组件通过查询指定给它的 DNS 服务器,查找某个名字
(DNS 客户)	的 IP 地址(或者反过来)。在 Red Hat Linux 计算机上, 客户从/etc/resolv.conf 文件获
	得其配置信息。有时候术语 DNS 客户指的是计算机本身 resolver (解析程序)为了
	实用而对 DNS 客户的称谓,二者是同义词
DNS server	向查询客户返回名字到 IP 地址转换(或者反过来)的软件组件。 在进行该操作时,该
(DNS 服务器)	服务器可能要向其他 DNS 服务器进行查询。在 Red Hat Linux 计算机上,服务器从
	/etc/named.conf 文件和该文件引用的文件中获得配置信息。在作为 DNS 服务器的
	Red Hat Linux 计算机上, DNS 服务由 named 守护程序提供
resolve	将名字转换为 IP 地址或者反过来的操作。DNS 完成解析,有时是其他软件 zoneDNS
(解析)	服务器授权的子域或者子网
master	在一个区上具有权限的域名服务器 , 从本地区数据文件获得数据。 注意域名服务器
(主 DNS)	可以是一些区的主 DNS,但是是从其他域的 DNS 术语定义
primary	主 DNS 的同义词
(初始 DNS)	

(续表)
ς.	ースしく	,

术语	定义
slave	在一个区上具有权限的域名服务器,它从域转换程序中的另一个域名服务器获得数
(从 DNS)	据。其他的域名服务器可以是主 DNS 也可以是另一个从 DNS。一旦获得信息,则
	存储在本地 , 因此即使数据源关闭它仍然能够工作。 注意域名服务器可以是一些域
	的主 DNS,但是是其他域的从 DNS
secondary	从 DNS 的同义词
(后备 DNS)	
Zone transfer	区间传输程序从主 DNS 或者从 DNS 服务器传输数据到另一个从 DNS 服务器。接
(区间传输程序)	收的从 DNS 在超出刷新时间或者接收到发送服务器来的数据已经改变的通知后初
	始化区间传输程序

10.3.3 域名服务器和解析器

域名系统是采用客户机/服务机制。解析器(Resolver)是客户方,它负责查询域名服 务器,解释域名服务器的应答,并将查询到的有关信息返回请求的程序或用户。

存储域名空间信息的程序(主机)就是域名服务器。一个域名服务器可以管理一个区, 也可以管理多个区。域名服务器有以下几种类型。

主域名服务器

一个区由主域名服务器管理,它维护这个区的所有域名空间信息。一个区至少要有两 个主域名服务器:第一主域名服务器和第二主域名服务器,它们位于不同的主机上。当第 一主域名服务器关闭、坏了或过载时,第二主域名服务器能作为备份服务器工作。第一主 域名服务器与第二主域名服务器的区别在于:第一主域名服务器从其硬盘上装入其管理区 内域名数据,并对本区内其他域名服务器授权;第二主域名服务器从第一主域名服务器获 得授权,从第一主域名服务器取得其管理区内域名数据,并周期性地与第一主域名服务器 上的数据进行比较,更新其管理区内的域名数据。

• Cache-Only 服务器

所有的域名服务器都是 Cache-Only 服务器。这就是说,域名服务器把从别的域名服务器所接收到的信息都保存在 Cache 中,直到信息失效。一个 Cache-Only 服务器对任何区都不负责任,它将收到的查询请求转发给其他域名服务器,直到获得所需信息。

转发服务器

转发服务器(Forward Server)接到地址映射查询请求时,在其 Cache 中查找。如果找不到,就把请求依次转发到指定的域名服务器,直到查询到结果为止。否则返回无法映射。

Linux 上的名字服务是由 named 程序来执行的。这是属于 bind 套件的一部份,它缺省 安装于/usr/sbin/named。可以直接使用它做构造名字服务器。

DNS 是个以整个网络为范围的资料库,所以要小心地将资料放进去。

每个启用网络的 Linux 计算机都有 DNS 客户软件,通常称为解析程序。DNS 客户软件 按照/etc/resolc.conf 文件中提供的 DNS 服务器的顺序对它们进行查询。一台计算机的 DNS 客户程序可以在同一台计算机上、另一台计算机上或者有时候在这两台计算机上指定其服 务器。 DNS 服务器是配置为返回查询数据的计算机。DNS 服务器软件依赖于/etc/named.conf 文件以及该文件中的区引用指向的文件。客户进行查询,服务器进行回答(有时在询问其 他服务器以后才回答)。

但是,当一台计算机同时是 DNS 客户和 DNS 服务器时,使客户指向服务器则容易使 人混淆。客户和服务器可以表现为一个实体因而产生混淆。因此,记住:/etc/resolv.conf 属 于 DNS 客户或者说解析程序;而/etc/named.conf 和它引用的其他文件属于 DNS 服务器。

10.4 配置服务器和解析器

named 的启动通常是在系统启动时开始的,而且会一直处于运行状态直到关机。它从 一个名为/etc/named.boot 的配置文件中取出自己的有关信息。

要运行 named,只需在提示行键入如下命令:

#/usr/sbin/named

然后, named 出现并读取 named.boot 文件和所有指定的 zone 文件。再以 ASCII 的形式 将自己的进程 ID 写入/var/run/named.pid。如有必要,将从主服务器下载所有的 zone 文件,并在端口 53 上开始监听 DNS 查询。

10.4.1 配置服务器

1. named.boot 文件

named.boot 文件一般都非常小,其中包含一些指向 master 文件和其他域名服务器的指针。master 文件中包含的是区信息。启动文件中的批注以分号开头,并扩展到下一行。下面介绍 named.boot 文件中最重要的几个选项:

· diretory

该选项指定区文件驻留的目录。另外,还可能给出与该目录相关的文件名。

重复利用 directory 选项,可指定若干个目录。按照 Linux 文件系统标准,这个目录应 该是/var/named。

primary

该选项将一个域名和一个文件名作为参数,声明本地域名服务器属于 named 域。作为 一个主要域名服务器, named 从给定的 master 文件中加载区信息。一般地,每个 root 文件 中至少总应该有一个主要条目用于逆向映射网络 127.0.0.0,该网络就是本地回送网络。

secondary

该选项将域名、地址清单和文件名作为参数。它声明本地域名服务器是指定域的辅助 服务器。辅助服务器中也保存有该域名的管理数据,但它不是从不同的文件收集这些数据, 而是试图从主要服务器上下载。因此,地址清单中至少应该为 named 指定一个主要服务器 的 IP 地址。然后,本地服务器将依次和各域名服务器取得联系直到成功传送区数据库。这 样,这个区数据库便保存在第三个参数中指定的备份文件中。如果无主要域名服务器作出 应答,本地服务器只好从备份文件中获取区数据。最后, named 将定期对区数据进行更新。 详情可参见 SOA 源记录类型。

cache

该选项将域名和文件名作为参数。这个文件中包含根服务器提示,是一份指向根域名 服务器的记录清单。目前只能识别 NS 和 A 类型的记录。域名参数一般是根域名"."。这 一点对 named 是绝对重要的:如果 cache 选项不能在根文件中执行, named 就根本不能开 发出一个本地缓冲区。如果下一个服务器查询不是在本地网络上进行,必然会降低性能, 增加网络负担。更有甚者, named 将不能抵达任何一台根域名服务器,因而也不可能解析 自己管理范围之外的其他地址。但在使用转发服务器时,情况有所不同(比较下面的 forwarders 选项)。

forwarders

该选项将地址清单作为参数。该清单中的 IP 地址指定的是一系列的域名服务器,如果 named 不能从自己的本地缓冲区内解析地址查询,它就会要求查询清单中的这些域名服务器。named 依次对这些服务器进行查询,直到其中之一能够作出应答。

• slave

该选项令域名服务器作为从属服务器。也就是说,这类服务器自身永远不会执行递归 查询,只是将查询转发到 forwarders 中指定的服务器。

2. named.conf 文件

/etc/named.conf 文件是在 named 进程启动时读入的文件。

在 named.conf 文件中两个最常用的段起始关键字为 options 和 zone。下面是一个 named.conf 示例文件。

// generated by named-bootconf.pl

options {

```
directory "/var/named";
```

```
/*
```

* If there is a firewall between you and nameservers you want

* to talk to, you might need to uncomment the query-source

* directive below. Previous versions of BIND always asked

```
* questions using port 53, but BIND 8.1 uses an unprivileged
```

```
* port by default.
```

```
*/
```

// query-source address * port 53;

```
};
```

```
//
// a caching only nameserver config
//
zone "." IN {
  type hint;
  file "named.ca";
```

};

```
zone "localhost" IN {
  type master;
  file "localhost.zone";
  allow-update { none; };
};
```

```
zone "0.0.127.in-addr.arpa" IN {
  type master;
  file "named.local";
  allow-update { none; };
};
```

options 部分保持整个 DNS 服务器的全局信息。在样本中只包含一条信息(directory 语句)通知 named 在配置中提及的所有文件名的位置。

zone "."是缓冲区。缓冲区不是主区也不是从区,但是包含服务器软件使用的许多提示,即 typehint;语句。zone "."的引用文件为 named.ca,在 Red Hat 安装时创建。文件 named.ca 包含 Internet 上所有根 DNS 服务器的列表。这些根服务器用于初始化 named 的缓 冲。

每个区有一个 type 语句指明 master、slave 或者 hint,以及一个 file 语句指向包含该区数据的文件。类型为 slave 的文件有一个嵌套的 master 部分。

每个区部分定义一个授权区,通常是一个域、一个子域或者在反序 DNS 方式下为子网。 几乎所有的区都定义一个文件,并从该文件获得信息。每个区有一个 type 语句。

另一个在 zone 和 options 部分经常出现的语句是 notify 语句。该语句可以是 notify yes 或者 notify no。默认为 notify yes ,因此不必输入 notify yes ,除非是为了文字说明。如果 notify 为 yes ,则通知该区的从服务器数据已经更改 ,因此它们可以初始化区间传输;如果为 no ,则不进行通知。经常插入 notify no 可以防止假的域扰乱真正的 Internet 域名服务器。

在 options 部分使用 forwarders 语句可以指定向一个或者多个域名服务器发送本地无法 解析的请求。

该语句发送没有解析的请求到两个服务器中,而不是将它们发送到本地缓冲 DNS。当 通过外部传输信息可以受益时该方法比较好。如果所有的内部服务器通过一个或者两个服 务器解析外部名字,则意味着大多数请求可以在内部解决,否则所有的服务器将内部请求 发送到外部服务器寻求解决。

如果因为某种原因转发器无法应答请求,则该请求试图通过通常的服务器缓冲 DNS。 要禁止从一个 DNS 服务器来的任何远程请求,则紧接着 forwarders 语句放入 forward-only; 语句。

10.4.2 配置解析器

在配置解析器库使之利用 BIND 域名服务进行主机查找时,还必须告诉它准备使用哪些域名服务器。对此有一个专门的文件,名为/etc/resolv.conf。如果该文件不存在,或者为

;		
directory		/var/named
cache		named.ca
primary	0.0.127.in-addr.arpa	named.local

directory 这行告诉 named 到哪里去找寻文件。所有其后命名的文件都是相对于此目录的。根据 Linux 文件系统标准,正确的目录应该是/var/named。

/var/named/named.ca 文件是在此命名的,该文件应该包含如下内容(这里删除了注释):

	3600000 IN	NS	A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.	3600000	А	198.41.0.4
	3600000	NS	B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.	3600000	А	128.9.0.107
	3600000	NS	C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.	3600000	А	192.33.4.12
	3600000	NS	D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.	3600000	А	128.8.10.90
	3600000	NS	E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.	3600000	А	192.203.230.10
	3600000	NS	F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.	3600000	А	192.5.5.241
	3600000	NS	G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.	3600000	А	192.112.36.4
	3600000	NS	H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.	3600000	А	128.63.2.53
	3600000	NS	I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.	3600000	А	192.36.148.17
	3600000	NS	J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.	3600000	А	198.41.0.10
	3600000	NS	K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.	3600000	А	193.0.14.129
	3600000	NS	L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.	3600000	А	198.32.64.12
	3600000	NS	M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.	3600000	А	202.12.27.33

(2) 建立/etc/resolv.conf 文件。

还需要一份像这样的/etc/resolv.conf 文件:

search subdomain.your-domain.edu your-domain.edu nameserver 127.0.0.1

search 这行指出任何想联网的主机名字应该搜寻的领域。nameserver 这行指出本机可以在哪个位址上找到一台名字服务器,在这个例子中是自己的机器,因为在它上面执行 named。如果想列出好几个名字服务器并把它们都放在一行 nameserver 里,就要用空格隔

开。

如果某个客户端尝试找寻 foo,那么首先尝试的是 foo.subdomain.your-domain.edu 这个 名字,接下来是 foo.your-fomain.edu 这个名字,最后则是 foo 这个名字。如果有某个客户端 尝试找寻 sunsite.unc.edu,那么首先尝试的是 sunsite.unc.edu.subdomain.your-domain.edu 这 个名字,接下来是 sunsite.unc.edu.your-domain.edu 这个名字,最后则是 sunsite.unc.edu 这个 名字。

这个例子假设本机属于 subdomain.your-domain.edu 领域,可能会叫做 your-machine. subdomain.your-domain.edu。在 search 这行不应该包含 TLD(顶层领域 Top LevelDomain, 在这个例子中是 edu 这个领域)。如果经常需要连到另外一个领域里的主机,那么可以像 下面把该领域这样加进 search 这行里:

search subdomain.your-domain.edu your-domain.edu other-domain.com

(3) 修正/etc/nsswitch.conf 和/etc/host.conf 文件。

根据 libc 版本的不同,选择修正/etc/nsswitch.conf 或者/etc/host.conf 文件。

/etc/nsswitch.conf 是一个很长的文件,它指出到何处去取得资料,从什么文件或是资料 库取得。它的顶端经常会包含一些注解。找出以 hosts:开头的行,它应该是这样的:

hosts: files dns

如果文件里没有以 hosts:开头的行,那么把上面一行加进去。这一行指出程序应该先在 /etc/hosts 文件里找寻,然后根据 resolv.conf 询问 DNS。

/etc/host.conf 文件可能包含许多行,其中应该有一行以 order 开始,而且它看起来会像这样:

order hosts, bind

如果文件里没有 order 这一行,那么应该添进去。它告诉主机名字解析函数先在/etc/hosts 里找寻,然后查问名字服务器。

(4)运行 named。

键入 ndc start 并且按下回车键。运行 named 的时候如果使用:

tail -f/var/adm/messages

观察一下指令系统记录信息文件(通常是/var/log/messages 文件),那么应该会看见下面的信息:

May 29 10:50:56 Apollo named[1156]: starting. named 8.2.2-P5 Sat Aug 5 13:21:24 EDT 2000 ^Iprospector@porky.devel.redhat.com:/usr/src/bs/BUILD/bind-8.2.2_P5/src/bin/named

May 29 10:50:56 Apollo named[1156]: hint zone "" (IN) loaded (serial 0)

May 29 10:50:56 Apollo named[1156]: master zone "localhost" (IN) loaded (serial 42)

May 29 10:50:56 Apollo named[1156]: master zone "0.0.127.in-addr.arpa" (IN) loaded (serial 1997022700)

May 29 10:50:56 Apollo named[1156]: listening on [127.0.0.1].53 (lo)

May 29 10:50:56 Apollo named[1156]: listening on [159.226.43.4].53 (eth0)

May 29 10:50:56 Apollo named[1156]: Forwarding source address is [0.0.0.].1029

May 29 10:50:56 Apollo named[1193]: Ready to answer queries.

如果有错误, named 将会指明有错误的文件, 可能是 named.boot 或 root.cache。杀掉 named 程序并检查这些文件。

可以用 nslookup 检查这些文件。如下所示:

nslookupDefault Server: localhostAddress: 127.0.0.1

>

如果得到这样的回应,就表示它已经能够运作。

10.6 习 题

1. 在自己的系统的/etc 目录下查看 hosts 和 host.conf 文件,执行添加主机名等简单操作,查看效果。

2. 查看 named 的配置文件。

3. 启动、停止和重新启动 named, 然后查看系统日志, 并检查它是否正常。

第 11 章 配置 FTP 服务器

本章主要介绍 Linux 下 FTP 服务器的安装、配置和使用。内容包括 FTP 服务器简介、 如何安装 FTP 服务器、如何配置 FTP 服务器、一些 FTP 管理工具的使用以及一些 FTP 客 户工具的使用等。

11.1 FTP服务器简介

文件传输协议 (File Transfer Protocol, FTP) 可以通过网络把一台计算机上的文件传输 到另一台计算机上。由于所有常用平台都编有相应的客户和服务程序,因此使得 FTP 成为 执行文件传输最方便的方式。

FTP 服务器有两种配置方式。一种 FTP 服务器是匿名服务器。匿名服务器在无需帐号的情况下允许任何人对服务器进行连接并传输文件。由于服务器软件固有的复杂性,不可避免地会存在一些错误,从而使得未经许可的用户有可能可以进入系统。因此这种配置方式含有潜在的不安全因素,因此应该采取预防措施,只允许用户访问系统上的特定目录。如果用户决定安装一个匿名 FTP 服务器,一定要关注由 CERT (http://www.cert.org)发出的安全性公告,并及时更新服务器软件。

另一种 FTP 服务器仅供专门用户使用。这种 FTP 服务器是默认的配置方式,本章将介 绍这种方式。专用的 FTP 服务器只允许系统的用户通过 FTP 进行连接并访问特定的文件, 管理员通过为这些用户设置访问权限,从而明确地拒绝和接受对系统的访问。

11.2 安装FTP服务器

安装 RPM 形式的 FTP 服务器是很简单的。了解 FTP 服务器的工作原理对于使用服务器和客户都是必要的。

11.2.1 FTP服务器的工作原理

FTP 服务程序由/etc/xinetd.conf/文件进行控制,并且每当有人连接 FTP 端口时,它便 被自动调用。当检测到某个连接时,FTP 守护程序/usr/sbin/in.ftpd 便被调用,从而开始会话。 在/etc/xinetd.conf 文件中,默认的 Red Hat Linux 包含执行这一步骤所必需的行。

当服务器初始化后,客户需要提供用户名和相应的口令。两个特殊的用户名 anonymous 和 ftp 保留作为对公共文件的访问,任何其他访问都需要在服务器上有一个帐户。

如果使用自己的帐户访问服务器,将执行一个附加的检查以确保具有有效的 shell。 如果不具有有效的 shell,系统将拒绝他进入。要想使一个 shell 有效,它必须列在 /etc/shells 文件中,如果要安装一个新的 shell,必须把它添加到/etc/shells 列表中,以便使用 这个 shell 的用户能够通过 FTP 连接到系统。

当访问 FTP 服务器的用户进入系统后,最初将放在自己的主目录中。此时他们可以进入系统中他们具有访问权限的任何目录;而对于匿名用户,则有一些限制。

匿名用户进入系统后,最初被放在 FTP 用户的主目录中。在默认情况下,这一目录被 anonftp RPM 软件包设置为/home/ftp。当用户进入此目录后,FTP 服务器执行一个 chroot 系统调用,这一调用有效地把程序的根目录改变为 FTP 用户的目录。对系统中任何其他目 录的访问,包括/bin、/etc 和/lib 目录在内,都将被拒绝。对根目录的这种改变产生的副作 用是服务器将不再能看到/etc/passwd、/etc/group 以及其他必要的二进制文件。为了弥补这 一问题,anonftp RPM 软件包在/home/ftp 目录下创建了 bin、etc 和 lib 目录,并且在这些目 录中放进了必要的库和程序。这样,即使执行了 chroot 系统调用,服务器软件也能够对它 们进行访问。

由于安全性的原因,放在/home/ftp 目录中的文件被设置为只有服务器才能看到。因此, 在/home/ftp 下创建的其他一些目录应该被设置成是可读的。大多数 FTP 站点都把这样的文 件放在 pub 子目录中。

11.2.2 获取和安装FTP服务程序

Red Hat Linux 使用的是 wu-ftpd 服务程序,它以 RPM 的形式提供。在 Red Hat Linux 7.0 的安装过程中可以选择安装 FTP 服务器。如果没有安装该 RPM 软件包,还可以从 Linux 安装光盘上找到 wu-ftpd-2.6.1-6.i386.rpm 文件进行安装。

要安装该软件包,请加载 Red Hat Linux 7.0 光盘,并以 root 身份运行下面的命令:

#rpm -i /mnt/cdrom/RedHat/RPMS/wu-ftpd-2.6.1.-6.i386.rpm

如果要提供匿名服务,还要安装光盘上的 anonftp-3.0-9.i386.rpm。以 root 身份进入系统并运行下面的命令:

#rpm -i anonftp-3.0.9.i386.rpm

anonftp 软件包包含建立匿名 FTP 服务需要的文件, wu-ftpdrpm 文件包含 FTP 守护程序,在 ftprpm 软件包中可以找到 FTP 客户程序。完整的服务需要安装所有这 3 个 RPM。

使用 FTP 客户程序连接这台服务器可以测试安装是否能够正常运行。使用登录名而不 是匿名进入系统可以测试专用的 FTP 服务器。如果没有访问网络的权限,可以通过使用主 机名 localhost 测试 FTP 服务器。

11.3 配置FTP服务器

FTP 服务器在安装好之后有一些缺省的配置,可以通过修改配置文件来控制服务器的 特性。利用配置文件,可以控制哪些用户、在什么时间和从什么地点可以连接到服务器, 并且可以建立一个跟踪文件,对连接后所做的工作进行检查。下面将对这些配置文件做详 细的介绍。

11.3.1 /etc/ftpaccess文件

/etc/ftpaccess 文件用于控制哪些用户能够访问服务器。文件中的每一行是一个命令,这 些命令的功能如表 11.1 所示。

表 11.1 /etc/ftpaccess 的命令及其功能

功能	命令
控制访问权	class、autogroup、deny、guestgroup、limit、loginfails、private
控制服务器和客户共享消息	banner, email, message, readme
控制日志记录	log commands, log security, log syslog, log transfers
设置权限	chmod、delete、overwrite、rename、umask、passwd-check、path-filter、
	upload
杂项命令	alias, cdpath, compress, tar, shutdown

下面将按照表中的功能对一些重要的命令进行介绍。

1. 控制用户的访问权

控制用户访问站点的能力是精确调整匿名 FTP 服务器的关键部分。下面的命令用来定 义标准,用于确定每个用户应该归入哪一组。

· class 命令

class 命令用来定义可以访问 FTP 服务器的用户类。可以定义任意个类,每个 class 命 令行的格式如下:

class <classname> <typelist> <addrglob> [<addrglob>]

<classname>是类的名字,<typelist>是这一类的用户类型,<addrglob>是允许访问这个 类的 IP 地址范围。

<typelist>是一个由逗号隔开的列表,列表中的每一项是 anonymous、guest 和 real 这 3 个值之一。anonymous 用户以 anonymous 或 ftp 身份连接到服务器并只能访问公共文件。guest 用户在系统上没有帐号,但具有 guest 组关键部分的访问权。real 用户必须在 FTP 服务器上 具有帐号并且还要通过验证。

<addrglob>采用的是正则表达式的形式。*代表所有的站点。一个类可以和多个<addrglob>相联系。下面是一个示例:

class anyone anonymous *

它定义了一个类 anyone,这个类只包含匿名用户,这些用户可以从网络上的任何位置 连接服务器。下面这个示例:

class localclass real 159.226.43.*

只允许在 FTP 服务器上具有帐号的用户 (real) 从局域网上访问服务器。

autogroup 命令

autogroup 命令用来在匿名用户访问系统时把某个组的权限赋给他。autogroup 命令的格式如下:

```
autogroup <groupname> <class> [<class>]
```

<groupname>是匿名用户将要设置到的组的名字; <class>是使用 class 命令定义的类的 名字。groupname 必须在/etc/group 文件中出现。

· deny 命令

deny 命令根据主机的名字和 IP 地址来拒绝其对服务器的访问。deny 命令的格式如下:

deny <addrglob> <message_file>

<addrglob>是一个正则表达式,包含被拒绝的地址;<message_file>是一个文本文件, 其中的文本将在拒绝主机连接时,在这些被拒绝主机上显示。

guestgroup 命令

guestgroup 命令为用户提供受限的 FTP 权限。这一命令的格式如下:

guestgroup <groupname> [<groupname>]

这里的<groupname>是要限制的组的名字。

· limit 命令

limit 命令用来控制访问服务器的用户数。limit 命令的格式如下:

limit <class> <n> <times> <message_file>

 <class>是要限制的类, <n>是类中允许同时访问服务器的最多人数; <times>是限制的 有效的时间, <message_file>是达到最大限制数时显示给客户的文本文件。

· loginfails 命令

loginfails 命令用来设置客户登录尝试的允许失败次数。达到这一次数后,就会断开客 户机和服务器的连接。默认的次数是 5。loginfails 命令的格式如下:

loginfails <n>

<n>是登录尝试的失败次数。

private 命令

private 命令可以使用户不使用实际账号,通过 SITEGROUP 和 SITEGPASS 命令进入 具有权限的组中。通过 private 设置支持或不支持这一功能,其格式如下:

private <switch>

<switch>YES (支持 private 功能)或 NO (不支持 private 功能)。

2. 控制提示消息

下面的命令将给访问服务器的用户各种信息。

· banner 命令

banner 命令在客户登录到服务器之前提供一个注册屏幕。banner 命令的格式如下:

banner <path>

这里的<path>是想要显示的文件的全路径名。在/home/ftp/中有一个.banner文件。

· email 命令

email 命令用来给出服务器管理员的 email 地址。email 命令的格式如下:

email <address>

<address>是服务器管理员的 email 地址。

• message 命令

当用户进入系统或进入到某一目录时, message 命令发送消息给客户。message 命令的 格式如下:

message <path> <when> {<class>}

<path>是将显示的文件的路径名,<when>是显示消息的条件;<class>是这一命令所应
用到的类的列表。

<when>有两种形式:LOGIN和CWD=<dir>。如果用LOGIN,则用户注册成功后显示 指定的消息;如果用CWD=<dir>,则用户进入<dir>目录时显示消息。

· readme 命令

readme 命令在一定的条件下通知用户当前目录下的某个文件被最后修改。该命令采用的格式如下:

readme <path> <when> <class>

<path>用来通知用户应该注意的文件;此处的<when>和 message 命令中的<when>功能 相同;<class>是该命令应用的类。参数<when>和<class>可选。

3. 控制日志记录

下面的命令用来确定服务器软件应该记录多少日志。

· log commands 命令

log commands 命令记录了 FTP 用户执行的命令。用户所调用的每一个命令都将被记录 到日志文件中。该命令的格式如下:

log commands <typelist>

这里的<typelist>是一个由逗号分开的列表,指定记录哪种类型的用户。3 种可以识别的用户为 anonymous、guest 和 real。

• log transfers 命令

log transfers 命令只对客户的文件传输作日志记录。该命令格式如下:

log transfers <typelist> <directions>

这里的<typelist>同前面的解释相同;<directions>也是一个由逗号分开的列表,用来指 定应该记录哪种方向的传输,可以选择的两个方向是 inbound 和 outbound。 4. 控制权限

除了控制登录和维护日志外,还需要严格控制档案中文件的权限。下面这些命令指定 在一定条件下应该设置的权限。

· chmod 命令

chmod 命令确定用户是否可以利用自己的 chmod 命令来改变服务器上的文件权限。该 命令的格式如下:

chmod <switch> <typelist>

<switch>和<typelist>的解释与前面的相同。

· delete 命令

delete 命令用来通知用户是否可以通过 FTP 删除位于服务器上的文件。该命令的格式如下:

delete <switch> <typelist>

<switch>和<typelist>的解释同上。

• overwrite 命令

overwrite 命令控制 FTP 用户是否能够上载并替换服务器上现有的文件。该命令的格式如下:

overwrite <switch> <typelist>

<switch>和<typelist>的解释同上。

· rename 命令

rename 命令确定用户是否可以更改服务器上的文件名。该命令的格式如下:

rename <switch> <typelist>

<switch>和<typelist>的解释同上。

· umask 命令

umask 命令确定用户是否能够以 shell 命令的 umask 那样的方式更改他们的默认权限。 该命令的格式如下:

umask <switch> <typelist>

<switch>和<typelist>的解释同上。

· passwd-check 命令

passwd-check 命令确定对匿名用户的口令(一般是一个 email 地址)的检查的严格程度。 该命令的格式如下:

passwd-check <strictness> <enforcement>

这里的<strictness>是这 3 个字符串之一:none、trivial 或 rfc822。<enforcement>是这两 个可能的字符串之一:warn 或 enforce。 <strictness>选择为 none,将不对口令做任何检查;选择为 trivial 将对口令有一些要求, 在口令中至少要有一个@符号;选择为 rfc822 是最严格的,它要求 email 地址必须遵守 RFC822 "消息头标准"。

<enforcement>选择为 warn 时,如果用户没有按照<strictness>的要求提供口令,则会提 出警告,但仍然允许用户同服务器进行连接;选择为 enforce 时,将拒绝用户的连接。

• path-filter 命令

path-filter 命令对用户通过 FTP 向服务器上载文件的文件名进行一些限制。该命令的格式如下:

path-filter <typelist> <mesg> <allowed-regexp> <denied-regexp>

<typelist>的解释同前;<mesg>是一个文本文件名,当文件名不符合标准时,将显示文件;<allowed-regexp>是一个正则表达式,只有文件名符合该表达式时,才可以上载;<allowed-regexp>也是一个正则表达式,当文件名符合它时,将被拒绝。

· upload 命令

upload 命令决定用户可以在哪些目录中上载文件。该命令还决定文件放置后将具有什 么样的权限。该命令的格式如下:

upload <directory> <dirglob> <switch> <owner> <group> <mode> <mkdir>

<directory>是该命令所影响的目录;<dirglob>是一个正则表达式,确定<directory>中的哪些目录可以上载文件;<switch>是 YES 或 NO,用来确定能否在那里进行上载;当文件被放在服务器上后,参数<owner>、<group>和<mode>用来确定文件的拥有者、用户组和权限;<mkdir>可指定为 dirs 或 nodirs,表示允许或不允许在指定目录下创建子目录。

5. 服务器杂项

下面这些命令提供了各种配置杂项功能,为服务器增添了灵活性。

alias 命令

alias 命令为 FTP 客户定义目录的别名。该功能有利于为经常使用的文件提供快捷方式。 alias 命令的格式如下:

alias <string> <dir>

<string>是别名, <dir>是实际目录。

· cdpath 命令

cdpath 命令建立与环境变量 PATH 类似的路径列表。当用户调用 cd 命令时要检查该列表。cdpath 命令的格式如下:

cdpath <dir>

这里的<dir>是客户使用cd命令时检查的服务器上的目录。记住:匿名用户要使用相对 于FTP主目录的目录。

compress 命令

compress 命令允许传输之前服务器对文件进行压缩或解压。compress 命令的格式如下:

compress <switch> <c1assglob>

<switch>的解释同前; <c1assglob>是由逗号分开的一个类的列表,指定该压缩选项所 适用的类。

· tar 命令

tar 命令基本和 compress 命令相同。tar 通知服务器是否根据需要为用户存档或打开文件。该命令的格式如下:

tar <switch> <c1assglob>

<switch>和<classglob>的解释同前。

11.3.2 /etc/ftphosts文件

/etc/ftphosts 文件用来规定哪些用户能够从哪些主机访问服务器。文件中的每一行都是 下面两种形式之一:

allow <username> <addrglob> deny <username> <addrglob>

allow 命令允许<username>指定的用户通过 FTP 从<addrglob>所列的地址连接到服务器。在<addrglob>中可以列出多个地址。

deny 命令明确地拒绝<usename>指定的用户从<addrglob>所列的地址处连接到服务器。 在<addrglob>中可以列出多个地址。

11.3.3 /var/log/xferlog文件

虽然/var/log/xferlog 不是一个配置文件,但是它扮演了一个重要的角色。在该文件中存储了 FTP 服务器生成的所有日志。如表 11.2 所示是/var/lag/xferlog 字段及其定义列表。日志的每一行都是由表 11.2 中的几方面内容组成的。

日志字段	定义
current-time	格式为 DDDMMMddhh:mm:ssYYYY 的当前时间。这里的 DDD 是星期几;
	MMM 是月份;dd 是哪一日;hh:m:ss 是军用格式的时间;YYY 是年份
transfer-time	传输文件所花费的总时间,以秒为单位
remote-host	发出传输命令的客户的主机名
file-size	被传输的文件的大小
filename	被传输的文件的名字
transfer-type	执行传输的类型。这里的 a 指的是 ASCII 码传输 , b 指的是二进制传输
special-action-flag	服务器对文件执行的一些操作。这里的 C 表示文件被压缩, U 表示文件被解
	压,T 表示文件被归档;-表示没有执行任何操作
direction	表明文件是传出还是传入的标志 , 分别利用 o 和 i 表示

表 11.2 /var/log/xferlog 字段及其定义

(续表)

日志字段	定义
access-mode	执行传输的用户类型。这里的 a 指的是匿名用户 , g 指的是客人用户 , r 指的
	是真正的用户
username	当用户的类型是 real 时,它指的是本地用户名
service-name	被调用的服务的名字(最常用的是 FTP)
authentication-method	所使用的验证类型。0 表示不进行验证(匿名用户),1 表示按照 RFC931
	验证服务器协议进行验证
authenticated-user-id	传输要验证的用户名
completion-status	文件传输状态完整或者不完整

11.3.4 配置示例

1. 配置 FTP 服务访问权限

下面的步骤将配置允许哪些人登录到 FTP 服务器:

(1)首先确定允许哪些人进入 FTP 服务器。

- (2) 以根用户身份登录进入系统。
- (3) 编辑/etc/ftphosts 文件。

(4)为允许访问的主机建立 allow 语句。例如,如果想让自己的站点 colors.org 上的 全部用户都有访问权限,请使用下面的语句:

allow * *.colors.org

(5)为不允许访问的主机建立 deny 语句。例如,如果不想让来自计算机 machine1.test.com 的用户 alice 访问,就可以使用下面的语句禁止其通过 FTP 进入:

deny alice machine1.test.com

2. 配置 FTP 服务的欢迎信息

下面的步骤将修改用户使用 FTP 服务进入服务器时看到的欢迎词:

(1) 以根用户身份登录进入系统。

(2)把路径切换到/home/ftp目录,或者是其他任何想把欢迎消息添加在其中的目录。

(3) 建立一个.message 文件,在其中输入想让用户通过 FTP 服务进入这个目录时看 到的文字。

(4)保存文件并退出。

3.测试 FTP 服务的设置效果

下面的步骤测试 FTP 服务的设置效果:

(1) 随便以某个用户的身份登录进入系统。

(2) FTP 用户连接到一个配置为 FTP 服务器的计算机上。例如,如果在 colors.org 域

的 yellow 机器上已经安装好匿名 FTP 软件,就应该输入 ftp yellow.colors.org。

(3)进行上载文件操作。

(4)进行下载文件操作。

(5)进行切换目录路径操作。

11.4 FTP管理工具

有些工具可以帮助管理 FTP 服务器,这些工具在安装 wu-ftp RPM 软件包时是作为它的一部分自动安装的。它们有助于观察服务器的当前状态并控制服务器的关闭过程。

这些工具包括 ftpshut、ftpwho 和 ftpcount。

11.4.1 ftpshut命令

ftpshut 命令有助于使 FTP 服务器的关闭变得更加容易。ftpshut 命令的格式如下:

ftpshut -l <login-minutes> -d <drop-minutes> <time> <warning message>

这里的<login-minutes>是服务器拒绝新的 FTP 用户进入的时间,它指的是服务器关闭 之前的分钟数;<drop-minutes>是服务器开始断开现有连接的时间,它指的也是服务器关闭 之前的分钟数。<login-minutes>的缺省值是 10, <drop-minutes>的缺省值是 5。

<time>是服务器被关闭的时间,可以采用3种方式来指定。第1种方式是以没有冒号的格式来指定时间(例如,0312表示凌晨3:12);第2种方式是指定关闭系统之前等待的分钟数,格式是+<min>,这里的<min>是等待的分钟数(例如,+60使服务器在60分钟后关闭);最后一种方式是最紧急的方式,通过使用字符串 now 服务器立刻关闭。

<warnin gmessang>是服务器关闭时显示给所有 FTP 客户的信息,通知他们服务器将要 关闭。

11.4.2 ftpwho/ftpcount命令

ftpwho 用来显示通过 FTP 正在与系统连接的所有用户。该命令的输出格式如下:

<pid> <tty> <stat> <time> <connection details>

各个字段的含义如表 11.3 所示。

表 11.3 ftpwho 命令的字段及其含义

字段	含义
<pid></pid>	处理传输的 FTP 驻留程序的进程号
<tty></tty>	总是一个问号。因为这是 FTP 连接而不是 telnet
<stat></stat>	守护程序实例的状态。S 表示它处于睡眠状态;Z 表示它处于无效状态,R
	表示它是当前正在运行的进程
<time></time>	FTP 实例所占用的 CPU 时间
<connection details=""></connection>	显示连接的来源、用户以及用户当前所执行的操作

ftpcount 是 ftpwho 的简化版本,它显示/etc/ftpaccess 中的每个类中进入的用户总数。

11.5 使用FTP客户程序

在 Red Hat Linux 中,支持标准化的 FTP 协议的客户程序有很多,下面介绍其中有代表性的几个客户程序。

11.5.1 ftp命令

ftp 程序是支持标准化的 FTP 协议的最古老的程序之一, ftp 命令的基本使用方法是使 用 get 和 put 命令交互地收发文件, ftp 命令还有许多用户可能还不熟悉的特性。

存放在子目录/usr/bin 中的 ftp 命令有 5 个不同的命令行参数,但是它最常使用的情况 是在这个命令后面加上一个远端计算机的名称,如下所示:

#ftp ftp.mcp.com

也可以不在命令行中指定一个主机名。ftp 程序可以交互式地与不同的计算机连接和断 开连接。

ftp 命令有超过 70 个不同的内建子命令及一个帮助功能。可以阅读 ftp 程序的使用手册 页或者使用命令内建的 help 子命令来获得这些命令的详细资料。

如表 11.4 所示是一些可能会使用到的 ftp 子命令及其说明。

表 11.4 常用 ftp 子命令及其说明

命令	说明
!	用来运行一个 shell 命令的提示符。这在浏览文件内容、删除已经下载的文件或者查看
	硬盘上还有多少剩余空间的时候非常有用
ascii	以文本格式下载任何指定的文件
binary	以二进制格式下载任何指定的文件。在下载大多数经过压缩的 Linux 文件之前必须使用
	这个命令
bye	关闭任何已经打开的连接并退出 ftp 程序
cd	进入其他子目录
close	关闭当前连接
get	从当前子目录中下载一个指定的文件到当前子目录
ls	列出当前子目录中的文件和其中的子目录
mget	下载几个文件,这些文件的文件名要匹配指定的模式
open	建立一个到指定主机的连接
prompt	切换交互式提示功能。如在下载多个文件的时候提示使用 y 或 n 键确认

11.5.2 ncftp命令

ncftp 命令是一种 ftp 命令的改进。该程序提供工具帮助构造 shell 脚本以自动传输文件, 并且在命令行提供 FTP 传输的伪图形化的界面。 ncftp 命令有如下的功能:

- 自动恢复下载。
- · 后台处理。
- ・ 书签。
- · 缓冲的目录清单。
- 命令行编辑。
- · 下载整个目录树。
- 文件名补全。
- 主机重新拨号。
- 进程计量。
- · 支持防火墙和代理。

简单的情况下, ncftp 可以和 ftp 一样使用, 加上远程计算机名。与 ftp 命令一样, ncftp 命令也有一个内建的帮助功能。

使用计算机的主机名和某个文件的完整路径,可以用一条命令检索到这个文件,如下 所示:

#ncftpget ftp://ftp.ncic.ac.cn/pub/Linux/file1.tgz

#ncftpget ftp.metalab.unc.edu/pub/Linux/apps/appliance/netlitng.tgz

这两个命令行都可以连接到 ftp.ncic.ac.cn 主机,以匿名的方式登录进入,然后检索 file1.tgz 文件。这是一个从远端计算机上检索某些文件名已知的文件的好方法。

这些命令行还可以用在 shell 的命令脚本程序中,或者与管道及其他命令联合使用对处 理下载的文件做进一步的处理。例如,可以自动转换下载来的图形文件,并把它们保存到 一个指定的目录中,如下所示:

#ncftpget ncic/pub/gif/pic1.gif
giftopnm
ppmtopcx > /home/ftp/pic. pcx

这个例子下载了一个.gif 格式的图片,并把这个图像转换为.pnm 格式,再把它转换 到.pcx 格式,最后保存到系统中的 ftp 子目录中。可以使用这种方法自动下载和更新文件, 特别是把这种方法使在一个 crontab 文件中的时候。

11.5.3 gftp命令

gftp 命令和其他 FTP 客户程序一样,支持直接登录。gftp 命令通常的语法如下:

gftp [--help][-h][-version][-v][[ftp://][user:pass@]ftp-site[:port][/directory]]

该命令显示可以使用可选关键字,如 ftp://,以及用户名、口令、远程 FTP 服务器特定 的端口号和文件传输的目的目录。使用该语法可以在远程计算机上的自己的目录中启动 FTP 会话。

gFTP 客户程序有一个方便使用的界面进行 FTP 文件传输。在 Gnome 下,通过单击

GNOME Panel 上的 Internet, 然后单击 gFTP 菜单项, 可以启动该客户程序。或者在命令行下执行:

#gftp

gFTP 客户程序将启动,如图 11.1 所示。



图 11.1 gFTP 程序使用界面

gFTP 主窗口提供一个菜单栏、7 个下拉菜单、4 个用于主机和端口号等信息的文本框, 两个分别针对本地计算机和远程计算机的目录窗口,其中一个进程窗口用于显示 Filename、 Progress 和计算机的 Hostname,另一个滚动会话窗口显示当前活动。

下面介绍其中的6个菜单:

- · FTP 菜单 用于指定文件传输类型和其他选项,它的 Options 级联菜单可以设置的 选项如图 11.2 所示。
- · Local 菜单 操作文件和目录。
- · Remote 菜单 用于连接、断开连接、更改服务器和操作远程文件和目录。
- · Transfers 菜单 控制传输会话和上载下载文件。
- · Logging 菜单 用于记录 FTP 会话。
- Tools菜单 比较本地和远程目录,并高亮显示没有同时在两个窗口中发现的文件。
 该功能可以使计算机之间同步文件目录非常容易。

在连接完成以后,通过单击文件,然后在 gFTP 对话框中适当的方向箭头上单击,就 可以在本地和远程计算机之间传输文件。该操作在计算机之间发送或接收文件。gFTP 的另 一个功能是:在使用 GNOME 时支持拖放功能。该功能表明传输文件到远程计算机上,只 需要从 GNU Midnight Commander 桌面上单击文件并拖动该文件,然后在远程计算机的文件清单窗口释放该文件即可。

Options		
General FTP Pr	oxy HTTP Pro	xy Local Hosts
Ema	ail address: ro	ot@Apollo
Vie	w program:	
Ed	lit program:	
Conn	ect timeout: 0	
Re	ad timeout: 30	
Conr	iect retries: 3	
Retry	sleep time: 30	
	Max KB/S: 0.0	00
🔄 Bring up rec	onnect dialog	🗖 Confirm delete
🗖 Do one trans	fer at a time	F Passive file transfers
🗖 Preserve pe	rmissions	Refresh after each file transfer
🔲 Resolve Ren	note Symlinks	🗖 Save geometry
🗖 Show hidder	n files	🗖 Sort directories first
🗖 Start file tran	sfers	🗖 Use cache
🗖 Use HTTP/1.	1	
ОК		Cancel Apply

图 11.2 FTP 菜单的 Options 级联菜单

- 11.6 习题
- 1. 适当调整 ftp 服务器的配置参数,检查服务器的变化。
- 2. 试用 ftpshut 和 ftpwho 等 ftp 管理工具。
- 3. 试用 ncftp 和 gfpt 等客户程序上传和下载文件。

第 12章 配置 Apache 服务器

本章主要介绍 Linux 下的 Web 服务器 Apache 的配置使用。内容包括 Apache 服务器简介、安装 Apache 服务器、Apache 服务器配置文件以及配置 Apache 服务器等。

12.1 Apache服务器简介

现在,WWW 服务是互联网上人们使用最多的一种服务,它已经变成了当今信息社会 必不可少的一种信息传播方式。无论人们年龄、工作和知识水平如何 , 动态的 html 页面都 是一种宣传和获取信息最直接的方式。

那么,数以万计的 WWW 页面是建立在什么样的服务器上的呢?这个服务器上运行的 是什么操作系统呢?根据对 Web 站点的调查,在当前的 Internet 上应用最广泛的 Web 服务 器是 Apache 服务器。这种服务器在 Linux 操作系统上可以完美地运行,并且, Apache 服 务器的安装和配置都不甚艰深,经过学习,完全可以掌握。

Apache 最初是源代码开放的 NCSA HTTPd Web 服务器的补丁。许多人为 NCSA HTTPd Web 服务器编写了许多补丁,修补问题并添加功能。后来,从 NCSA 1.3 服务器版本和这些补丁中诞生了 Apache 服务器。

由于 Apache 产生于 NCSA HTTPd Web 服务器,所以 Apache 力争和 NCSA HTTPd Web 服务器兼容。现在, NCSA HTTPd Web 服务器已经略显老旧, 安装 WWW 服务器的人大都 选择 Apache 服务器。

有关 Apache 的最新消息可以在以下网站找到:

Apache group 的 Web 站点 http://www.apache.org

Apache Week 的 Web 站点http://www.apacheweek.com

关于 Apache 的全部文档还可以在 Apahce 服务器的 HTML 目录中找到。

Red Hat Linux 包括 Apache 的一个版本,但是由于 Apache 版本更新太快,提供的版本 一般都不是最新的。在 Red Hat 的 FTP 服务器上的/pub/contrib 目录中,有 RPM 安装文件 形式的 Apache。或者从 Apache Web 站点获得源代码,然后按照 Linux 惯例,自己构造该 软件。

本章后面介绍的 Apache 的 RPM 安装版本是 apache-1.3.12-25.i386.rpm, 它随 Red Hat Linux 7.0 发售。

12.2 安装Apache服务器

如果在安装 Red Hat Linux 7.0 时已经选择了安装 Web 服务器,那么系统中就已经有了

Apache 服务器。如果想了解如何安装 Apache 服务器,可以从第 12.3 节中学会。如果只想 合理地配置 Apache,可以跳过本节,直接学习配置 Apache。

在安装完 Red Hat Linux 7.0 之后,可以使用 RPM 包管理工具另行安装 Apache,或者自己从源代码编译生成该软件。Apache 源代码可以在 Linux 操作系统中顺利编译。

下面分别介绍使用 RPM 工具安装和自己编译安装 Apache 服务器的方法。

12.2.1 使用RPM工具安装

使用 RPM 工具可以很容易地安装 Apache 服务器,所需要做的工作只有简单的两件事: 找到 Apache 服务器的 RPM 安装包,然后在命令行安装 Apache RPM 安装包。

Apache RPM 安装包可以在 Red Hat Linux 7.0 的安装光盘上找到,也可以在 Red Hat 的 FTP 服务器上找到。这样,第一步就完成了。

下面用命令行形式的rpm工具安装 Apache RPM。如果在 Red Hat Linux 7.0 的 X Window 环境下,请打开一个虚拟终端,运行下面的命令;如果在 ALT-F1 到 ALT-F6 的终端下,请 直接运行下面的命令。

如果从光盘上安装 Apache RPM 安装包,那么请在终端下运行:

mnt /dev/cdrom /mnt/cdrom

cd /mnt/cdrom/RedHat/RPM

Rpm –Uvh apache_1.3.12-25.i386.rpm

命令运行成功之后, Apache 服务器即告安装完毕。 Apache RPM 将把文件安装在如表 12.1 所示的目录中。

表 12.1 Apache RPM 文件的安装位置

目录	安装的内容
/etc/httpd/conf	Apache 的所有配置文件,包括 access.conf、httpd.conf 和 srm.conf
/etc/rc.d/	系统的启动脚本。ApacheRPM 在这里安装了 Web 服务器的整套脚本,这些脚本可
	用来从命令行启动和停止服务器,并且也可在工作站关闭、启动或重新引导时自动
	启动或停止服务器。在调用时应带 start 或 stop 参数(若不带参数执行此脚本文件,
	将产生一个提示调用方法的信息)
/home/httpd	这是缺省的文档根目录。在该目录下的文件和目录是 Web 可访问的。当使用 Web
	浏览器与 Apcache 服务器连接时,该目录下的 index.html 文件是缺省的被浏览文件。
	如果想在其他地方保存 Web 内容,通过在服务器的配置文件适当的地方进行更改
	可以实现
/usr/doc	这个目录还包含了支持 Web 站点的一些可执行代码文件。这些可执行文件必须遵
	守 CGI 的规范,但是可以用多种语言来编写,包括已编译的 C/C++代码、Java、Perl、
	PHP、BASH、awk 或其他已编译的代码或解释器
/usr/man	RPM 包含手册页和 readme 文件。像大多数 RPM 软件包一样, readme 文件以及其
	他相关的文档放在/usr/doc 下的一个以服务器软件包的版本命名的目录中

(续表)

目录	安装的内容
/usr/sbin	可执行程序。包括服务器程序本身,还包含一些 Web 服务器的支撑工具,如用于
	创建验证口令文件的 htpasswd 程序
/var/log/http	这个目录包含了下面一些 Web 服务器的日志文件:access_log (记录所有与 Web 服
	务器的连接), error_log (报告运行时服务器所遇到的所有错误, 如 CGI、.htaccess
	配置错误等)。日志处理系统将定期地为日志改名或删除。在默认情况下,包含这
	两个日志文件 , 但是可以定义任意多个包含各种信息的自定义日志文件

如果要升级 Apache 到更新的版本, RPM 不会覆盖当前的配置文件。RPM 将在当前文件名后添加.rpmnew 后缀。例如, srm.conf 变为 srm.conf.rpmnew。

12.2.2 自己编译源代码安装

自己编译源代码安装 Apache 服务器也是一种很容易掌握的方法。这种情况下需要取得 Apache 服务器的源代码,这可以从ftp://www.apache.org/apache/dist取得。

从上述站点取得的源代码是 apache-xxxx.tar.gz(xxxx 代表下载的 Apache 源代码的版本 号),请将其解压缩,如下所示:

#tar -xvf apache-xxxx.tar.gz

解压缩的结果是产生一个 apache-xxxx 目录。进入这个目录,运行这个目录中的 configure 命令。该命令可以接受一个--help 参数,以供查看可配置的编译参数,如下所示:

#cd apache-xxxx #./configure --help

可以看到, configure 命令接受很多的编译参数,其中绝大部分都不用理会,只要注意 其中的--prefix 参数即可。该参数将指定 Apache 服务器安装的路径。Apache 服务器缺省的 安装路径是/usr/local/apache/目录。如果想将该软件安装到默认目录以外的地方,则需要在 运行 configure 命令时用 prefix 参数指定,如下所示:

#./configure -prefix /preferred_directory

这个命令在/preferred_directory/src/目录下创建名为 Configuration 的文件,还产生用于 编译服务器代码的 makefile 文件。

运行 configure 命令后,请输入 make 命令,然后使用 make install 命令编译服务器代码。 最后,输入/usr/local/apache/bin/apachectl start 启动 Web 服务器进程。如果使用--prefix 参数,则将/usr/local/apache/目录更改为前面自己输入的值。

整个过程用到的命令如下所示:

#tar -xvf apache-xxxx.tar.gz
#cd apache-xxxx
#./configure --prefix /preferred_directory
#cd / preferred_directory/src
#./make

#./make install

#/preferred_directory/bin/apachectl start

它列出了从源代码的压缩文件到运行 Apache 服务器的所需步骤,在正确执行之后, Apache 服务器应该已经运行起来了。

12.3 配置Apache服务器

Apache 服务器的配置主要依赖于其配置文件,所以本节将集中介绍配置文件。

由于 Apache 服务器试图和 NCSA 服务器兼容,所以它的配置文件也包括 NCSA 服务器的 3 个配置文件 access.conf、httpd.conf 和 srm.conf。

这3个文件存放在/etc/httpd/conf 目录下,它们的作用分别是:

- · httpd.conf 主要的 Web 服务器配置。
- · access.conf 访问限制和安全。
- · srm.conf MIME 与文件关联。

最初,对 Apache 服务器进行配制的不同配置项要放入不同的配置文件中,但现在,各 个配置文件的界限已经变得非常模糊,将任何配置选项放入任何文件都将生效。

所以,现在的服务器运行配置只存储在一个文件——httpd.conf文件中。其他文件仍然存在,但是只包含注释,告诉用户该文件只是由于历史原因而保留,应该将所有配置放入 httpd.conf文件。

如果打开 access.conf 和 srm.conf 文件,可以看到如下的注释:

#

This is the default file for the ResourceConfig directive in http.conf.

It is processed after httpd.conf but before access.conf

#

To avoid confusion, it is recommended that you put all of your

Apache server directives into the httpd.conf file and leave this

one essentially empty

#

本节只在 httpd.conf 文件中配置 Apache 服务器。

当Apache进程启动(或重新启动)时,该进程从配置文件中读取数据。使用httpd reload 命令可以使Apache进程重新装载配置信息。在本节后面将介绍启动和停止服务器。服务器 在运行时的配置是通过配置指令来完成的,这些指令是用来设置某些选项的命令,它们可 用来向服务器通知所要启用的各种选项。例如对服务器的配置和操作都非常重要的文件所 处的位置。

配置指令遵循如下的语法:

directive option option

每行可以指定一个指令。有些指令只用来设置如文件名之类的值;有些指令却可以指 定多个选项;另外还有一些称为节(section)的特殊指令,它们就像 HTML 的标签一样。 节指令由尖括号包围,就像<directive>这样,节通常包含一组指令,这一组指令只适用于由 节所指定的目录,如下所示:

<Directory somedir/in/your/tree> directive optiono ption directive option option </directive>

所有的节都利用像</directive>这样的与<directive>相匹配的节标签作为结束标志。节标签同其他指令一样,每行只能指定一个。

12.3.1 httpd.conf配置文件

配置文件中的大多数设置可以保持不变,特别是在默认位置安装Apache并且没有在其上做什么不寻常的事时。通常,当不理解某一指令的用途时,应让它保持缺省值。

下面介绍一些配置指令,可以更改这些配置以定制一个适合自己的服务器的配置。

ServerType 指令

ServerType指令用于定义服务器启动的方式。服务器的启动方式有stand alone(单独启动)和inetd(借助超级服务程序inetd启动)。该指令缺省的配置是standalone。如果将standalone改成inetd,那么,从这条指令到ServerAdmin指令中间的指令都可以跳过。

在单独启动的情况下,只启动一个服务器;在inetd启动的情况下,对于任何传入的HTTP 请求,将启动一个新的服务器,该服务器在请求服务完成以后立即消亡。显然,inetd启动 非常消耗资源,因为对于每个请求都有服务器启动的开销。所以,如果设想网站可能被多 人访问,最好将ServerType置为standalone。

· Port 指令

Port指令指定服务器运行在哪个端口上。标准的HTTP端口号为80。将ServerType设置为standalone时,http所占用的端口号一般设为80。端口号也可以选择其他的号码,但一定要小于1023。大于1023的端口号,普通用户没有权限使用。

· ServerRoot 指令

该指令用来设置服务器目录的绝对路径,即Apache服务器的home目录。它通知服务器 到哪里查找配置文件、错误信息和日志文件。在配置文件中所指定的资源有许多是相对于 ServerRoot目录的。如果从RPM安装,则ServerRoot指令设置为/etc/httpd,如果从源代码安 装,则为/usr/local/apache。

尽量不要将这个目录指定为NFS文件系统上的一个目录,这样会带来不必要的麻烦。

• ServerName 指令

ServerName指令用来设置服务器的主机名,它应该被设置为一个完全限定的合法的域名。例如,将之设置为www.mydomain.com而不是简单的www。

在一些情况下,服务器需要知道自己的名字,这个名字可能和实际的主机名不同。例如,主机名可能为hostname.mydomain.com,该服务器名www.mydomain.com是主机名的一

个别名。如果希望服务器的URL为http://www.mydomain.com/, ServerName能够精确地设置这个名字。

如果不设置ServerName,服务器将自行判定并设置服务器的名字。

• DocumentRoot 指令

DocumentRoot指令设置文档目录树的绝对路径,该路径是Apache提供的文件的顶级目录。也就是说,应该把想要发布的html页面放在DocumentRoot指令设置的目录下。当用户访问该Apache服务器时,看到的缺省页面将是这个目录下的index.html文件.

在默认情况下,DocumentRoot被设置为/home/httpd/html;如果是自己编译源代码安装服务器,它设置为/usr/local/apache/htdocs。

• UserDir 指令

UserDir指令定义和本地用户的主目录相对的目录。可以将公共的HTML文档放入该目录中。该指令默认的设置为public_html。

也就是说,服务器上的任何用户在自己的主目录下都能够创建名为public_html的目录, 并在在该目录下放置一个index.html文件,该页面可以通过http://servername/~username这样 一个URL访问。这里servername 是如前所述定义的服务器名,username是特定用户的名称。

• DirectoryIndex 指令

DirectoryIndex指令指明作为目录索引的文件名,它可以有多个选项。例如,当请求的 URL为http://www.mydomain.com/Directory/时,index.html文件作为目录的索引将被访问。 DirectoryIndex的作用是允许指定一个代替index.html的缺省访问页面。

通常指定多个候选文件非常有用。这样,当index.html(默认的值)找不到时,可以使用另一个文件替换。例如:

DirectoryIndex index.html index.cgi

以上介绍了一些服务器配置指令。它们是简单易懂的。如果需要进一步了解配置指令, 可以查看配置文件,其中包含大量的注释。

12.3.2 htaccess文件

.htaccess文件中会出现刚才讲过的配置文件中的任何指令。该文件在httpd.conf文件的 AccessFileName指令中指定用于进行针对单一目录的配置。

系统管理员可以指定该文件的名字和该文件可以覆盖的服务器配置。当站点有多组内 容提供者时,可以通过配置.htaccess文件控制用户对他们的空间的操作。

要限制.htaccess文件能够覆盖的内容,需要使用AllowOverride指令。该指令可以进行全局设置或者单个目录设置。要配置默认可以使用的选项,需要使用Options指令。

例如,在 httpd.conf 文件中,可以见到如下清单:

each directory to which appach has access can be configured

with respect to services and features are allowed and/or disabled

in that directory (and its subdirectory)

first, we configure the "default" to be avery restrictive set of permissions

<Directory/>

optionsFollowSymlinks

AllowOverrideNone

</Directory>

下面将介绍 Options 指令和 AllowOverride 指令。

1. Options 指令

Options 的值可以为 None、All 和任何 Indexes、Includes、FollowSymLinks、ExecCGI 或者 MultiViews 的组合。MultiViews 不包含在 All 中,必须显式指定。这些选项及其含义 如表 12.2 所示。

选项的值	含义
None	该目录没有启用任何可用的选项
All	该目录启用了所有选项,除了 MultiViews
Indexes	当 Index.html 文件或者另一个 DirectoryIndex 文件不存在时,目录中的文件列表将作
	为 HTML 页产生 , 显示给用户
Includes	该目录允许服务器侧包含(SSI)。如果允许包含但是不允许在包含中有 exec 选项,
	则可以写为 IncludesNoExec。基于安全的原因,对于没有完全控制权限的目录,如
	UserDir 目录,该选项是一个很好的选择
FollowSymLinks	允许访问符号链接到文档目录的目录。这种方法不好,不要将整个服务器全部设置
	为该选项。对某个目录可以这样设置,但是仅当有足够的理由时才这样设置。该选
	项是一个潜在的安全隐患,因为它允许 Web 用户跳出文档目录以外,并且可能潜在
	地允许用户访问文件系统的分区 , 而这些地方是不希望其他人访问的
ExecCGI	即使该目录不是 ScriptAlias 化的目录,也在其中允许 CGI 程序
MultiViews	该选项是 mod_negotiation 模块的一部分。当客户请求的文档没有找到时,服务器试
	图计算最适合客户请求的文档

表 12.2 Options 选项的值及其含义

2. AllowOverrides 指令

AllowOverrides指令指定.htaccess文件可以覆盖的选项。它可以对每个目录进行设置。 例如,可以对主要文档root和UserDir目录的覆盖有不同的标准。

该功能对于用户目录特别有用。在这些目录中用户没有访问主服务器配置文件的权限。 AllowOverrides可以设置为All或者Options、FileInfo、AuthConfig以及Limit选项的组合。这 些选项及其含义如表12.3所示。

选项的值	含义
Options	.htaccess 文件可以为该目录添加没有在 Options 指令中列出的选项
FileInfo	.htaccess 文件包含修改文档类型信息的指令

表 12.3 AllowOverrides 选项的值及其含义

		(1121)
选项的值	含义	
AuthConfig	.htaccess 文件可能包含验证指令	
Limit	.htaccess 文件可能包含 allow、deny、order 指令	

12.3.3 虚拟主机

Web服务器提供的更广泛的服务之一是它可以作为一个虚拟域的主机,也称为虚拟主机。这是一个完整的Web站点,它有自己的域名,就像是单机,但是其实它和其他Web站点在同一台计算机上。例如,Linux系统中的eth0以太网接口已经配置了IP地址159.226.43.1,还可以使eth0对IP地址159.226.43.5也同样作出响应,这就是建立虚拟主机的思想。Apache通过在httpd.conf文件中设置指令实现了该功能。

在Apache服务器上设置虚拟主机有两种方法。一个方法是使用一个IP地址多个CNAME (Computer Name),另一个方法是多个IP地址,每个地址有一个名字。Apache有不同的指 令集处理这些选项的每一种。

1. 基于 IP 地址的虚拟主机

假设以太网接口已经配置了 IP 地址 159.226.43.1,使用如下命令就可以使它对 IP 地址 159.226.43.5 也作出响应:

#/sbin/ifconfig eth0:1 159.226.43.5

#/sbin/routeadd -host 159.226.43.5 dev eth0:1

显然,在使用这两条命令之前必须确认网络中的其他主机没有使用 IP 地址 159.226.43.5。然后应该把这些命令加入到系统的启动脚本中。但是要注意的是,它们应该 在主要的 if config 命令之后执行。虚拟接口的次序如下:从 eth0:1 开始,继之 eth0:2、 eth0:3、 eth0:4.....。这样各个独立的 IP 地址就与对应的虚拟接口相关联。在一个网络接口上允许 的最大虚拟 IP 地址数为 255。一旦在网络接口上添加了新的 IP 地址,对应地应该加入一条 新的 DNS 入口记录,该记录中包含有指向新入口的地址。

一旦使用多个 IP 地址配置了 Linux 计算机, Web 服务器必须为每一个 IP 地址配置不同的文档根目录。在 Apache 服务器中通过为每一个接口在/etc/httpd/conf/httpd.conf 文件的结尾处加入如下的一段代码,为每个希望成为独立的 Web 站点的地址设置 VirtualHost 指令,就可以实现上述动作的初始化过程:

ServerAdmin webmaster@username.com

DocumentRoot /home/username/public_html

ServerName www.username.com

ErrorLog logs/username-error_log

TransferLog logs/username-access_log

RefererLog logs/username-referer_log

AgentLog logs/username-agent_log

</VirtualHost>

(续表)

<virtualHost 159.226.43.5>

为了真实反映用户的 Web 站点配置 , 上述清单中的一些行必须更改。通常需要被定制的内容包括:文档根目录、服务器名称和各种日志文件。

Apache的文档中建议对上述代码中的VirtualHost指令使用IP地址,而不是主机名。若 Apache服务器启动时,DNS服务器因为某种原因不可用,则VirtualHost指令解析失败,从 而引发整段代码被忽略而没有执行。使用IP地址就可以避免这种潜在的错误。

在<VirtualHost>标签中可以指定任何配置指令。例如,可能需要为虚拟主机设置与主服务器不同的AllowOverrides指令。任何没有指定的指令默认设置为主服务器的设置。在 VirtualHost 部 分 不 能 设 置 的 指 令 为 ServerType 、 StartServers 、 MaxSpareServers 、 MinSpareServers 、 MaxRequestsPerChild 、 BindAddress 、 Listen 、 PidFile 、 TypesConfig 、 ServerRoot和NameVirtualHost。

关于Linux下虚拟主机的完整描述可以在IP-Alias-HOWTO中找到。

2. 基于名字的虚拟主机

基于名字的虚拟主机允许在同一个IP地址上运行多个主机。需要对DNS添加额外的名 字作为有关的计算机的CNAME。当HTTP客户(浏览器)从服务器上请求文档时,随请求 一起发送一个变量指明向哪个服务器请求文档。基于该变量,服务器决定从哪个虚拟主机 提供文档。

基于名字的虚拟主机只需要比基于IP地址的虚拟主机多进行一个设置。首先需要指明 哪个IP地址有多个DNS名字。使用如下的NameVirtualHost指令实现该功能:

NameVirtualHost 159.226.43.1

然后需要为该地址上的每个名字建立一个小节,设置该名字的配置。对于基于 IP 的虚 拟主机,只需要设置和主机不同的配置。但是在基于名字的虚拟主机,必需设置如下的 ServerName 指令,因为这是一个主机和另一个主机唯一的区别。

<VirtualHost 159.226.43.1>

ServerName server1.mydomain.com

ServerAlias /home/server1/htdocs

ScriptAlias /home/server1/cgi-bin

TransferLog /home/server1/logs/access_log

</VirtualHost>

<VirtualHost 159.226.43.1>

ServerName server2.mydomain.com

ServerAlias server2

DocumentRoot /home/server2/htdocs

ScriptAlias /home/server2/cgi-bin

TransferLog /home/server2/logs/access_log

ErrorLog /home/server2/logs/error_log

</VirtualHost>

如果在 Intranet 或者 Internet 上设置 Web 站点,就可以经常使用该计算机的缩写而不是

全限定的域名。例如,可能在浏览器的地址字段输入 http://server1/index.html,而不是 http:// server1.mydomain.com/index.html。在这种情况下, Apache 将识别不出这两个地址指向同一 个虚拟主机。通过为 server1 和 server1.mydomain.com 都设置 VirtualHost 指令可以解决这一 问题。解决问题的简单方法是使用 ServerName 指令,该指令列出该计算机所有有效的别名:

ServerAlias server1

12.3.4 启动和停止服务器

到目前为止, Apache 服务器的安装和配置已经介绍完了, 现在可以启动它了。

1. 手动启动服务器

Apache 服务器 httpd 有几个命令行选项,可以使用这几个选项设置一些默认值,指定 httpd 从哪里读取配置指令。Apache httpd 能够识别如下的选项:

Httpd [-d directory] [-f file] [-C "directive"] [-c "directive"] [-V] [-V] [-h] [-l] [-L] [-S] [-t]

- -d 覆盖 ServerRoot 目录的位置,它把 ServerRoot 变量的初始值即 Apache 服务器 被安装的目录改变为指定的任何路径。ServerRoot 的默认值通常从 httpd.conf 配置 文件中的 ServerRoot 指令处读取。
- · -f 指定主配置文件 conf/httpd.conf 的位置。在启动时,服务器将读取并执行在 ConfigurationFile 中发现的配置命令。如果 ConfigurationFile 不是一个绝对路径(不 是以/开始),那么它的位置将被假定为相对于 httpd.conf 配置文件中的 ServerRoot 指令所指定的路径。在默认情况下,这一值通常设置为 ServerRoot/conf/httpd.conf。
- · -v 用来显示 Apache 服务器的开发版本并终止这一进程。
- · -V 显示当服务器编译时所有有效的设置。
- · -h 输出关于服务器用法的信息。如下所示:

Usage : Httpd [-d directory] [-f file] [-C "directive"] [-c "directive"]

[-v] [-V] [-h] [-l] [-L] [-S] [-t]

Options:

-D name : define a name for use in < If Define name > directives

-d directory :specify an alternate initial ServerRoot

-f file :specify an alternate ServerConfigFile

- -C "directive" : process directive before reading config files
- -v :show version number
- -V :show compile settings
- -h :list available command line options(this page)
- -l :list compiled-in modules
- -L :list available configuration directives
- -S :show parsed settings(currently only vhostsettings)
- -t :run syntax test for configuration files only

- · -1 列出编译到 Apache 服务器中的模块。
- · -L 列出可以使用的模块中可以使用的所有配置指令。
- · -S 列出服务器的虚拟主机设置。
- · -t 对配置文件进行语法检测。一旦更改了配置文件,在重新启动服务器之前进行 这种检测非常有意义。

当从命令行手动启动服务器时,需要作为 root 用户。这样做是因为如下两个原因:

如果单机服务器使用默认的HTTP端口(端口号80),只有超级用户才能够绑定低于 1025的Internet端口。

只有root拥有的进程才能够更改由User和Group指令指定的UID和GID。如果在另一个UID下启动服务器,则运行时只拥有启动它的用户的权限。

2. 使用 httpd 脚本控制 Apache 服务器

Red Hat Linux使用/etc/rc.d目录中的脚本控制各种服务的启动和关闭,包括Apache Web服务器。Apache Web服务器安装的主要脚本是/etc/rc.d/init.d/httpd。

/etc/rc.d/init.d/httpd是一个shell脚本 和/usr/sbin中的Apache服务器不同。即/usr/sbin/httpd 是一个可执行的程序文件,而/etc/rc.d/init.d/httpd是一个shell脚本,它帮助控制该程序。 可以使用以下选项控制 Web 服务器:

- · start 系统使用该选项在引导时启动该 Web 服务器。
- · Stop 系统使用该选项安全地停止该 Web 服务器。应该使用该脚本,而不是 kill 命令停止服务器。
- · Reload 在进行更改以后使用该选项给 httpd 服务程序发送 HUP 信号,使之重读配置文件。
- · Restart 该选项是停止并立即启动该 Web 服务器的简便途径。
- Status 该选项指明服务器是否正运行,如果正在运行则提供该服务器的所有实例的 PID。例如,要检查当前状态,输入命令:/etc/rc.d/init.d/httpd status 该命令输出:

httpd (pid 8643 8642.....) is running...

结果表明该Web服务器正在运行。实际上当前有多个服务器在运行。

12.4 示例

下面的配置文件是著者所用的 httpd.conf 文件的节选,限于篇幅,删除了以"#"开头的说明注释行,并且在格式重复多次的段落之中选用了段落的开头和结尾,删除了中间的部分。

这个示例配置文件中包含许多前面没有介绍的行,它们的含义可以从名字上了解。一 般情况下,这些行不需要用户修改。 ServerType standalone ServerRoot "/etc/httpd" LockFile /var/lock/httpd.lock PidFile /var/run/httpd.pid ScoreBoardFile /var/run/httpd.scoreboard Timeout 300 KeepAlive On MaxKeepAliveRequests 100 KeepAliveTimeout 15 MinSpareServers 5 MaxSpareServers 20 StartServers 8 MaxClients 150 MaxRequestsPerChild 100 Listen 80

LoadModule vhost_alias_module modules/mod_vhost_alias.so

.....

#此处有删节 LoadModule setenvif_module modules/mod_setenvif.so

<IfDefine HAVE_PERL> LoadModule perl_module

modules/libperl.so

</IfDefine>

.....

#此处有节略 <IfDefine HAVE_SSL> LoadModule ssl_module modules/libssl.so </IfDefine>

ClearModuleList AddModule mod_vhost_alias.c

#此处有节略 AddModule mod_setenvif.c

<IfDefine HAVE_PERL> AddModule mod_perl.c </IfDefine>

..... #此处有节略 <IfDefine HAVE_SSL> AddModule mod_ssl.c

</IfDefine>

Port 80 User apache Group apache ServerAdmin root@localhost ServerName Apollo DocumentRoot "/var/www/html" <Directory /> Options FollowSymLinks AllowOverride None </Directory> <Directory "/var/www/html"> Options Indexes Includes FollowSymLinks AllowOverride None Order allow, deny Allow from all </Directory> UserDir public_html DirectoryIndex index.html index.html index.shtml index.php index.php3 index.cgi AccessFileName .htaccess <Files ~ "^\.ht"> Order allow, deny Deny from all </Files> UseCanonicalName On TypesConfig /etc/mime.types DefaultType text/plain <IfModule mod_mime_magic.c> MIMEMagicFile /usr/share/magic </IfModule> HostnameLookups Off ErrorLog /var/log/httpd/error_log LogLevel warn LogFormat "%h %l %u %t \"%r\" %>s %b \"% {Referer }i\" \"% {User-Agent }i\"" combined LogFormat "%h %l %u %t \"%r\" %>s %b" common LogFormat "% {Referer }i -> % U" referer LogFormat "% {User-agent}i" agent CustomLog /var/log/httpd/access_log common ServerSignature On Alias /icons/ "/var/www/icons/"

<Directory "/var/www/icons"> Options Indexes MultiViews AllowOverride None Order allow, deny Allow from all </Directory> ScriptAlias /cgi-bin/ "/var/www/cgi-bin/" <Directory "/var/www/cgi-bin"> AllowOverride None Options ExecCGI Order allow, deny Allow from all </Directory> AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip #此处有删节 AddIcon /icons/blank.gif ^^BLANKICON^^ DefaultIcon /icons/unknown.gif ReadmeName README HeaderName HEADER IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t AddEncoding x-compress Z AddEncoding x-gzip gz tgz AddLanguage en .en AddLanguage fr .fr AddLanguage de .de AddLanguage da .da AddLanguage el .el AddLanguage it .it LanguagePriority en fr de <IfModule mod_php4.c> AddType application/x-httpd-php .php4 .php3 .phtml .php AddType application/x-httpd-php-source .phps </IfModule> <IfModule mod_php3.c> AddType application/x-httpd-php3 .php3 AddType application/x-httpd-php3-source .phps </IfModule> <IfModule mod_php.c> AddType application/x-httpd-php .phtml </IfModule> AddType application/x-tar .tgz AddType text/html .shtml

AddHandler server-parsed .shtml BrowserMatch "Mozilla/2" nokeepalive BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0 BrowserMatch "RealPlayer 4\.0" force-response-1.0 BrowserMatch "Java/1\.0" force-response-1.0 BrowserMatch "JDK/1\.0" force-response-1.0 <IfModule mod_perl.c> Alias /perl/ /var/www/perl/ <Location /perl> SetHandler perl-script PerlHandler Apache::Registry Options +ExecCGI </Location> </IfModule> Alias /doc/ /usr/share/doc/ <Location /doc> order deny,allow deny from all allow from localhost Options Indexes FollowSymLinks </Location> <IfDefine HAVE_SSL> Listen 443 <VirtualHost _default_:443> DocumentRoot "/var/www/html" SSLEngine on SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.key <Files ~ "\.(cgi|shtml)\$"> SSLOptions +StdEnvVars </Files> <Directory "/var/www/cgi-bin"> SSLOptions +StdEnvVars </Directory> SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown CustomLog /var/log/httpd/ssl_request_log \ </VirtualHost> </IfDefine>

12.5 习 题

- 1. 阅读 httpd.conf 文件。
- 2. 适当修改 httpd.conf 文件,观察服务器有什么变化。
- 3. 学习配置虚拟主机。

第13章 配置邮件服务器

本章主要介绍 Linux 下的邮件服务器的安装、配置和使用。内容包括电子邮件简介、 配置 sendmail 服务器、设置 POP3、邮件客户程序的 elm 和 pine 的使用等。

13.1 电子邮件简介

电子邮件也称为 email,是 Internet 最有用的应用程序之一。计算机网络诞生自从以来, 其主要用途便是收发电子邮件。

最初的电子邮件是一项简单的服务,它把一台机器上的文件复制到另一台机器,并把 它添加到接收端的邮箱文件内。现在,尽管路由要求日益复杂多样,信息量日益增长,但 电子邮件的基本原理始终没太在变化。

13.1.1 使用的协议

在 Internet 初期出台了许多 RFC 标准。许多 RFC 标准和电子邮件标准相关,现在仍是 如此。Internet 上的站点采用的标准包含在 RFC-822 中,后来又增加了一些 RFC,说明如 何采用与机器无关的方式传输特殊字符等。另外一些标准则针对目前的多媒体邮件传输, 用于处理包含在多媒体邮件中的图像和声音。

通常,邮件的编写是用 mail 和 mailx 之类的邮件程序接口来完成的,或者使用更为复 杂的邮件程序接口如 elm、mush 和 pine。这类程序称为"邮件用户代理",简称 MUA。如 果需要发送一条电子邮件消息,这个接口程序大多会将其交给另一个程序进行投递。这就 是所谓的"邮件传输代理",简称 MTA。在有些系统上,本地投递和远程投递采用的邮件 传输代理是不一样的;有些系统则采用同一个代理。

当然,本地邮件的投递不仅仅是把收到的消息添加到收件人的邮箱内。通常情况下,本地 MTA 可以识别别名(设置本地收件地址,使其指向另一个地址)和转发邮件(将用户的邮件重新定向到其他目的地)。另外,无法投递的邮件则必须返回,也就是说,随错误消息一起返回发件人。

对于远程邮件的投递,所用的传输程序和链接的属性有关。如果邮件必须在一个使用 TCP/IP 的网络上传递,则使用 SMTP(简单邮件传输协议)。SMTP 通常直接链接到发件人 的计算机,和远程主机的 SMTP daemon 程序协商消息的传输。

SMTP 规范最初开始于 1980 年的邮件传输协议(MTP),包括 1981 年的简单邮件传输 协议(SMTP),进而增强成为现在的协议。SMTP 的定义见 RFC-788 和 RFC-821。

13.1.2 邮件消息

邮件消息一般由消息主体(发送者写的文本内容)和指定接收者、传输载体等特殊数

据组成。

管理性数据分为两类。第一类数据是指与传输媒体相关的所有数据。例如发送人和收 信人的地址。因此,这类数据也被称为信封。它可以随邮件消息的投递通过传输软件进行 转换。第二类数据是处理邮件消息所需的所有数据,它不是某个传输机制特有的,例如, 消息的主线、收信人清单和发送消息的数据等。许多网络中,都将其加入邮件消息内形成 所谓的邮件头。

现在,许多邮件传输软件都采用 RFC-822 中定义的邮件头格式。由于被设计成不依赖 于任何一个环境,所以稍作改动,它就可以适用于其他网络。

邮件头由若干行组成,中间用一个新行字符分开。一行又由一个字段名(从第一列开 始)字段本身、冒号和空格字符或制表符组成。字段名不同,相应的字段格式和含义也会 有所不同。如果下一行以标签开头,头字段就可能继续出现在下一行。

通常,所有必要的头字段都是由使用的接口生成的,如 elm 和 pine 等。

但是,有些字段是可选的,也可以是添加的。如 elm 允许对部分消息头进行编辑。其他的则由邮件传输软件添加。如表 13.1 所示是部分常见的头字段及其含义。

字段	
From	发件人的邮件地址和真名
То	收件人的邮件地址
Subject	简短的几句话说明邮件内容
Date	发件日期
Reply-To	可选字段用于指定地址 ,发件人要求收件人直接回复到这个地址。 如果有若干个账户 ,
	但常用的只有一个时,这个字段便特别有用
Organization	可选字段用于指定发邮件的计算机的所属公司。如果是私有的,可以令其为空,或插
	入"私用"
Message-ID	发件系统上邮件传输生成的一个字串。是该邮件消息独有的
Received	对邮件进行处理的每个站点(包括发件人和收件人的计算机)都会将这一字段插入邮
	件头 , 给出其站点名、消息 ID 以及它收到该消息的时间和日期、该消息来自的站点和
	所使用的传输软件等等
X-anything	假如邮件头以 X-开头,那么任何邮件相关程序都不进行严格的检查

表 13.1 常见头字段及其含义

13.1.3 邮局协议

随着大型异构型网络的出现,在网络客户上工作并连接到邮件服务器发送和接收邮件 的邮件阅读器被邮局协议(POP)取代。POPRFC 自发布以来进行了许多次修订。现在的 协议版本为 POP3。POP3 邮件阅读器非常繁荣,在几乎任何想象得到的平台上都有其客户 软件。服务器软件不仅在各种 UNIX(包括 Linux)上已经实现,而且在其他操作系统平台 也已实现。

POP3 有许多出色的共享限制。主要的一个是只使用一个邮箱访问邮件,通常是该用户 在服务器上的传入邮箱。例如,当使用 Linux 上的客户程序(如 Pine 或者 Elm)阅读邮件 时,可以创建文件夹管理邮件。当使用 POP3 客户程序时,也可以创建文件夹管理邮件, 但是该文件夹只能存在于该 POP3 客户程序运行的计算机上。例如,如果在 Windows 计算 机上运行邮件客户程序通过 POP3 访问 Linux 服务器上的邮件,可以将邮件保存在文件夹 中,但是通常这些文件夹在 Windows 硬盘上。如果使用另一个有兼容 POP3 客户程序的 PC 访问电子邮件,则不允许访问在第一台 Windows 计算机上的这些文件夹。

13.1.4 访问协议

作为 POP3 的提高,开发了一个称为 Internet 邮件访问协议(IMAP)的协议。它的第 一个 RFC 基于版本 4,因此通常作为 IMAP4。IMAP4 克服了 POP3 的一些限制因素,主要 的特点是在服务器上可以有多个文件夹保存其阅读的邮件。因此只要使用 IMAP 访问邮件, 对以前阅读和修改的所有邮件都具有完全控制权限。

POP3 协议的另一个限制是在邮箱中不保留邮件的状态(不同的邮件应该有不同的状态),如已读、未读或者标记为删除。因此大多数 POP3 客户程序下载邮箱中的所有邮件。 IMAP4 克服了该问题,只下载所有邮件项的消息头,并且根据所选的邮件下载特定的邮件。 IMAP4 服务器可以在当前大多数主要的网络操作系统上运行。

13.2 配置Linux邮件服务器

sendmail 服务器是一个复杂的系统,有很多概念需要介绍。配置 sendmail 服务器是一项艰巨的任务,不到万不得已不要自行配置。本节介绍的有关安装 Sendmail 服务器和服务器配置的内容请斟酌选读。

在电子邮件协议迅速改变期间,一个作为邮件传输标准的软件包应运而生,这就是著 名的 sendmail。该软件由加州大学伯克利分校(U.C Berkeley)编写,它适合于大量的平台, 可以在 Red Hat Linux 上运行。

sendmail 程序为 Linux 提供支持 SMTP 连接所需的服务。它对邮件消息进行分析并把 它传送到目的地。对于普通的使用者,其副作用是它的复杂性。

sendmail 采用开放源码的开发方式编写,其所有源码都可以免费得到并且自由发布。 sendmail 与其他邮件传输代理(MTA)不同的核心功能之一是将邮件路由、邮件发送以及 邮件阅读分开。

sendmail 只提供邮件路由功能,将发送留给管理员可以选择的本地代理。这也表明用 户可以选择自己喜爱的邮件阅读器,只要该阅读器能够阅读由发送软件编写的消息格式。 对于大型的站点,sendmail 的配置方案非常灵活,它的路由功能表现出色,因而是邮件服 务器的首选。

sendmail 支持一个必须自定义的配置文件集。除了能使邮件子系统正常运行的基本信息外(如主机名),还有大量需要调整的参数。对于普通用户, sendmail 的主要配置文件比较难以理解。

同任何大型软件包一样, sendmail 也有一些缺陷。虽然引起 sendmail 失败或系统崩溃的错误已基本被完全排除,但提供 root 访问权限的安全性漏洞仍然时有发现。

13.2.1 sendmail 的工作情况

要想更好地了解如何设置 sendmail,需要知道它所做的各种工作以及这些工作如何才能适合 MUA、MTA、邮件路由器、最终投递代理以及 SMTP 客户程序和服务程序的方案。 sendmail 可以作为邮件路由器、SMTP 客户程序和 SMTP 服务程序。不过,它不执行最终的邮件投递。

1. sendmail 作为邮件路由器

sendmail 主要是一个邮件路由器。这就是说,它获取信件、检查收件人地址并确定发送信件的最好路径。sendmail 如何执行这一任务呢?

sendmail 可以确定它所需的关于自己的一些信息。例如当前的时间和它运行在的主机 名。但它的大部分工作是由邮件管理员通过配置文件 sendmail.cf 的形式来提供的。这一文 件确切地告诉 sendmail 如何处理各类邮件。该文件功能非常灵活和强大,并且看起来似乎 难以理解。但是, sendmail 版本 8 的特性之一是具有一套模块化的配置文件构造块。大多 数站点都能够利用这些模块轻易地构造出配置文件,并且还包括许多例子。从头开始编写 配置文件是一项难度较大的任务,因此应该尽量避免这么做。

2. sendmail 作为一个 MTA

像前面提到的那样, sendmail 可以作为一个 MTA 运行,因为它理解 SMTP 协议。由于 SMTP 是一个面向连接的协议,因此客户程序和服务程序总是存在。SMTP 客户程序可以 向 SMTP 服务程序投递信件,SMTP 服务程序将不断地监听计算机的 SMTP 端口。sendmail 可以是一个 SMTP 客户程序,也可以是一个 SMTP 服务程序。当 MUA 运行 sendmail 时, 它成为一个 SMTP 客户程序,并把客户端 SMTP 通知给 SMTP 服务程序。当 sendmail 以守 护程序模式启动时,它以服务程序方式运行。在服务程序方式, sendmail 完成以下两件工 作:

不断地监听计算机 SMTP 端口的传入邮件。

管理仍然没有投递的邮件队列并定期尝试重新投递这些邮件,直到邮件最后成功投 递出去或者超出重试次数。

3. sendmail 不作为最终投递代理

sendmail 不进行邮件的最终投递,它把这一任务留给了其他程序。Sendmail 是一个大型的、复杂的以及利用超级用户权限运行的程序,因此需要有一些方法来解决安全性的问题, 有关安全性的问题在 sendmail 的历史上已经发生过许多次。邮件投递的额外复杂性是 sendmail 有待解决的最后一个问题。

13.2.2 安装sendmail

安装 sendmai 要经过如下步骤:

(1) 安装包含 sendmail 的 RPM 软件包以及 sendmail.cf 软件包。该软件包安装构造用 户自己的配置文件所需的所有文件。

(2)选择一个同当前站点要求接近的 sendmail.cf 文件并对它进行必要的加工。

- (3) 使用 make 工具构造 sendmail.cf 文件。
- (4)测试 sendmail 和它的配置文件。
- (5) 安装 sendmail、sendmail.cf 以及其他辅助文件。

这是基本步骤。还要确认 sendmail 配置在当系统重新引导时能正确地启动。

最简单的实现方法是作为 root 运行 control-panel 程序。然后选择第一个图标。这是运行级编辑器,如图 13.1 所示。



图 13.1 Red Hat control-panel



图 13.2 Runlevel Manager

单击 Runlevel 编辑器,出现如图 13.2 所示的许多运行级。确认在需要 sendmail 的所有运行级中都有该程序。在默认情况下,每个运行级中均有 sendmail。如果没有,单击最左边的窗口,向下滚屏到 sendmail,选择该程序,单击 Add 并选择 Runlevel3(现在只在运行

级3加入该程序)。

除此之外,如果系统没有别名文件,还必须创建一个。在 Red Hat Linux 7.0 系统下, 该文件为/etc/aliases。别名文件的位置在 sendmail.cf 中提供,因此可以把它放在需要的任何 地方。sendmail 将所有辅助的文件放入/etc/mail 目录中。

1. 获取源代码

随 Red Hat Linux 7.0 发布的是 sendmail 8.11.0-8.i386.rpm。

经常查看 comp.security.announce 新闻组可以获得所有的新版本,提高 sendmail 的安全性。最新版本的 sendmail 在 http://www.sendmail.org 中可以找到。当发现安全漏洞时,新的版本或者补丁立即推出。因此在新版本发布以后,应该尽快升级自己的系统。明智的做法是下载包含新补丁的 Red Hat RPM。可以从 http://updates.redhat.com/6.0/i386 站点获取 Red Hat RPM 的更新。

2. 解压源代码并编译 sendmail

如果有一个新版本的 sendmail 还没有作为 RPM 发布,通过阅读本节可完成更新。现 在假定已经从 http://ftp.sendmail.org 下载了源文件。在获得了源代码后,需要把它解压。由 于它是一个被压缩的 tar 映像文件,因此首先必须对它进行解压,然后从 tar 档案中解压出 单独的文件。完成该功能的命令如下:

#tar zxvf sendmail.xxxxx.tgz

现在,可以准备编译 sendmail 了。首先要阅读下面的文件,这些文件包含关于所下载的 sendmail 版本的最新消息。

RELEASE_NOTES KNOWNBUGS READ_ME

还可能需要检查 http://www.sendmail.org/faq 中的 sendmailFAQ。这是一个有用的信息 源,它描述有关普通配置错误或者如何设置特定配置的问题。

另外,还要注意 doc/op 子目录中的 Sendmail Install ation and Operation Guide (SIOG)。 接下来运行 cd 和 ls,查看这一源代码目录中都有哪些文件:

#cd sendmail/src #ls

sendmail 的全体开发人员已经做了一些非常有意义的工作,从而使得 sendmail 的安装 过程变得非常简单。要编译新的 sendmail 版本,只需运行下面的命令(从 sendmail src 目 录)并查看其构造情况即可:

#./build

可以看到 sendmail 被编译了。

该命令创建包含编译结果的 obj.*目录。要进行更详细的配置,可以查看/BuildTools/Site 和/BuildTools/OS 目录。

在安装新的 sendmail 配置以前,一定要为将要替换的所有文件创建一个备份,特别是 原来的 sendmail 守护进程。如果新的 sendmail 不能正常工作,在排除新版本的故障时,将 需要恢复原来的版本。

要安装新版本的 sendmail,首先利用下面的命令停止当前运行的守护进程:

#/etc/rc.d/init.d/sendmail stop

然后输入以下命令:

#./build install

当所有这些步骤都完成后,便可以利用下面的命令重新启动新的 sendmail 守护进程:
#/etc/rc.d/init.d/sendmail start

13.2.3 sendmail的配置文件

sendmail 依靠许多辅助文件来执行它的工作。最重要的文件是别名文件和配置文件 sendmail1.cf。根据是否需要统计数据,决定统计文件 sendmail1.st 是否创建。统计数据还包 括以 KB 为单位的电子邮件传输总量。如果打算把 sendmail 作为 SMTP 服务程序运行,则 应该安装 SMTP 的帮助文件 sendmail.hf。

主机上需要的另一个文件是 sendmail.cw。该文件包含所有可作为电子邮件服务器的主机名。例如,如果主邮件服务器为 mail.whizzer.com,指定为 whizzer.com 的 MX,则需要将 whizzer.com 放入该文件,通知 sendmail 在 mail.whizzer.com 投递 whizzer.com 的电子邮件。

这就是关于文件 sendmail.st、serdmail.hf 以及 sendmail.cw 所需说明的内容。其他辅助 文件的说明在 Sendmail Installation and Operating Guide 中,简称为 SIOG。SIOG 通常在 /usr/doc/sendmail/doc/op/op.ps 目录或者 sendmail 源代码中。由于别名文件和 sendmail.cf 文 件非常重要,下面将详细介绍。

1.别名文件

sendmail 总是检查收件人的地址是否为别名。例如,每个 Internet 站点都需要有一个有 效地址的邮局管理员,有关邮件的问题可以向他报告。然而大多数站点并没有这一名字的 实际帐户,而是把邮局管理员的邮件转移到负责邮件管理的一个或几个人。例如,在假设 的站点 gonzo.gov 上,用户 duwei 和 mxn 共同负责邮件管理,因此别名文件会具有下面一 项内容:

postmsater:duwei,mxn

这行代码告诉 sendmail,寄向邮局管理员的邮件将被改寄到登录名为 duwei 和 mxn 处。 事实上,这些名字也可以是别名,例如:

postmaster:firstshiftops,secondshiftops,thirdshiftops firstshiftops:duwei,mxn secondshiftops:ygh,lxd thirdshiftops:xyl,wwg,lzg

在所有这些例子中,别名的名字在冒号的左边,这些名字所代替的名字在冒号的右边。 Sendmail 将反复分析别名直到它们被解析为一个真正的用户或一个远程地址为止。在 前面的例子中,要解析别名 postmaster, sendmail 首先把它扩展为收件人名单 firstshiftops、 secondshiftops 和 thirdshiftops,然后进一步扩展这些别名中的每一个,从而得到最终的名单 duwei、mxn、ygh、lxd、xyl、wwg 和 lzg。

虽然别名的右边可以指定为一个远程主机,但左边却不能。别名 duwei : duwei@whizzer.com 是合法的,而 duwei@gonzo.gov : duwei@whizzer.com 是不合法的。

只要更改了别名文件,就必须运行 newaliases。否则, sendmail 将不会知道这些更改。

2. 使用 include 指令包含别名文件

别名可用于创建邮递列表。对于大型的或经常改动的列表,可以利用:include:的别名 形式指示 sendmail 从文件中读取列表的成员。假定别名文件包含以下行:

homeboys::include:/home/alphonse/homeboys.aliases

并且假定文件/home/alphonse/homeboys.aliases 包含:

alphonse

duwei

mxn

george

其效果与下面的别名相同:

homeboys:alphonse,duwei,mxn,george

这一指令对于经常变动的邮递列表或那些由用户而非邮件管理员管理的邮递列表非常 方便。

如果发现一个用户经常要求改变邮件别名,则用户要对别的文件有读写权限。一定要 小心,sendmail 最新的版本对文件权限的定位要求非常严格。如果任何这些文件或者目录 有组或者所有人写入的权限,该文件很可能被忽略,从而导致 sendmail 停止工作或者根本 不启动。这些问题通常会在系统消息文件中记录下来。

向程序发送邮件

别名文件也可用于向某个程序发送邮件的内容。例如,通过设置许多邮递列表,可以 获取关于列表的信息或通过向某一特定地址发送信件来订阅它。这样的信件通常在它的主 体中只包含一个单词,如 help 或 subscribe。这一信件将使某个程序向寄件人邮寄信息文件。 假设 gonzo 的邮递列表具有名为 gonzo-request 的如下地址:

gonzo_request://usr/local/lib/auto_gonzo_reply

在这一形式的别名中,管道符()通知 sendmail 使用程序的邮件发送器,它通常被定 义为/bin/sh。sendmail 将把信息提供给/usr/local/lib/auto-gonzo-reply 的标准输入。通常,如 果它存在, sendmail 便考虑投递信件。

向文件发送邮件

还可以创建一个使 sendmail 向文件发送邮件的别名。这种别名以一个转发斜线 (/)开始,后面是希望添加到的文件的全路径名。下面是别名 nobody 的一个例子,在运行 NFS 的系统上经常用到:

nobody:/dev/null

指定文件的别名使 sendmail 把它的消息追加到被指定的文件中。由于这一特殊的文件 是/dev/null,因此这一别名只是把邮件扔掉。

3. sendmail.cf 配置文件

sendmail 使用了一种简便的方法自动创建 sendmail.cf 文件。建议不要手动创建或者更改 sendmail.cf 文件。

sendmail的强大在于它的灵活性,这种灵活性来自它的配置文件 sendmail.cf。sendmail.cf 文件是 sendmail 的控制中心。sendmail.cf 的语句构成了一种隐含的编程语言,这种语言有 些难以理解。

一般格式

配置文件的每一行都以一个单一的命令字符开头,这个字符用来指明这行的功能和语 法。以#开始的行是注释行;空行被忽略;以空格或制表符开始的行是上一行的续行,不 过通常应该避免续行。

功能说明

该配置文件具有以下3方面的功能:

1.可以为 sendmail 设置环境。可以设置选项以及指定需要使用的文件和数据库的位置。

2. 可以定义 sendmail 把邮件发送到那里的邮件发送器。

3. 可以指定用来改写寄件人和收件人的地址以及选择邮件发送器的规则集。

· D操作符——宏

宏类似于 shell 变量。当为一个宏定义了值后,可以在配置文件的后面对它进行引用, 宏将会被它的值所取代。宏可以多于一个字符。

sendmail 定义了几个宏,这些宏不应该被再次定义,除非是为了解决出错的系统的问题。sendmail 使用小写字母来定义自已的宏,大写字母可以自由使用。

C和F操作符——类

类与宏类似,但是在改写规则中用于不同的目的。与宏一样,类的名字也是单一的字符,小写字母由 sendmail 使用,大写字母用于由用户定义的类。一个类包含有一个或多个词。为了方便起见,大型的类可以继续到下一行。

使用宏和类最主要的原因是它们可以集中配置文件中的信息。如果不使用宏,可能一些定义散布在文件的各处,那么会在某些位置忘记对它们进行修改。另外,如果集中的是

重要的信息,可以在单独的一个位置详细地对它进行注释。

· H操作符——邮件头的定义

一般不需要对配置文件所提供的邮件头进行修改。下面是一些邮件头的示例:

H?D?Date:\$a

H?F?Resent-From:\$q

H?F?From:\$q

H?x?Full-Name:\$x

从中可以看出,在邮件头的定义中可以使用宏。当插入到某信件中时,这些宏将被展 开,由它相应的值取代。

在邮件头的定义中,可选的?mailerflag?结构用来通知 sendmail,只有在被选择的邮件 发送器具有这一邮件发送器标志时才插入邮件头。

使用?mailerflag?这一特性是因为不同的协议可能需要不同的邮件头。

另外,由于它们也需要不同的邮件发送器,因此可以通过在邮件发送器的定义中定义 合适的邮件发送器标志并使用邮件头定义中的?mailerflag?结构,来通知 sendmail 是否要插 入相应的邮件头。

· O操作符——设置选项

sendmail 具有许多选项,这些选项用来改变 sendmail 的操作或者用来通知它所使用的 文件的位置。大部分选项既可以通过命令行提供,也可以在配置文件中给出。例如,既可 以在命令行也可以在配置文件中指定别名文件的位置;要是在命令行上指定别名文件,可 以使用-o 选项,如下所示:

#/usr/sbin/sendmail -oA/etc/aliases [other arguments...]

要是在配置文件中指定别名文件,可在配置文件中添加如下一行代码:

OA/etc/aliases

这两种方法是等价的。但由于像别名文件的位置这样的选项很少被改变,因此人们通常把它们设置在配置文件 sendmail.cf 中。

· P操作符——邮件的优先级

可以在邮件中包含标题,用来指示邮件中的相对比较重要的内容。sendmail 可以利用 这些标题来确定信件的优先级。sendmail 提供的优先级如下:

Pspecial-delivery=100 Pfirst-class=0 Plist=-30 Pbulk=-60 Pjunk= - 100

V操作符——sendmail.cf的版本级别

随着 sendmail 版本 8 的发展,它的作者不断地为它添加新的特性。V 操作符通知版本

8 的 sendmail 在配置文件中可以找到哪些特性。老版本的 sendmail 不识别这一命令。在 SIOG 中,对配置文件的版本级别做了详细的说明。

· K操作符——关键字文件

sendmail 使用了一些关键字数据库,例如别名数据库。假如给定了关键字 postmaster, sendmail 将查找与这一关键词相关的数据,并返回 postmaster 的邮件应该投递到的帐号的 名字。

sendmail 把这一概念扩充为任意的数据库,包括 NIS 映射。K 操作符可用来向 sendmail 通知数据库的位置、类别以及访问方法。sendmail 支持的数据库类别有:dbm、btree、hash 和 NIS。当在 Linux 下编译时,使用的缺省类别是 dbm 格式。

· M操作符——邮件发送器的定义

邮件发送器或者是 MTA 或者是最终投递代理。前面介绍过别名文件,它可以向一个登录名、一个程序或一个文件发送邮件。针对每种目的,可以定义一个专门的邮件发送器。

邮件发送器的定义很重要,因为所有收件人的地址必须在0规则集中解析到一个邮件 发送器。解析到一个邮件发送器其实就是 sendmail 的邮件路由功能。

例如,解析到本地邮件发送器将通过定义在其中的最终投递代理(例如/bin/mail)把信件发送给某个本地用户;解析到 SMTP 邮件发送器将通过定义在其中的 sedmail 内置的 SMTP 邮件发送器把信件发送给另一个主机。

· S和R操作符——规则集和改写规则

配置文件由一系列规则集组成。规则集用来探测错误的地址,把地址改写为远程邮件 发送器能够理解的形式,以及把邮件路由到 sendmail 内部的某个邮件发送器。

sendmail 按照固有的顺序向规则集传送地址,而规则集可以不按照固有顺序调用其他规则集。固有顺序将根据被处理的地址是寄件人的地址还是收件人的地址以及选择用来投递信件的邮件发送器的不同而不同。

规则集由 S 命令声明,跟在它后面的是用来标识规则集的编号。sendmail 收集后续的 R (规则)行,直到它发现另一个 S 操作符或配置文件的结尾。

· sendmail 内置规则集的处理规则

sendmail 采用 3 段式方法处理地址:一段选择投递代理,另一段处理寄件人地址,最后一段处理收件人地址。

所有地址首先被传送到 3 号规则集进行预处理,使之变为规范的形式,以便其他规则 集能够方便地处理它们。

接下来,0号规则集将确定用于每个收件人的正确的投递代理(邮件发送器)。经过0 号规则集的地址必须解析到一个邮件发送器。

当 sendmail 在 0 号规则集中选择了一个邮件发送器后,它将通过 1 号规则集来处理寄件人的地址,然后把它们发送到由这个邮件发送器中的 S=标志所提供的规则集。

同样, sendmail 将把收件人的地址发送到 2 号规则集中进行处理, 然后再把它们发送 到邮件发送器的标志 R=所提供的规则集。

最后, sendmail 将在4号规则集中对所有地址进行后处理。4号规则集将把在3号规则

集中插入的尖括号删除。

· 规则集中的处理规则

当 sendmail 把一个地址传递给某个规则集后,规则集中的每一个规则行将一行接一行 地对它进行处理。在规则集的最后, sendmail 将把被改写的地址返回给调用这一规则集的 规则集或是固有执行顺序的下一个规则集。

· 测试规则和规则集

调试 sendmail.cf 是一项十分棘手的工作,幸运的是, sendmail 提供了一些在安装前对 规则集进行测试的方法。

- · -bt 用来通知 sendmail 进入规则测试模式。
- · -d -d 选项并不只限于与 sendmail 的地址测试模式-bt 同时使用,当发送一封信件时,还可以利用它来查看 sendmail 如何处理规则集。
- · -C 可用来在安装新的配置文件前对它们进行测试,这通常是一个很好的做法。 如果想测试一个文件,使用-C/path/to/the/file。它还可以与-bt和-d选项组合起来使用。

13.2.4 自动生成sendmail.cf文件

使用配置工具可以自动生成 sendmail.cf 文件。在生成之前,先要确认 RPM 安装包是 否安装了配置文件和宏文件。输入以下命令:

#rpm -q sendmail-cf
sendmail-cf-xxxx

如果得到类似的输出,则配置文件和宏文件已经安装在/usr/lib/sendmail-cf 目录。否则 必须从 Red Hat 光盘或者从 Red Hat FTP 站点下载相关的 RPM 并安装。

在/usr/lib/sendmail-cf/cf 目录存在一个主 Makefile 文件,它使用 m4 宏处理器从 sendmail.mc 构造 sendmail.cf 文件。mc 文件是用来编辑产生当前站点特定的 sendmail.cf 的 文件。需要检测已经安装了 m4 软件包,如下所示:

```
#rpm -q m4
```

m4-1.4.1-3

如果得到的输出与上面类似,则在系统中可以使用 m4。否则从 Red Hat 光盘上安装 m4 RPM。/usr/lib/sendmail-cf 目录内容与下面类似:

#ls sendmail-cf

README cffeaturem4ostypesiteconfigREADME checkdomainhackmailersh

README 文件包含可以添加到 mc 文件中各种特性的信息,还有其他重要信息,应该 阅读该文件。

要创建自己的 sendmail.cf 文件,请更换到 cf 目录,选择合适的模板。

cf 目录中有可以创建 cf 文件的各种 mc 文件。redhat.mc 可能是需要编辑的文件,建议

将该文件复制为另一个名字,如下所示:

#cp redhat.mc duwei.mc

下面将简要解释该样本 mc 文件的各个部分:

divert(-1) include('../m4/cf.m4')

这些是 m4 处理器处理文件需要的指令,在其他 mc 文件中可以找到类似的项目。

define('confDEF_USER_ID',"8:12") OSTYPE('linux') undefine('UUCP_RELAY') undefine('BITNET_RELAY')

define 表明需要更改 sendmail 中的设置。如允许的消息最大转发次数或者最大消息尺寸。

不同的 UNIX 操作系统对于文件放置的位置以及传递给邮件发送器的标志位有不同的 约定,这就是 OSTYPE 宏的用处。

两个 undefines 删除该 sendmail 主机接受 UUCP 和 BITNET 寻址的邮件的功能。

```
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
```

FEATURE(local_procmail)

FEATURE 宏允许添加当前站点需要的各种 sendmail 特性。

redirect特性拒绝所有到address.REDIRECT的邮件寻址,并输出一定的消息。

always_add_domain 特性总是添加本地主机的全限定域名,即使 To:地址没有限制的本地投递的邮件也需要添加。

use_cw_file 特性通知 sendmail 在文件/etc/sendmail.cw 中查找本地主机的其他可选的名称。

下面的特性指示 procmail 用作本地邮件发送器:

MAILER(procmail) MAILER(smtp)

两个 MATLER 行定义:只有两个邮件发送器, procmail 和 SMTP。

HACK(check_mail3, hash -a@JUNK /etc/mail/deny') HACK(use_ip,'/etc/mail/ip_allow') HACK(use_names,'/etc/mail/name_allow') HACK(use_relayto,'/etc/mail/relay_allow') HACK(check_rcpt4) HACK(check_relay3)

在 sendmail 中默认不使用外部主机中继。该设置停止其他主机使用自己的邮件主机作

为中继点发布广告邮件或者用于其他目的。HACK 行允许指定允许哪些主机使用自己的邮件主机作为中继点以及不从哪些主机接受邮件。

可以看到, redhat.mc 文件提供一个非常好的模板构造自己的 mc 文件。该文件已经复制为 gonzo.mc,现在所有要做的工作是再添加两项,然后创建 sendmail 可以使用的 cf 文件。 在 FEATURE 列表后面添加以下行:

FEATURE(masquerade_envolope)

 $MASQUERADE_AS(gonzo.gov)$

现在假定 Gonzo 邮件主机称为 mail.gonzo.gov。没有这些行,则所有从邮件主机发送出 去的电子邮件在信封和消息体中有 From: address,类似于 duwei@mail.gonzo.gov。最好能 够隐藏发送该邮件的主机,使接收者的 From: address 阅读到类似于 duwei@gonzo.gov 的 信息。这是 MASQUERADE_AS 行的效果。masquerade_envelope 特性使消息信封的 From: 标题产生类似加密的效果。

现在所有要做的工作是构造 sendmail 配置文件,如下所示:

#make gonzo.cf
#rm -f gondor.cf
#m4 ../m4/cf.m4 gondor.mc >gonzo.cf
#chmod 444 gonzo.cf

在输入 make 命令后可能得到一个错误消息。请输入 make-f Makefile.dist gonzo.cf 。如 果构造成功,在 cf 目录中将出现一个 gondor.cf 文件。这样就创建了自己的 sendmail.cf 文件。

13.2.5 测试sendmail和sendmail.cf文件

在安装一个新的或经过修改的 sendmail.cf 之前,必须对它进行测试。因为即使是很小的修改也可能会导致严重的后果。

测试的第一步是创建一个地址列表,这些地址应该是当前站点需要用到的有效地址。 一些需要考虑的地址包括各种别名(例如,posmaster、: include:列表、用来向文件发送邮件的别名和用来向程序发送邮件的别名等)非本地地址以及源路由地址等等。如果想更彻 底地进行测试,可以为 RFC822 中的每种合法地址格式创建一个测试地址。

当创建好测试地址的列表后,可以使用-C和-bt选项来查看将要发生的情况。这些地址 至少应该被传递给3号规则集和0号规则集,从而确定它们是否能够被路由到正确的邮件 发送器。执行这一任务的一个比较简单的方法是:创建一个包含被调用的规则集以及测试 地址的文件,然后对它执行 sendmail。例如,如果文件 addr.test 包含下面的行:

3,0 duwei

3,0 duwei@pc1.gonzo.gov

3,0 duwei@gonzo.gov

那么,可以通过输入下面的命令来测试配置文件 test.cf:

#/usr/sbin/sendmail -Ctest.cf <addr.test

rewrite: ruleset 3 input:duwei

rewrite: ruleset 3 returns:duwei

[etc.]

如果到目前为止 sendmail.cf 的工作正常,那么可以准备开始发送一些真正的信件,这 一任务可通过使用如下命令来执行:

 $\#/usr/sbin/sendmail \ \text{-v -oQ/tmp -Ctest.cv recipient} < /dev/null$

-v

用来通知 sendmail 以冗余方式显示信息,从而可以看到所发生的事情。

-0

设置一个选项覆盖 sendmail.cf 文件中的对应选项。还可以使用-O 设置长选项。

-oQ/tmp

用来通知 sendmail 使用/tmp 作为队列目录。

sendmail 也将在信件中插入标题 Apparently-to:duwei。这是因为虽然在命令行上指定 了收件人,但信件中却没有任何内容。在本例中,由于信件来自空文件/dev/null,因此没有 可用的 To:标题。收件人设为自己,可以检查邮件头的正确性。

应该重复这一过程来测试在第一次测试中使用的各种地址,或许为了某些测试还需要 创建指向自己的特殊别名。

当确信 sendmail.cf 文件正确以后,复制该文件到/etc 目录,如下所示:

#cp /etc/sendmail.cf /etc/sendmail.cf.bak

#cp gonzo.cf /etc/sendmail.cf

第一次复制将当前的 sendmail 配置备份,以备在任何需要的时候恢复原来的版本。接下来可以停止和启动 sendmail。以下命令使 sendmail 重新读取其配置文件:

#/etc/rc.d/init.d/sendmail restart

13.3 设置 POP3

在 Linux 服务器上, http 和 ftp 代理的应用非常广泛。大家都很熟悉 squid,但是在有些情况下,可能需要让 Linux 也做为 pop3 代理服务器,代理客户端从互联网上接收邮件,并传送邮件给客户工作站。

也许这种服务对于专线连到 ISP 的用户更适合。在有的公司的内部网中,需要从外部 收信,但可能不能直接与外部通信,却能很容易地访问防火墙计算机。这时,在防火墙机 器上安装 pop3 代理就起到了一种不可替代的作用。

安装 pop3 代理非常简单, 仅需要下载一个很小的程序 plug-gw 即可, 下载后放入 /usr/local/etc 目录下。

这里假设需要代理两个邮件服务器,一个是 ict.ac.cn,另一个是 pop3.263.net. 在/usr/local/etc 目录下创建存取表文件 netperm-table,如下所示:

root@sh-proxy etc plug-gw: port 110 159.226.43.* -plug-to mail.ict.ac.cn-port110 plug-gw: port 159.226.43.5 -plug-to pop3.263.net -port110 root@sh-proxy etc

第一行允许159.226.43.0 网段上任何计算机使用该代理服务器接收 mail.ict.ac.cn 上的邮件。第二行仅允许一台机器 159.226.43.5 使用该代理服务器接收 pop3.263.net 上的邮件。 修改/etc/services,加上下面的一行:

pop3proxy 114/tcp

在命令行下运行两句程序,也可加入到/etc/rc.d/rc.local的最后,使得每次启动时装载:

#/usr/local/etc/plug-gw -daemon pop3 pop3 &
#/usr/local/etc/plug-gw -daemon pop3proxy pop3proxy &

请确信/etc/services 中存在一行 pop3 110/tcp,若前面的名字不是 pop3,而是其他的如 pop-3 等,那么或者更新为 pop3 110/tcp,或者运行:

#/usr/local/etc/plug-gw -daemon pop-3 pop-3 &

若在 Windows 客户端用 outlook express 收发邮件 若从 mail.ict.ac.cn 收信 ,仅需改 pop3 服务器为用户的防火墙即可。若从 pop3.263.net 收信,还得更改标准 pop3 端口号为 114,而 不是 110。

13.4 邮件客户程序

标准的邮件客户程序有很多,本节以 elm 程序和 pine 程序为例介绍邮件客户程序的配置和使用。

13.4.1 elm程序

elm 代表 electronic mail (电子邮件), 是最理想的命名工具之一。它提供了一个非常有用的全屏接口,这是一个交互式的操作画面,可以在控制台或者某个 X11 的终端窗口的命令行上使用它。

这个程序与 pine 程序不同的地方是:elm 程序必须在一个至少有 14 行的窗口中启动, 否则它将报警并退出运行。

1. 配置 elm

未经配置的 elm 理论上可以运行。如果运气好,一切都会正常运行。但有几个选项是可能需要设置的。

启动时, elm 读取 elmrc-info 文件内的配置参数集,该文件位于/usr/lib/elm。然后,再 试着读取根目录下的.elm/elmrc 文件。通常,这个文件不需要自行编写。从 elm 的"选项" 菜单中,选定 Save 选项,便可建立它。

私用 elmrc 文件的选项集也可用于全局 elmrc-info 文件。私用 elmrc 文件中的多数设置 优先于全局文件内的设置。

全局 elmrc-info 文件中必须设置和主机名相关的选项。

只有在全局配置文件内设置它们时,这些选项才会生效;如果是在私用 elmrc 文件内 配置的,它们就会被忽略。

2. 使用 elm

当第一次启动 elm 程序的时候,它会要求回答是否想建立 elm 程序的缺省邮件文件夹 Mail,如下所示:

elm

Notice :

This version of ELM requiires the use of a .elm directory in your

home directory to store your elmrc and alias files. Shall I create the

directory .elm for you and set it up ($y\!/\!n/q$) ?

按下Y键,再按下回车键。elm 将创建.elm 目录。然后,elm 进入使用界面。如图 13.3 所示:

	Term	inal			
	File	Edit	Settings	Help	
		Mailbo	ux is '∕var	/spool/mail/root' with 0 messages [ELM 2.5 PL3]	Σ
G	You (d)elet To (To (can use e or u) read a :∎	any of th indelete ma message, p	we following commands by pressing the first characte wil, m)ail a message, r)eply or f)orward mail, q ress <return>. j = move down, k = move up, ? = hel</return>	r; Wit ₽

图 13.3 elm 的使用界面

可以使用 elm 从命令行发送消息,例如:

elm -s "How is it going ? "bball@staffnet.com < author. msg Sending mail . . . Mail to bball@staffnet.com mail sent!

可以使用 elm 程序删除、发送、保存和编写信件中的消息。在 elm 的主显示画面中敲

入一个字母"o"使用户能够对它的某些功能进行个性化设置。先按下画面左边一栏右括号前面的字母,然后输入这个项目的新设置值,比如缺省的编辑器程序(画面中的 E(ditor) Primary项目),再按下回车键。当按下"<"键之后,这些个性化设置的功能将保存在用户子目录中的.elm 文件夹的 elmrc 文件中。

13.4.2 pine程序

pine 是 Pine Is No longer Elm 的首字母缩写。pine 邮件程序保存在子目录/usr/bin 中。

使用 pine 程序不仅可以收发邮件,还可以使用它来阅读 Usenet 新闻组。下面介绍如何 配置并使用 pine 邮件程序收发电子邮件。

pine 程序带有一个非常容易使用的叫做 pico 的文本编辑器程序,它有可能成为最受欢迎的 Linux 文本编辑器程序,因为 pico 可以和任何其他程序一起使用,而不仅限于 pine 程序。

pine 程序的初始化设置和使用操作都很容易。当第一次启动这个程序的时候,就已经 自动完成了大部分的工作。当第一次使用这个程序的时候,它会显示如下所示的内容:

#pine

Creating subdirectory "/root/mail" where Pine will store its mail folders .

pine 程序启动运行并在用户子目录中建立一个名为 mail 的子目录和一个名为.pinerc 的 配置文件。pine 程序的用户界面如图 13.4 所示。

Ter	minal			
File	Edit	Settings	Help	
PINE	4,21	MAIN MENU		Folder: INBOX No Messages
		HELP		Get help using Pine
		COMPOSE	MESSAGE -	Compose and send a message
		MESSAGE	INDEX -	View messages in current folder
	L	FOLDER L	.IST –	Select a folder to view
		ADDRESS	BOOK -	Update address book
		SETUP		Configure Pine Options
		QUIT		Leave the Pine program
Cop	yright∶	1989-1999.	PINE is a tr	rademark of the University of Washington.
? Help D OTHE	R CMDS I	ListFldr	.der INBUX" (P PrevCmd s] N NextCmd	R RelNotes

图 13.4 pine 程序的用户界面

在开始编写或者发送邮件之前,需要配置 pine 程序识别用户名、ISP 商的邮件服务器 以及 ISP 商的新闻服务器。虽然可以编写消息并使用 pine 命令的 postpone 功能保存这些已 经编写好的消息, pine 程序在发送邮件时还是必须使用这些配置信息。

如果想配置 pine 程序,在上面的界面中输入 s,再输入 c,就可以进入如图 13.5 所示的 pine 程序的配置界面。

Terminal	_ = ×			
File Edit Settings Help				
PINE 4.21 SETUP CONFIGURATION Folder: INBOX No Mess	ages 🛛			
sersonal-name = KNo Value Set: using "root"> user-domain = KNo Value Set: smtp-server = KNo Value Set> nntp-server = KNo Value Set> inbox-path = KNo Value Set> inbox-path = KNo Value Set> unmort-folders = KNo Value Set> prumed+folders = KNo Value Set> default-foc = KNo Value Set> default-saved-message-folder = KNo Value Set> postponed-folder = KNo Value Set> read-message-folder = KNo Value Set> signature-folder = KNo Value Set> signature-file = KNo Value Set>				
[Composer Preferences] [] compose-cut-from-cursor [] compose-maps-delete-key-to-ctrl-d				
2 Help = Exit Setup = Prev = PrevPage = Add Value & Print = [Change Val] Next = Spc NextPage = Delete Val D WhereI	s /			

图 13.5 pine 程序的配置界面

输入用户的名字、ISP 商的域名、ISP 商的邮件服务器的名称和 ISP 商的新闻服务器的 名称。然后敲入一个字母 e, pine 程序会询问是否要保存这些设置。配置将会被保存在用 户子目录中的.pinerc 文件中。虽然也可以使用喜欢的文本编辑器程序来编辑这个文件,但 使用 pine 程序更加容易一些。

如果想编写一条消息的时候,按下字母 c 键,就进入到如图 13.6 所示的 pine 程序的信件消息写作界面。

Termir	nal					
File	Edit	Settings	Help			
PINE 4.	.21	COMPOSE ME	SSAGE		Folder: INBOX	No Messages
To :	:					
Cc : Attchmnt:	:					
Subject :	; ; ssane	Text				
	bougo					
SE Cot He	-1n	Send	🕮 Rich Hde	🐼 PeuPo/Top	🙀 fut Line	Postpope
10 Cancel	1 1	1 Del Char	^J Attach	V NxtPg/End	I 10 UnDel Line	1 To AddrBk

图 13.6 pine 程序的信件写作界面

如果想把信件发送给多个收件人,可以在不同的收件人姓名之间敲入逗号。如果想在 发送信件的时候附带上其他的文件,可以在 Attchmnt:数据域中输入这些文件的文件名或 者输入文件的完整的路径名。

当编写消息的时候,可以使用光标方向键在文本中移动。有许多编辑功能键可以用来 修改文字内容。也可以对文本行进行删除或者恢复删除、剪切或者粘贴等操作。这些编辑 功能键如表 13.2 所示。

表 13.2 p	oine	程序编写消息用到的编辑命令
----------	------	---------------

动作	组合键
光标向后退一个字符	Ctrl + B
删除光标前面的一个字符	Ctrl + H
取消消息	Ctrl + C
删除光标后面的一个字符	Ctrl + D
删除光标所在行	Ctrl + K
光标向下移动一行	Ctrl + N
光标向前移动一个字符	Ctrl + F
编辑帮助	Ctrl + H
在光标位置插入其他文件中的内容	Ctrl + R
段落对齐	Ctrl + J
光标移动到行首	Ctrl + A
光标移动到行尾	Ctrl + E
光标移动到下一个单词	Ctrl + @
向后翻页	Ctrl + V
向前翻页	Ctrl + Y
延迟发送(保存)消息	Ctrl + O
拼写检查	Ctrl + T
光标向上移动一行	Ctrl + P
搜索单词	Ctrl + W

当完成编写消息之后,可以使用 Ctrl+X 组合键立刻就把这个消息发送出去,或者使用 Ctrl+O 组合键延迟发送这个消息。

当检索邮件时, pine 程序会到子目录/var/spool/mail 中去查找, 然后从邮件文件中取出 消息,并存入到用户子目录中名为 mail 的文件夹中的缺省文件夹 INBOX 中。也可以建立 其他的文件夹,在不同的文件夹中保存或者删除消息,对邮件进行管理。

消息的文件夹索引显示在一个清单之中,可以使用光标方向键在其中上下查找选择消息。从消息的主清单中,可以删除、恢复删除、保存、阅读或者输出消息到子目录。如果 想要阅读某条消息,只需要敲一下回车键就行了。

13.5 习题

1. 阅读 sendmail 的帮助文件。

2. 设置 POP3 协议。

3. 使用 elm 或者 pine 编写邮件。如果条件允许,尝试发送。

第 14 章 配置 Samba 服务器

本章主要介绍 Samba 服务器的安装、配置和使用。内容包括 Samba 服务器简介、安装 Samba 服务器、启动 Samba 守护进程、配置 Samba 服务器、使用 Samba 服务器以及测试 Samba 服务器等。

14.1 Samba服务器简介

不管身在何处,只要身边有计算机,几乎都可以看到微软的 Windows 系列操作系统的 踪影,因为 Windows 操作系统的 GUI 界面漂亮、直观且简单易用,不能不让人想使用它。 但 Linux 的爱好者喜欢 Linux 系统的开放和每天有挖掘不完的新东西,而且可以把 Linux 设置的千变万化,完全适用于自己的计算机与本身的喜好。

如果两种操作系统的特性能互相配合将是一件非常好的事。并且,随着时间的推移和 各种或好或坏的基于 Windows 系统的应用程序的实现,Linux 系统也必须在 Windows 网络 中占有一席之地。

但是 Linux 与 Windows 系统之间的连线只能靠 TCP/IP 来实现,从 Windows 系统 telnet 到 Linux 系统或是用 ftp 做文件传输而己。Linux 虽然有 NFS 可以有效地和 Windows 相连, 然而 Windows 目前似乎没有一套免费、易用的 NFS 软件可以和 Linux 沟通。两种操作系统 之间似乎毫无关系。

Samba 的出现为 Linux 和 Windows 搭起了友谊的桥梁。

可以把 Samba 当成是一个局域网络上的 File/Printer Server,它可以提供文件系统、打印机或其他的信息并与 Samba Server 在同一个子网域的 Samba client(如 Windows98、WindowsNT等)共享。因为它的基本工作原理是让 NetBIOS(Windows 网络邻居的通信协议)与 SMB(Session Message Block)这两种协议运行在 TCP/IP 的通信协议上,且使用 NetBIOS name server 使 Linux 机器可以在网络邻居上被看到,所以 Linux 就可以和 Windows 在网络邻居上沟通,互相浏览分享文件了。

网络环境 NetBIOS 是作为 Microsoft 和 IBM 的结合项目而提出来的。它最初使用为小型计算机网络设计的无路由协议 NetBEUI 实现,接着又在 TCP/IP 之上重新加以实现。在 TCP/IP 之上实现的 NetBIOS 仍然有着其最初设计的局限性(如 WINS), NetBIOS 和 TCP/IP 的接合不能说十分协调。Microsoft 曾提出"通用 Internet 文件系统 (Common Internet File System)"的概念。可以这样认为,在网络规划中,NetBIOS 可以被完全忽略,但是事实并非如此。

Samba 是 UNIX 的 TCP/IP 之上的 NetBIOS 的自由实现。虽然在设计中存在一些问题,但 Samba 还是在企业中取得了较为广泛的应用,并经受住了很多挑战。Samba 的目标是在

UNIX 中重新生成 Windows 系统中的所有网络功能。

如果有许多人需要通过网络服务访问文件,并且还在同一时间对文件进行存取,请使 用 Samba 服务代替 NFS 服务。因为 NFS 服务不锁定文件,所以多个用户可以在同一时间 里访问同一个文件并试图对之进行修改,而最终的结果是只能够有一个用户的修改被保存 下来。另一方面,客户程序和服务器程序却能够跟踪对文件的存取访问,在某个文件被打 开的时候,对它进行"锁定"。这样在这个文件存盘并退出操作之前,其他人是不能再打开 它的。另外一个选择 Samba 代替 NFS 的理由是:采用微软公司的 Windows 作为操作系统 的计算机不支持 NFS,因此无法通过 NFS 服务跨 Linux/Windows 网络访问数据。

在 Red Hat Linux 7.0 中, Samba 是一个在系统安装时可选择安装的组件。若选择安装 了 Samba, 当系统初启时, Linux 系统是一个完全可操作的 NetBIOS 文件服务器。

14.2 安装Samba

Samba 服务器的安装可以有多种方式,本节介绍源文件形式和 RPM 形式的安装方法。

14.2.1 Samba的组成

整个 Samba 包括两个服务器守护进程、两个客户程序和一个管理工具程序。这些不同的组件在 Samba 套装软件中各有特定的作用。由于 Samba 里面包含了 Samba Server 和 Samba Client,所以安装 Samba 后,不仅可以提供资源给别人,同样也可以用 Samba Client 去分享 其他 Samba Server 或是 Windows 网络邻居上的机器共享的资源。

1. 硬盘驱动器和打印机共享

服务器组件 smbd 负责硬盘驱动器和打印机的共享。对 smbd 服务器程序进行访问的客 户端组件是 smbclient 程序。

如果选择安装的是 Red Hat Linux 操作系统,那么 smbd 服务器程序就已经安装并运行 在系统中了。而且,在系统的开机引导文件中,它也已经被配置为直接运行了。

2. 域名解析和浏览

服务器组件 nmbd 将激活 NetBIOS 信息的管理和传递。它还可以跨整个网络提供一系 列文件和打印服务。这些服务能够被网络中的其他机器浏览。用来访问 nmbd 服务器程序 的客户程序是 nmblookup。

当安装 Red Hat Linux 7.0 发行版本的时候,可能已经安装好了 Samba。如果没有把握, 请输入 rpm -q samba 命令检查这个软件包是否已经安装到硬盘上了。

14.2.2 安装Samba

在 Red Hat Linux 下安装 Samba 很简单,只要几个步骤就可以完成。目前 Red Hat Linux 7.0 的 Samba 版本为 2.0.7-21, 表示 2.0.7-21 版 patch 第 21 次。在 Red Hat Linux 发行版所 附光盘中捆绑有 Samba-2.0.7-21.i386.rpm 软件。可以直接从光盘上安装。当然也可以从 Red Hat 的站点上获得 Samba 软件,如果是 Samba 的源程序代码压缩文件,应该首先将它进行 解压缩,然后进行编译才可以使用。

如果使用 Red Hat Linux 7.0 光盘中的 Samba-2.0.7-21.i386.rpm 软件,因为它是 RPM 类型文件,无须编译,可直接阅读下面的"2"中的内容。

1. 使用源文件安装

下面以 Samba-xxxx.tar.gz 文件为例 (xxxx 代表下载的 Samba 源文件的版本号),介绍 编译 Samba 的基本步骤:

(1) 解压缩文件。

取得 samba-xxxx.tar.gz 文件,选定一个目录将它解压缩,包文件解压缩后,直接会产 生目录 samba-xxxx,命令如下:

#tar zxvf samba-xxxx.tar.gz

(2) 编辑 Makefile 文件。

先看一下 README、INSTALL 这样的文件,对于安装软件会有很大的帮助。因为 Samba 可适用于多种 UNIX 操作系统,所以 Makefile 里面定义了非常多的 UNIX 系统,只要找出 定义 Linux 的部份,然后将#删除就可以了。建议用含有 shadow passward 与 quota 的方式来 编译 Samba,如下所示:

Use this for Linux with shadow passwords

先从 Makefile 中找到上面这一段字。假如系统安装了 shadow password, 就将 Makefile 里的如下这段的#号删除, 然后储存 Makefile, 进行第(3)步骤。

FLAGSM = -DLINUX -DSHADOW_PWD -DFAST_SHARE_MODES # LIBSM = -lshadow

Use this for Linux without shadow passwords

假如系统没有安装 shadow password,就将 Makefile 里的如下这一段的#号删除,然后 储存 Makefile,进行第(3)步骤。

FLAGSM = -DLINUX -DAXPROC -DFAST_SHARE_MODES # FLAGSM = -DLINUX -DFAST_SHARE_MODES # LIBSM =

Use this for Linux with shadow passwords and quota

假如系统安装了 shadow password 和 quota,就将 Makefile 里的如下这一段#号删除,将 Makefile 储存起来,并直接在 source/目录执行如下命令:

#FLAGSM = -O3 -m486 -DLINUX-DSHADOW_PWD -DQUOTAS -DFAST_ SHARE_MODES # LIBSM = -lshadow

ln -s /usr/include/linux/quota.h /usr/include/sys/quota.h

多做这个步骤的原因是因为要包含 quota,所以编译时需要用到 quota 的 source。
Use this for Linux with quota and glibc2

glibc2 是 GNU 新开发出的 C Library, 所以若有使用 glibc2 与 quota,请将下面选项的# 号删除:

#FLAGSM=-DLINUX -DQUOTAS -DNO_ASMSIGNALH -DGLIBC2 -DFAST_SHARE_MODES # LIBSM = -L/lib/libc5-compat-lcrypt Use this for Linux with glibc2(libc6)

请只使用 glibc2 的用户将下面选项的"#"号删除:

#FLAGSM=-DLINUX -DNETGROUP -DSHADOW_PWD -DALLOW_CHANGE_PASSWORD
-DFAST_SHARE_MODES = 1 -DNO_ASMSIGNALH_DGLIBC2
LIBSM = -lnsl -lcrypt

(3) 使用 make 构造 Samba。

make 成功后会得到好几个可执行文件,但执行 SambaServer 或 client 最需要的是 smbd 和 nmbd 这两个 Samba 的守护进程和 smbclient 这个 Samba 的客户进程的执行文件并生成它。

接着执行 makeinstall 会将 Samba 的全部安装到/usr/local/samba/目录,就完成了安装 samba。若以前安装过 Samba,可以执行 makerevert 将旧的版本删除并安装新版本的 Samba。

#make

#make install

做完这3个步骤就完成了编译 Samba。

2. 使用 RPM 安装包安装

在 Red Hat Linux 7.0 中,可以直接使用光盘中捆绑的 Samba-2.0.7-21.i386.rpm 软件,因为它是 RPM 类型文件,无须编译即可直接安装。操作步骤如下:

(1) 首先将光盘放入 CDROM 中, 然后执行如下命令:

mount

cd /mnt/cdrom/

(2) 确认 CD-ROM 已经 mount 上,这时可以从光盘的/RedhatLinux/RPMS/目录下取 得 Samba-2.0.7-21.i386.rpm 这个 rpm 软件。

然后用 RPM 将它解压缩进行安装。RPM 这个安装软件可以将新软件的 source code 重新整理包装成另外一种 source 或 binary 的文件形式(.rpm),这样可以免去安装软件时编译 发生错误所产生的不良后果。命令如下:

rpm -ivh Samba-2.0.7-21.i386.rpm

可以看到 Samba 的软件名称以及它的安装过程,等到#字符跑完,就表示软件安装完成了。

安装完成后会自动在/etc/目录下放置产生 Samba 的 samba.conf 配置文件,在 /var/log/samba/下面放置 log.nmb 文件。

14.3 启动Samba守护进程

启动 Samba 守护进程有多种方式,本节将分别进行介绍。

14.3.1 从inetd中运行

请按照下面的步骤配置守护进程 smbd 和 nmbd 从 inetd 中运行:

- (1) 以根用户身份登录进入系统。
- (2)制作一份 xinetd.conf 文件的备份拷贝。
- (3) 编辑 xinetd.conf 文件。
- (4) 在文件中把光标移动到想输入语句代码的地方,文件末尾比较好。
- (5) 写出 smbd 语句。在 xinetd.conf 文件中,语句代码的顺序是:

[Service] [Socket] [Protocol] [Flags] [User] [Server Path] [Arguments]

按照上面的顺序, smbd 的相关数据依次是:

- · [Service] smbd 提供的服务是 netbios-ssn。
- · [Socket] smbd 使用的嵌套字是 stream。
- · [Protocol] smbd 使用的协议是 tcp 协议。
- · [Flags] 使用这个服务需要的标志是 nowait。
- · [User] 运行 smbd 的用户是根用户。
- · [Server Path] 这个守护进程的存放位置是/usr/sbin/smbd 目录。
- · [Arguments] 运行这个程序的参数是 smbd。

写出的 Smbd 语句是:

netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd

(6) 写出 nmbd 语句。经过与 smbd 语句相同的处理过程, 该语句是:

netbios-ns dgram udp wait root /usr/sbin/nmbd nmbd

(7)保存文件并退出。

(8) 输入 kill -HUP 1 命令重新启动 inetd 运行。

14.3.2 从系统的开机引导脚本程序中运行

如果通过一个 RPM 包安装了 Samba,那么全部必要的链接都将会自动地设置好。但如果是从源代码安装的,请按照下面的步骤进行:

(1) 以根用户身份登录进入系统。

(2)检查是否已经存在必须用到的命令脚本程序 smb。方法是:把路径切换到目录 /etc/rc.d/init.d/中,检查其中有没有 smb 文件。

(3)检查是否需要设置符号链接。方法是:先输入 ls -la smb 命令。

(4)在列出的文件清单中查看存取权限后面紧跟的数字。这个数字代表了到该文件的 链接个数。如果这个数字是1,就需要建立3个链接。

(5) 输入 ls -l /etc/rc.d/rc1.d *samba 命令。如果列出了一个 Samba 文件项就前进到下 一步。如果没列出,输入 ln -s /etc/rc.d/init.d/smb/etc/rc.d/rc1.d/K35smb 命令建立第1个链接。

这个链接设置的效果是:当进入第1运行层次,即单用户模式时,杀掉 Samba 进程。 (6)输入 ls -1 /etc/rc.d/rc3.d *samba 命令。如果列出了一个 Samba 文件项就前进到下 一步。如果没列出,输入 ln -s /etc/rc.d/init.d/smb/etc/rc.d/rc3.d/S91smb 命令建立第2个链接。

这个链接设置的效果是:当进入第3运行层次,即命令行模式时,启动 Samba 进程。

(7) 输入 ls -l /etc/rc.d/rc6.d *samba 命令。如果列出了一个 Samba 文件项就前进到下 一步。如果没列出,输入 ln -s /etc/rc.d/init.d/smb/etc/rc.d/rc6.d/K35smb 命令建立第3个链接。

这个链接设置的效果是:当进入第6运行层次,即重启动模式时,杀掉Samba进程。

如果不需要建立任何链接,说明系统已经配置好了从系统的开机引导脚本程序中运行 这些守护进程。

14.3.3 不从系统的开机引导脚本程序中运行

如果打算让 smbd 和 nmbd 从 inetd 中开始运行,但是系统当前的缺省设置是从系统开 机引导脚本程序中运行,就必须首先设置这些守护进程,使它们不再从系统的开机引导脚 本程序中开始运行。请按照下面的步骤进行操作:

(1) 以根用户身份登录进入系统。

(2) 输入 ls -l /etc/rc.d/rc1.d *samba 命令。如果列出了一个链接,就需要输入 rm /etc/rc.d/rc1.d filename 命令删除它。

(3) 输入 ls -l /etc/rc.d/rc3.d *samba 命令。如果列出了一个链接,就需要输入 rm /etc/rc.d/rc3.d filename 命令删除它。

(4) 输入 ls -l /etc/rc.d/rc6.d *samba 命令。如果列出了一个链接,就需要输入 rm /etc/rc.d/rc6.d filename 命令删除它。

14.3.4 手动启动Samba守护进程运行

如果想手动启动 smbd 和 nmbd 守护进程运行,请以根用户身份登录进入系统,然后输入/etc/rc.d/init.d/smb start 命令。

14.4 配置Samba

Samba 有很多选项,但对于大多数任务,了解很少几个选项就足够了。下面介绍这些 选项和约定。注意许多选项后面有(G)或(S),表明它们所对应的是[global]段或者共享 段。

许多以用户作为值的选项也可以使用组作为值。在这种情况下组名的前面有一个@符 号。 例如,组 acct 可以表示为@acct。

在 smb.conf 文件中可以使用几个替换字符, 它们在 smb.conf 手册页中都有介绍。其中的两个%u 和%H 特别有用。%u 用用户名替换, 而%H 用用户的主目录替换。

read Only

read Only=与 writeable =、 writable =和 write ok =的功能相反,后面的3个是等效的, 彼此可以任意替换。所以 read Only = yes 可替换 writeable =no, read Only = no 可替换 writeable = yes。默认为 read Only = yes。

如果该选项在[global]段中指定,则它作为所有共享的默认值。

这些选项可以被 write list = 选项覆盖。

• valid users = (S)

任何共享中不选择 valid users = (S)选项或者等号右边为空,表示共享可以被任何人访问。

该选项允许 myuid、tackett 以及组 acct 访问。该选项可被 invalid users =选项覆盖。

• invalid users = (S)

不能访问该共享的用户列表。该列表覆盖任何在 vaild users = 选项中指定的用户。该 smb.conf 文件片段只允许 art 访问[ateam]。

• read list = (S)

该选项的值是一个列表,列出允许只读访问的用户。该选项覆盖任何 read Only=等选项,限制列出的用户只读访问。如果某个在 read list=列表中的用户也在 write list=选项中,则覆盖 read list=,该用户可以在该目录中写入。

• write list = (S)

任何共享都有一个可以写入该共享的用户的列表,而不管 writeable =或者 read list =选 项的内容。

• path = (S)

这是通过共享访问的目录。在打印共享的情况中,表示假脱机打印目录。如果[global] 段包含 root =、root dir =、或者 root directory = 选项,则 path =是相对于 root 指定的目录。

• create mask = π create mode = (S)

这两个选项等效,它们指定最近创建的文件的最大权限。DOS 权限将进一步限制该文件,它默认为 744,即用户拥有完全控制权限而组和其他人只可读。如果拥有者以后在 DOS 环境中将该文件设置为只读,则在 Linux 系统中该文件的权限改变为 544,以反映丢失的写权限。

• browseable = (S)

browseable=选项指示 SMB 客户是否在 SMB 客户的浏览器列出该共享。对于不在 valid users =列表中的用户不授予访问权限。browseable =no 不拒绝在 valid users=列表中的用户的访问。

如果设置为 yes,则该共享可以被没有权限的用户看到。如果设置为 no,则即使在 validusers = 列表中的用户也看不到。但是在客户端允许访问没有列出的共享(例如 smbclient), browseable = no 并不阻止合法用户访问共享,只要输入正确的命令和共享名。

• printable = (S)

允许从共享打印。因此必需在任何打印机共享中使用,而不能用于其他共享。在[printers] 段, printable=的默认值为 yes,在其他任何地方的默认值为 no。

• hosts allow =, hosts deny =, allow hosts = π deny hosts = (S)

hosts allow = 控制可以访问共享的主机和子网。如果该选项在[global] 段中使用,则成 为所有共享的默认值。如果使用该选项,则拒绝所有没有被指定的主机或子网入口。

hosts allow =覆盖任何 hosts deny =选项,后者简单拒绝对主机或子网的访问。allow hosts=和 hosts allow =等效; deny hosts =和 hosts deny =等效。

• public = π guest ok = (S)

这两个选项等效。该选项的目的是允许在服务器上没有登录帐号的用户可以访问共享。 这是一种安全的折衷方法,对于打印机有意义。必需小心使用该选项,避免允许破坏性地 使用资源的可能。基于这个原因,默认值为 no。

• comment = (S) π server string = (G)

这两个选项的相关之处在于在 SMB 客户的用户界面上都提供了标识 Samba 资源的易 读的字符串。comment=描述共享,而 server string=在[global]段中出现,它描述整个 Samba 服务器。

• domain logons = (G)

默认值为 no。如果设置为 yes, 允许 Samba 服务器作为 Windows98 工作组的域服务器 使用。

• encrypt passwords = π smb passwd file = (G)

该选项在为 Windows 客户服务时十分重要。默认值为 encrypt passwords = no 和 smb passwdfile = /etc/smbpasswd。

• hosts equiv = (G)

这是一个危险的选项,它指向一个包含不需要口令就可以登录进入的主机和用户名的 文件。这是安全隐患,默认值为 none。最好的策略是使该选项从 smb.conf 文件中消失。

• interfaces = (G)

当服务器服务于多个子网时需要这个选项。

• load printers = (G)

默认值为 yes。yes 值装载 printcap 中的所有打印机用于 Samba 浏览。

• null passwords = (G)

该选项默认值为 no,表明在服务器上没有口令长度为 0 的用户可以登录到 Samba。如 果该选项设置为 yes,则安全显然受到影响。

• password level 和 username level (G)

这些选项决定大小写无关的用户名和口令比较的级。默认值为 0,表明客户提供的口 令或用户名首先与服务器上的记录进行大小写比较。如果失败,则客户端的用户名或者口 令转换为小写,然后和服务器上的记录进行比较。

在解决 Windows 客户端的 Samba 连接问题时,将该值设置得高一些(如 24)通常比较好,然后查看是否解决了问题。尽管这意味着较小的安全问题和缓慢的初始连接速度, 通常却能够解决问题。一旦问题解决,则重新注释这两个选项,保证安全。

Windows 客户端的连接问题经常还可以通过 encrypt passwords = 和 smb passwd file =

选项解决。

• security = (G)

默认值是 security = user ,强制通过用户和口令获得安全。通常这是最好的选择,这样 具有极好的安全性和可预测性。

security = server 和 security = domain 主要使用在口令验证自动由另一台计算机完成的 情形。

security = domain 用于将 Samba 连结到 WindowsNT 域。security=share 提供较小的安 全性和较小的可预测性操作,但是有时候,当大多数客户的用户名不在服务器上或者如果 大多数使用的是不需要口令的打印机时,在这种安全性关系不大的情形下,这也是一种合 理的选择。

• workgroup=(G)

这是服务器出现的工作组,它也控制使用 security=domain 设置的域名。

• config file=(G)

这是一种指定另一个 Samba 配置文件的方法。当 Samba 遇到该选项时,从指定的文件中重新装入所有的参数。

14.5 使用Samba

Samba 服务器的使用比较复杂,本节将从各个方面介绍 Samba 服务器的使用。

14.5.1 配置Linux到Windows共享

请按照下面的步骤把 Linux 操作系统中的某个驱动器配置为允许 Windows 共享:

(1) 在作为 Linux 方面 Samba 服务器的 Linux 计算机上以根用户身份登录进入系统。

- (2)编辑/etc/samba/smb.conf文件。
- (3) 在开头部分加上[global]字样。
- (4)从[global]语句下面一行开始设置全局性定义参数。

(5)在各个共享部分自己的定义段中建立各自的共享设置。为每一个希望共享的驱动 器建立一条相应的定义语句。下面给出一个例子:向普通用户提供一个名为 public 的共享 硬盘空间。需要在自己的文件系统中为它专门分配一个硬盘空间。推荐给公共空间分配独 立的硬盘分区,这样可以加强一些安全性,因此需要在硬盘驱动器上添加一个名为/public 的分区,如下所示。普通用户不能拥有写操作权限,但能够读其中的文件。

[public]

comment = Items available to the general public .

path = /public

public = yes

readable = yes

- writeable = no
 - (6)保存文件并退出。

(7) 重新启动 samba 守护进程运行。在 Red Hat 发行版本中,先输入/etc/rc.d/init.d/smb stop 命令,然后输入/etc/rc.d/init.d/smb start 命令。

在 Samba 套装软件中有一个工具程序专门用来测试 smb.conf 文件是否正确地设置好 了。请输入 testparm 命令运行这个工具程序。

14.5.2 配置Windows到Linux共享

请按照下面的步骤确保 Linux 计算机中的 Samba 可以看到对方 Windows 机器上的硬盘 驱动器:

(1) 打开 Windows 的 Control Panel。

(2) 双击 Network 控制面板。

(3) 单击 Configuration 标签。

(4) 单击 File and Print Sharing 按钮, 打开 File and Print Sharing 对话框。

(5) 确认 I want to be able to give others access to my files .选项已经被选中。这个选项 通常是缺省选择的。

(6) 单击 OK 关闭 File and Print Sharing 对话框。

(7)在 Network Components 组合列表框中单击选中 TCP/IP。

- (8) 单击 Properties 按钮打开 TCP/IP Properties 对话框。
- (9) 单击 WINS Configuration 标签。
- (10) 选中 Enable WINS Resolution 单选按钮。
- (11) 输入 Linux 方面 Samba 服务器运行在其上的计算机的 IP 地址。
- (12)单击 Add 按钮把这个 IP 地址添加到 WINS 清单中。

(13) 单击 OK 按钮关闭 TCP/IP Properties 对话框。

(14) 在 Network Components 组合列表框中单击选中 File and printer sharing for Microsoft networks.。

(15) 单击 Properties 按钮打开 File and printer sharing for Microsoft Networks Properties 对话框。

(16)在 Properties 组合列表框中单击选择 Browse Master。

(17) 在 Value 下拉列表中选中 Disable。

(18) 单击 Access Control 标签。

(19) 确认 Shared-level access control 单选按钮被选中。这个选项通常是缺省选择的。

(20) 单击 OK 按钮关闭 Network 控制面板。等待 Windows 生成新的驱动程序。

(21)使用 Windows 中的"资源管理器"或者其他喜欢的 Windows 文件管理工具继续 操作,对从 Linux 方面的 Samba 客户程序中允许进行共享访问的任何硬盘驱动器或者文件 夹进行配置。

14.5.3 测试Linux共享设置

请按照下面的步骤测试一台计算机上的 Windows 能否正确访问共享的 Linux 硬盘驱动器:

(1) 登录进入希望对之进行测试的 Windows。

(2) 双击桌面上的 Network Neighborhood 图标。在 Samba 服务器上配置为允许访问的 Linux 硬盘和目录都应该显示为可用状态。

14.5.4 测试Windows共享设置

请按照下面的步骤测试 Linux 能否正确访问共享的 Windows 硬盘驱动器:

(1) 登录进入要测试的 Linux。

(2) 输入 rpm -q samba 命令检查客户程序是否已经安装好了。如果还没有安装,挂 装发行版本的 CD-ROM 光盘,然后从 RPMS 目录中安装它。

(3) 输入 smbclient -L 命令查看登录的这个帐户都允许访问哪些共享资源。把看到的 清单与在网络的 Windows 方面已经配置好的情况加以比较。

(4) 使用下面的命令把 Windows 硬盘挂装到 Linux 文件系统上:

#/usr/sbin/smbmount //machinename/path mount_point

(5)访问 Windows 硬盘上的文件。

(6) 当操作完成的时候,使用下面的命令卸载刚才挂装的 Windows 硬盘:

umount mount_point

14.5.5 配置Linux打印机共享

请按照下面的步骤配置 Windows 机器通过 Samba 服务使用一台 Linux 打印机:

(1) 确认这台打印机在 Linux 操作系统中工作正常。

(2) 以根用户身份登录到连接着打印机的计算机中。

- (3)编辑/etc/smb.conf文件。
- (4)前进到文件的末尾。

(5)如果希望用户能够被允许连接到已经在 Linux 网络中配置好的全部打印机上,请 建立一个叫做[printers]的定义段,并填好其中的定义值。这个定义段一般应该是如下所示:

```
[printers]
path = /usr/spool/public
writeable = no.
guest ok = yes
printable = yes
```

(6)如果希望手动操作(这样不至于包括全部打印机),就需要为每一台想让 Windows 机器使用的 Linux 打印机建立一个定义语句。每个语句的开头需要指明这个语句定义的打 印机的名字。比如[laser1]。

(7)把想要设置给打印机的各个定义选项填写好。

(8)保存/etc/smb.conf文件并退出。

(9)重新启动 samba 守护进程运行。在 Red Hat 发行版本中,先输入/etc/rc.d/init.d/smb

stop 命令, 然后输入/etc/rc.d/init.d/smb start 命令。

14.5.6 配置Windows打印机共享

按照下面的步骤把一台连接到 Windows 计算机的打印机配置为可以从 Linux 机器上通 过 Samba 服务访问:

(1) 以根用户身份登录进入到每一台 Linux 机器上。

(2)为每一台 Windows 打印机在所有 Linux 机器中的/etc/printcap 文件添加一个定义数据项。可以使用 Red Hat 发行版本中的 GUI 打印工具程序生成这些定义数据项,也可以对它们分别进行手动设置。

(3)确定是否需要把 ASCII 文件转换为 Postscript 格式。如果需要,请以根用户身份 登录进入系统,并输入 nenscript 命令查看 Postscript 转换程序是否已经安装在 Linux 系统中 的某个位置。

如果找到了这个命令脚本程序,请记下它的存放位置。如果还没有安装,就需要安装 一份这个命令脚本程序的拷贝,它应该包括在发行版本的 CD-ROM 光盘中的 RPM 软件包。

(4)如果已经从 Red Hat 发行版本的 CD-ROM 光盘中安装好了 samba 软件包,那么 smbprint 命令脚本程序就已经在硬盘上的/usr/bin 目录中了。如果想处理来自 Windows 机器 的打印作业,就必须用到这个命令脚本程序。它的使用方法是:在/etc/printcap 文件中加上 一条语句,为准备用来提供共享打印服务的打印机加上一个输入过滤器,如下所示:

: if = /usr/bin/smbprint : $\$

14.6 测试Samba

如果 SambaServer 不能正常工作,可以依照下列步骤完成测试工作。至少要有两台计算机可以使用,一台是 Linux,一台的操作系统是 Windows98,这样才能进行测试工作。 另外可以将/usr/local/samba/bin/这个路径加入到.profile 或.cshrc 中(如果使用 bash 或 tcsh),以方便使用 Samba 的执行程序。

(1)使用 Samba 提供的测试程序 testparm,测试 smb.conf 的设置是否正确,命令如下:

#testparm

(2)使用 ping,分别从 Linux 和 Windows98 互相 ping 对方。如果双方都有反应,表示网络是顺畅的。

(3)在 Linux 上执行 smbclient -L hostname,测试能不能列出 Samba Server 共享的资源。smbclient 是 Samba 提供的 client 程序,使用它可以存取网络上任何一台 SamabServer 共享的文件。测试中加-L 就是列出 Samab Server 提供的资源清单。

(4) 在 Linux 上执行 nmblookup -B hostname_SAMB。应该会得到 Samba Server 回传的计算机的 IP 地址。例如:

#nmblookup -B david_SAMBA_

假如没有回传而且是用 inetd 启动 Samba,则请检查 xinetd.conf 中关于 nmbd 的设置 是否正确。如果使用 standalone 启动,可能是在 smbd 或 nmbd 的地方加错参数了。

(5)在 Linux 上执行 nmblookup -B IP_Address*', 其中的 IP_Address 是 Windows98 的网址, 假如使用的 Windows 98 的 IP 为 140.115.219.145, 则执行:

#nmblookup -B 140.115.219.145

Windows 98 应该回传它的 I P。如果没有,应该是 IP 地址不正确,或是 Windows 98 没 开机,不然就是 Windows 98 关于网络的设置没有安装好。

(6)在 Linux 上执行 nmblookup -d 2, 会回传和 Samba Server 在同一个子网的所有的 Windows 98 计算机网址。

(7)在 Linux 上执行 smbclient \\hostname\Sharename, 登录自己的 Samba Server 。 hostname 是计算机名称, Sharename 则依据前面 smbclient -L david 的清单可以登录暂存区 这个目录。

(8) 在 Linux 上执行如下命令:

smbclient ' \\hostname\sharename ' -I xxx.xxx.xxx

分享 Windows 98 网络邻居上共享的文件。但必须先知道 Windows 98 的 hostname 和它 共享的目录,以及它的 IP 地址,才能存取这台机器上的文件。

(9)在 Windows 98 上执行 net view\\samba_name,其中 samba_name 就是 Linux 的 hostname。如果能看到 david 这台机器,就表示 Samba 没问题,如果发生错误,如显示 network name not found,则必须查看 nmbd 有没有错误,或在 Windows 98 网络设置的 TCP/IP 协议中的 wins server 上加上计算机名称,或在 Windows 98 计算机的 C:/etc/Imhost 文件里面加上 Windows 98 的网址,格式如下:

#C:/etc/lmhost ip address hostname tester

(10) 查看能不能在 Windows 98 计算机的网络邻居上看到 Samber Server 是最直观的 方法。只要在网络邻居上可以看到 Server,并且可以进入共享的目录,就完全没有问题。

14.7 习 题

1. 如果系统未安装 Samba 服务器,尝试安装。

2. 配置 Samba 服务器,使用不同的启动方式。

3. 适当修改 Samba 配置,查看系统有何不同。

第15章 维护系统的网络安全

本章主要介绍 Linux 系统的网络安全维护。内容包括服务器安全保护简介、防火墙简 介、加强 Linux 的安全管理以及设置 Linux 防火墙等。

15.1 服务器安全保护简介

对于网站管理人员而言,日常的服务器安全保护主要包括以下4个方面内容:

1. 文件存取合法性

任何黑客的入侵行为的手段和目的都可以认为是非法存取文件,这些文件包括重要数 据信息和主页页面 HTML 文件等,这是计算机安全最重要的问题,一般地,未被授权使用 的用户进入系统,都是为了获取正当途径无法取得的资料或者进行破坏活动。

良好的口令管理、登录活动记录和报告以及用户和网络活动的周期检查都是防止未授 权存取的关键。

2. 用户密码和用户文件安全性

这也是计算机安全的一个重要问题。在具体操作上就是防止已授权或未授权的用户相 互存取重要信息。文件系统查帐、su 登录和报告、用户意识以及加密都是防止泄密的关键。

3. 防止用户拒绝系统的管理

这一方面的安全应由操作系统来完成。操作系统应该有能力应付任何试图或可能对它 产生破坏的用户操作。

4. 防止丢失系统的完整性

这一方面与一个好系统管理员的实际工作和保持一个可靠的操作系统有关。一个好系 统管理员的实际工作包括定期备份文件系统;系统崩溃后运行 fsck 检查;修复文件系统; 当有新用户时,检测该用户是否可能使系统崩溃的软件等工作。

15.1.1 Linux系统的文件安全

Linux 的文件系统是由文件和目录构成的树形结构,每个文件目录记录包括文件名、文件类型、文件大小、文件创建修改时间、文件所有者和所有组以及文件相关权限等内容。 任何一项内容遭受未授权的修改,文件安全性都遭到破坏。保护文件系统的安全性,应该 从以下几个方面入手。

1. 文件相关权限的设置

Linux 的文件权限决定了对该文件的操作能力和操作允许范围。

文件权限通过设置文件权限标志位实现。当一些关键的系统文件的属性被错误设置时, 就会导致不可挽回的破坏。设置文件属性一定要非常小心,否则可能导致致命的安全漏洞。

2.SUID 和 SGID 程序

与文件有关的还有两个附加权限位 SUID 和 SGID。SUID 是 SetUserID(设置用户标识)的缩写; SGID 是 SetGroupID(设置组标识)的缩写。带有这种权限的程序运行时就会带来很大的安全性漏洞。因为当运行一个 SUID 程序时,它的有效 UID 被设置为拥有该程序的用户 ID,而不管实际上是哪个用户在运行,SGID 与此类似。虽然 SUID 程序是必需的,但应该尽量减少使用机会,并且要尽最大努力保证此程序安全。作为管理员还应该经常使用 find 命令浏览自己的文件系统以检查新的 SUID 程序。

15.1.2 用户访问安全

1. 口令安全

每个 Linux 的用户都拥有一个帐号,通过登录到这个帐号才能有限制地使用系统。而 保护自身的文件安全的惟一屏障就是口令,一旦这道屏障被突破,整个系统的安全便无法 得到保证。

从安全角度看,口令最好是随机产生的,并且不断变换。但实际上任何一个用户都不 愿意花费时间去记忆刚刚更换过的口令。所以应该做到尽可能保持频繁的口令更换频率, 并且聪明地选择自己的口令保证其安全。切忌选用与自己有关的一些数字、名词、住址、 配偶名称、宠物名称和电话号码等,更不要选用字典中的词汇作为口令。因为当今使用的 很多破解口令程序都是通过一定的加密算法将字典中的词汇一个一个与口令作比较,以期 闯入系统。另外从概率角度讲,口令位数每增加一位,被破解的可能性就会相差很多数量 级,因此在系统允许的范围内尽可能长地设置自己的口令实为明智之举。

常用的选择口令的方法可以参考以下 3 种:

- · 选择一个短语,取出每个单词中第一个字母组合起来形成口令,这样形成的口令 接近于随机字符序列,但同时也容易记忆。如从 Do you still love me today 得到的 口令就是 dyslmt 。
- · 选择几个较短的单词,其间用符号或者某特定的字母加以连接。
- · 一个变形的单词,或者有错误的短语。

2. 登录安全

在系统中查找可能有安全问题的帐号并及时处理是很重要的。需要经常注意的问题有 如下几个:

(1)管理员应该经常检查口令文件,如果发现没有口令的帐号,应该立刻通知用户或 者禁止其使用。因为很多黑客是使用没有口令的帐号进入计算机系统的。

(2)对于不使用的帐号应该及时删除,即使不删除,至少也应该在口令字段写入符号, 暂时停止此帐号的使用,使之不至于成为黑客进入的通道。

(3)对于几个标准系统帐号,一般情况下应该禁止这些帐号的使用。因为这些帐号几 乎是每个非法闯入者的目标,所以最安全的办法就是禁止其使用。还有一些软件在安装过 程中会自动在系统中创建帐号,所以一定要注意禁止这些帐号的使用。

(4) 对于 guest 帐号,一般情况下是不建议使用的。由于 guest 帐号一般不设密码, 所以是进入系统的捷径。黑客进入系统以后可以进一步获得更高的权限,将导致安全性灾 难。

(5)系统中还有几个命令帐号,也就是运行给定命令然后退出。这些帐号没有口令, 虽然它们并不运行 shell,但是从安全方面来讲也是极其危险的。这种帐号可能会泄露系统 的有关信息。

(6)组帐号是供多人使用的同一个帐号。这对系统安全是极为不利的。建议创建帐号 时遵循"一个帐号一个用户"的原则。

15.2 防火墙简介

防火墙这个概念的由来有两种说法,一种说法是防火墙是汽车中一个部件的名称。在 汽车中,利用防火墙把乘客和引擎隔开,一旦汽车引擎着火,防火墙不但能保护乘客安全, 同时还能让司机继续控制引擎。另一种说法是防火墙术语来自于建筑业。防火墙通常用防 火材料建成,位于相连接的公寓或建筑之间。有了防火墙,在墙这边发生的火灾不会蔓延 到墙那边的建筑。

在计算机网络中,防火墙是在被保护的网络与 Internet 之间限制访问的部件。它可以以路由器或计算机等形式作为硬件来实现,也可以是在一个网关系统上运行的软件,或是它们的组合。Steve Bellovin 在 1987 年在给同事的一封电子邮件中第一次把该术语用于计算机安全。

15.2.1 防火墙的作用

Firewall FA Q 的作者 Marcus J.Ranum 和 Matt Curtin 很好地总结了使用防火墙的理由: "Internet 像任何其他社会一样被各种各样的人所困扰,他们喜欢用画笔在别人的墙上 写字、撕破别人的邮箱或只是坐在街上按喇叭来享受电子平等。有些人试图通过网络做一 些真正的工作,其他人有一些保密或私人数据需要保护。一个防火墙的目标就是把那些无 事生非的人挡在网络外,而你仍能完成自己的工作"。

防火墙就像交通警察一样做两件事之一:让交通工具通过,或把它们挡住。防火墙可 以允许或拒绝数据通过。没有防火墙,可能是因为 NIS 的 NFS 这样的服务、薄弱的口令或 系统的错误配置,一个 LAN 或子网对于外部攻击表现得很脆弱。整个网络的安全完全依赖 每台主机上维护一致的、高水平的安全措施,但是在复杂的网络上很难维持这种状态。

防火墙能够完成的工作包括:

- 1. 控制到内部系统的访问,只有邮件和公众信息服务器可以从外部到达。
- 2. 针对每台机器或每个用户来阻止对特定 Internet 站点的访问。
- 3. 通过过滤 NFS 这样不安全的服务来提高网络安全性。
- 4. 让系统管理员致力于单独一个系统的安全而不是 LAN 中每台主机。
- 5. 通过阻止到 finger 和 DNS 这类服务的访问来加强保密。

7. 在两个远程位置之间建立一个加密连接。

15.2.2 防火墙的分类

防火墙大致可分为网络级防火墙(包过滤防火墙)和应用程序级防火墙(代理服务器)。 网络级防火墙只是根据数据包的内容来判断是否允许数据包的传输。大多数的包过滤 防火墙是根据源地址或目标地址、源端口或目标端口和数据包是否为正在进行的会话的一 部分来进行判断。

应用程序级防火墙充当客户和服务器程序之间的中介。客户程序不直接连接到服务器 中,而是连接到代理服务器上,并到那里询问信息。代理服务器打开一个到服务器的连接, 发出请求,并继续在服务器和客户之间传输数据。

随着防火墙技术的混合和搭配,两者之间的区别越来越少。

1. 网络级防火墙

网络级防火墙在 IP 层工作。它依据起点、终点、串口号和每一个数据包中所含的数据 包种类信息控制数据包的流动。这种防火墙非常安全,但是缺少有用的登录记录。它阻挡 别人进入个别网络,但不能记录何人进入公共系统,或何人从内部进入网际网。网络级防 火墙是绝对的过滤系统。即使要让外界的一些人进入私有服务器,用户也无法让每一个人 进入服务器。

网络级防火墙的一个基本例子就是位于 Internet 和内部网络之间的路由器,它根据数据 包的来源、目的地址和端口来过滤。一个包过滤路由器能以不同的方式阻止来自指定主机 或网络的连接,或者阻止到指定端口的连接。一个站点也许要阻止从一个敌对或不可信地 址到来的连接,或者阻止所有来自外部地址的连接,除了与外部交换邮件的 SMTP。

网络级防火墙速度快且对用户透明。它们相对比较容易建立而且便宜,因为许多机构 已经拥有了与 Internet 相连的路由器。它们可以用直观的规则来编程,如"阻止未启用服务 的包"或"允许到信息服务器但阻止到其他主机的 TCP 连接"。它们还很灵活,如某一子 网的某人攻击用户的一个主机,用户可以配置一个网络级的防火墙来阻止来自该子网的所 有访问。

包过滤路由器有一些潜在的弱点:包过滤规则叙述起来很复杂,过滤规则集可能太绕 弯,使漏洞很难检测到。

2. 应用程序防火墙

克服包过滤弱点的一个通用方法是使用一个应用程序级网关。应用程序级网关是一个运行 proxy 服务的主机,用于过滤 telnet 和 FTP 这类服务连接。它可以和一个包过滤路由器组合起来提供更好的安全性和灵活性。

应用程序防火墙允许通过防火墙间接进入网际网。最好的例子是 telnet 到系统,然后 从该处再 telnet 到另一个系统。在有应用程序防火墙的系统中,这项工作就完全自动完成。 利用客户端软件连接应用程序防火墙后,应用程序防火墙启动它的客户端软件(代理),然 后传回数据。由于应用程序防火墙重复所有通信,因此能够记录所有进行的工作。只要配 置正确,应用程序防火墙就是安全的,这是它最可取之处。由于没有直接的 IP 通路,它阻 挡任何人进入。

这两类防火墙都各有优缺点,很多的站点两个都使用。网络级防火墙被认为安全方面 不如应用程序防火墙,因为它只利用了应用程序所使用的协议。当然,除了 IP 地址之外它 还可以使用附加的验证方法。

使用应用程序防火墙的另一个好处是它可以减少带宽的使用。如一个机构可以将代理 服务器加入到防火墙中并存储用户请求的页面。如果另一个用户请求了同样的页面,代理 服务器将把页面从自己本地存储器中取出并发送给用户,而不需要再次连接到原始服务器。 这种用法作用于上百个用户,就可以很好地节省带宽。

另一方面,应用程序级别防火墙必须为每个协议创建代理。这样,就有了 HTTP 代理、 FTP 代理、telnet 代理和 SMTP 代理等等。这又产生了不被支持的应用程序问题,即这个应 用程序所使用的协议还没有代理。虽然有一般性的代理,但是应用程序必须在更改并重新 编译它之后才可以使用。而数据包过滤器工作在单数据包上,因此对应用程序是透明的。

网络级防火墙可以做像网络地址转换或伪装这样的事情。也就是说防火墙将源地址转 换到输出数据包中,这使得其他的主机认为是自己连接到防火墙,而将目标地址转换到输 入数据包中,这样它们会到达最开始申请连接的主机。如整个网络可以使用一个 IP 地址连 接到 Internet。因为外部无法看到内部的 IP 地址,所以内部的主机对外部的攻击更有抵御 能力。缺点是为外部提供服务的主机必须通过防火墙提供服务。但是,防火墙需要特殊的 配置来允许一些数据包直接发送到特定的主机。

15.3 Linux的安全管理

有一些简单的方法可以加强 Linux 的安全管理,本节介绍一些使用小技巧。

15.3.1 关闭简单的薄弱环节

不论什么版本的 Linux 操作系统,其中都有一些能够禁用的薄弱环节,下面的做法能 够提高系统的安全性:

1. 在/etc/xinetd.conf 文件中禁用任何不打算使用的服务功能。方法是:把它们改为注 释语句(在那些语句开头加上一个#符号)。任何一种服务都会为试图进入系统的人多打开 一条通路,因此应该只使用确实需要的服务。

2.编辑/etc/issue.net 文件,删除其中关于机器上运行的特定硬件和 Linux 发行版本的 介绍信息。这些信息会在诸如 telnet 之类的远程登录任务操作过程中显示在登录端的屏幕 上。

3.把某些特殊的用户们分配到 console 用户组中。这样当这些人实际坐在服务器计算 机前面时可以执行命令,但同时要禁止任何其他人调用这些命令。例如,如果需要强调安 全性,那就应该只允许 console 用户组的成员挂装磁盘。

4. 查看/etc/securetty 文件,确定其中列出的设备都是真实存在的物理 ttys (如 tty1 到 tty8)端口。这个文件限制了人们能够以根用户身份登录进入系统的位置。允许任何远端用 户以根用户身份登录进入系统是极其危险的,因为这样做就为潜在的侵入者缩短了侵入超 级用户帐户的过程。

5. 一定要保证在 FTP 服务器上唯一允许匿名用户进行写入操作的部分是/incoming 目录。

15.3.2 列出成功登录的记录清单

如果想查看都有哪些人最近成功地进入了系统,可以使用 last 命令。如果想列出比缺 省数目更多的记录,可以使用格式 last -n number 告诉 last 命令需要显示多少登录记录。

如果这个命令执行失败,说明登录操作还没有被记录下来。为了确保它们能够被记录 下来,请以根用户身份执行 touch /var/log/wtmp 命令来建立日志记录文件。

15.3.3 列出不成功登录的记录清单

如果想查看都有哪些人最近没有成功地登录进入系统,可以使用 lastb 命令。和 last 命 令相类似,如果想列出比缺省数目更多的不成功尝试记录,可以使用 lastb -n number 格式。

如果这个命令执行失败,说明不成功登录操作还没有被记录下来。为了确保它们能够 被记录下来,请以根用户身份执行 touch /var/log/btmp 命令来建立日志记录文件。

15.3.4 查找安全漏洞

如果安全出现了漏洞,或者认为已经出现了漏洞,就可以抓住这个机会从系统登录记 录或者在其他地方查找蛛丝马迹。比较好的办法如下:

1.系统会记录下一次成功的登录操作和不成功的登录尝试操作。特别是"列出不成功 的尝试登录记录清单",它能够很快地查找出是否有人在千方百计地获取根用户权限或者尝 试某个用户的口令字。

2.如果使用了 shadow 隐藏口令字软件,请检查/etc/passwd 文件,看看其中是否禁用 了 shadow 隐藏口令字功能。如果是这样,口令字将会被保存在/etc/passwd 文件里。修改这 个设置必须拥有根用户的权限,所以如果情况真的如上面所说的这样,就应该知道在这台 机器上黑客已经获得了根用户的操作权限。

3.检查是否有不熟悉的用户帐户拥有特殊的优先权。

4. 一个拥有高级工具和知识的黑客能够替换某些重要的系统功能。用一台独立的计算 机或者一台没有遭到攻击的机器把诸如 ls 之类的命令备份到一张软盘上,然后检查程序 ls、 find、ps 和所有网络守护进程的系统版本中的日期与它们生成时的日期数据是否一致。

5.如果网络服务日志或者任何系统日志中有时间缺口,而并没有在时间缺口期间关闭 过系统日志记录功能,就应该怀疑安全受到破坏。

15.4 设置 Linux 防火墙

Linux 内核中包含了对数据包过滤和伪装的支持, Linux 也提供了不少的基于代理的防火墙软件包,如 Squid HTTP/FTP 代理缓存和 SOCKS 代理。请查找文件/proc/net/ip_fwchains 以证实内核是否支持 IP 链。如果这个文件存在, 说明内核支持 IP 链。

如果这个文件不存在,需要启用内核防火墙。为启用它,必须配置内核并启用 Network

Firewalls 和 IP:Firewalling。如果要进行伪装,还要启用 IP:AlwaysDefragment、IP:Masquerading和 ICMP:Masquerading。在这之后,重新编译安装内核,并重新启动。

从 2.2 版的 Linux 内核开始,包过滤是基于链的概念。链使用/usr/sbin/ipchains 工具进行配置。

如果系统中不存在这个工具,必须安装 Linux 光盘 RedHat/RPMS 目录中的 ipchains-1.3.9-17.rpm 文件。

ipchains 工具在内核数据包过滤部分中增加或删除规则。内核数据包过滤中的规则都属于一组被称为链的规则。初始时,内核包含3个缺省的链:

- · 输入链 对到达的数据包起作用。
- · 输出链 对将要离开主机的数据包起作用。
- · 转送链 对已经通过输入链,但它们的目标主机是另一台主机的数据包起作用。

当数据包进入一个链时,它将被链中所有的规则检测。如果通过检测,它将被送给规则的目标。如果到达链的末尾,它将送到链的缺省目标。目标为数据包和一定的规则匹配 之后或到达链的末尾时要进行的操作。共有6个系统目标,如表15.1所示。

名称	功能
ACCEPT	允许数据包通过
DENY	丢弃数据包
REJECT	丢弃数据包并通知发送者
MASQ	在转送链中调用时有效。伪装数据包
REDIRECT	在输入链中调用时有效。将数据包发送到防火墙主机的端口中,而非它真正的目
	的地。可能是由于端口声明将数据包重定向到一个新端口而非它的目的地端口
RETURN	迅速传递到当前链的末端。数据包将由链的缺省目标控制

表 15.1 ipchains 的系统对象

目标也有可能为另一个链,包括用户自定义的链。可以随时定义自己的链,并将它们 附加到预定义的链。用户定义的链的缺省目标为调用用户链的规则后的规则。所以用户定 义链可以被用作子程序,并且可以在很多链中使用。这样可以减少规则的数量并使得文档 更容易被读懂,因为可以将链命名为它们要做的事情。

15.4.1 使用ipchains创建防火墙

概括地讲,防火墙可以有两种策略:几乎开放和几乎封闭。在一个几乎开放的防火墙 系统中,除非声明,系统将允许所有的数据包通过。一个几乎封闭的防火墙正好相反,除 非声明允许,系统将拒绝所有的数据包。第二种方法中的防火墙通常被认为是更安全的。 在一个几乎封闭的防火墙系统中,首先将所有链的缺省目标都设置为 DENY 或 REJECT, 并为允许通过的数据包设置 ACCEPT 规则。一个几乎开放的防火墙系统显然正相反。它首 先将所有链的缺省对象都设置为 ACCEPT,并为拒绝通过的数据包设置 DENY 或 REJECT 规则。

要为每个允许通过的数据包增加规则。使用带有数据包特征和操作参数的 ipchains 程

序来设置规则。如表 15.2 和表 15.3 所示为 ipchains 的命令和通常使用的选项。

表 15.2 ipchains 命令

命令	操作
-A chain	向链中增加规则
-D chain [rulenum]	从链中删除号为 rulenum 的规则。如果 rulenum 被省略 , 缺省为第一条规则
-I chain [rulenum]	在 rulenum 号规则前加入规则
-R chain rulenum	在链中替换号为 rulenum 的规则
-F chain	清除链。和对链中所有规则使用-D 选项的效果一致
-L chain	显示链中的规则
-N chain	建立新的用户定义链
-X chain	删除用户定义链
-P chain target	将链的缺省目标设置为 target

表 15.3 ipchains 常用选项

选项	说明
-s [!] address [/mask] [!] [port[:port]]	数据包的源地址和端口
-d [!] address [/mask] [!] [port[:port]]	数据包的目的地址和端口
-i [!] interface	数据包进入(在输入链中)或离开(在输出或转送链中)的接口
-p [!] protocol	数据包协议。可以是文件/etc/protocols 中声明的任一种
-j target [port]	数据包的目标
[!] -y	数据包是 SYN 数据包。只针对声明为-ptcp 的规则
icmp-type type	ICMP 类型为 type。只针对声明为-picmp 的规则
- 1	将对数据包的记录放入 syslog。numeric 或-n 选项和-L 选项一起
	使用将得到主机的 IP 地址和端口地址而非主机名

上述的一些选项中支持!(not)标志。not标志可以用来否定或反转条件。如规则-p!icmp将和没有使用 ICMP 协议的数据包相匹配。

可以有很多方法声明源地址和目标地址。为声明一个主机,可以使用它的 IP 地址或主 机名。声明一个网络可以使用 CIDR 表示,或用 4 个由点隔开的数字来表示。所以 1.1.1.0/24 和 1.1.1.0/255.255.255.0 是相同的。源端口号和目标端口号可以用数字声明或用服务名 (/etc/services)映射。也可以使用用冒号(:)隔开的起、止端口表示一定范围内的端口。

当允许 TCP 的连接只是单方向的时候要使用-y 选项。如在创建一个防火墙系统时,可 能要求内部主机可以访问外部网络而不允许外部网络访问内部主机。但连接需要数据包可 进可出,禁止从网络外部输入的所有数据包是不合适。

解决方案是阻塞那些需要初始化连接的数据包。这些数据包被称为 SYN 数据包,因为 在它们的头部设置了 SYN 标志而 FIN 和 ACK 标志被清除了(这些标志是数据包头部的一 些位设置)。如果阻塞了任何带有 SYN 位设置的数据包,那么内部主机就可以和外部交谈 却不允许外部主机初始化到内部主机的连接。 网际控制报文协议(ICMP)用来控制消息。如 host not found 或 ping 的回应。禁止大部分的 ICMP 消息通常会使网络更安全。当然,一些消息类型是各种工具所需要的或是系统的一部分。--icmp -type 选项使用指定的 ICMP 消息类型来匹配这些数据包。例如,ICMP 消息如 dest -inationunreachable 广泛地被 TCP 和 UDP 使用。也需要允许内部用户使用 ping 或 traceroute 这样的工具。如表 15.4 所示是通常使用的 ICMP 消息类型。

表 15.4 常用的 ICMP 消息类型

数字	名称	需要这些消息的工具或情况
0	echo-reply	ping
3	destination-unreachable	任何 TCP/UDP 通信
5	redirect	寻找路由但路由守护程序并没有运行
8	echo-request	ping
11	time-exceeded	traceroute

15.4.2 代理服务

基本的数据包过滤防火墙在补充了代理服务之后可以增强其安全性,并且缓存数据可 以减少网络带宽的使用。

当今在 Internet 中最普通的代理服务为 Web 缓存代理服务。Linux 中包含名为 Squid Web 缓存代理服务软件。使用 rpm -q squid 命令查看它是否已被安装。如果没有安装,必须安装 Linux 光盘 RedHat/RPMS 目录中的 squid-2.3.STABLE4-1.i386.rpm 文件。

为了配置 Squid,到目录/etc/squid 中并拷贝 squid.conf.default 文件到 squid.conf。编辑 squid.conf 文件来适应自己的需要。文件中有很多的注释,其中的选项是很容易看懂的。

文件中还提供了切合实际的缺省选项,可以不更改它们。如表 15.5 所示,列出了在使用 Squid 之前需要修改的参数。

选项	含义
cache_dir dirname mbytes level1	缓存存储在 dirname 中。它最多可以占用 mbytesMB 磁盘空间。
level2	dirname 包含 level1 个一层子目录,它们又包含 level2 个二层子
	目录。它们的缺省值分别为 100、16 和 256。只需要修改第一个
	值就可以改变缓存可以占用的磁盘空间大小 , 剩下的两个参数
	是用来做调整用的
pid_filename filename	创建 filename 文件来存储 Squid 进程 ID。这个 ID 是当接到 stop
	参数时, /etc/rc.d/init.d/squid 脚本杀死进程时使用的。需要把本
	行的注释符去掉并保留缺省值不变
logfile_rotate nfiles	当使用循环日志时,保留 nfiles 个存档文件。虽然 Squid 的安装
	程序将建立文件/etc/logrotate.d/squid 来控制日志的循环记录,但
	存档拷贝的数量是在这里更改的

表 15.5 需要在文件/etc/squid/squid.conf 中修改的选项

	(续表)
选项	含义
ftp_user user@domain.name	一些 FTP 站点需要有效的 email 地址作为口令。需要修改这个
	参数使其包含一个有效的用户 ID(通常为站点管理员)
cache_mgr user@domain.name	站点管理员的 email 地址。当缓存出现问题时,这个人将收到
	email
cache_effective_user username	Squid 通常是在启动时由 root 运行/etc/rc.d/init.d/squid 来启动的。
	以 root 用户运行 Squid 并不安全,所以必须在/etc/passwd 中创建
	一个新用户并将新用户的用户名和组名写入 squid.conf 文件中。
	Squid 将把脚本的有效用户和组的 ID 设置为刚配置的用户的 ID
	和组的 ID
cache_effective_group groupname	参见 cache_effective_user
err_html_text html_text	这个选项应该被更改为包含 HTML 代码,包括管理员的 email
	地址。它将被 Squid 加在给用户的所有错误页之后

一旦配置完 Squid,就可以使用/etc/rc.d/init.d/squid start 命令来启动 Squid。在它第一次 启动的时候会比较慢,因为它要建立自己的缓存目录结构。Squid 将把错误信息记录到 /var/log/squid/squid.out 中,所以可以使用 tail -f /var/log/squid/squid.out 来检测是否有启动错 误。

第16章 管理硬盘和文件系统

本章主要介绍硬盘和文件系统管理。内容包括设备的概念和管理、文件系统的概念和 管理、文件系统的使用、文件系统的组织以及文件权限管理等。

16.1 设备的概念

在 Linux 的设计中,最简单也最优秀的特性之一是把所有内容都表示为文件,甚至用 来存放文件的设备也被表示成了文件。硬件设备与提供文件接口的驱动器相关联,代表硬 件设备的特殊文件保存在/dev 目录中。

Linux 能够识别两类不同的设备:随机访问块设备(如磁盘)和字符设备(如磁带和串 行线路),还有一些既是串行线路又是随机访问的设备。已获支持的每个设备在文件系统中 都以一个设备文件的形式表达出来。在读或写一个设备文件时,数据就来自或发送到它所 代表的设备中。通过这种方式不再需要特定的程序来访问设备。例如,如果打算向打印机 发送一个文件,只须输入如下命令:

cat filename > /dev/lp1

文件内容就打印出来了。但是,这样使用打印机并不好。一般都要用一个特殊的程序 lpr 来发送准备打印的文件。该程序保证一次只打印一个文件,只要当前文件打印一结束, 就自动把下一个文件送入打印机。

由于设备以文件的形式保存在文件系统中的/dev 目录下,所以,利用 ls 命令马上就能 知道有哪些设备文件。在 ls -1 的输出中,第1 列中包含文件类型及其访问权限。例如,要 检查系统所提供的串行设备,可以输入如下命令:

ls -1 /dev/cua0

crw-rw-1 root uucp 5, 64 Nov 30 1993 /dev/cua0

第1列中的第1个字符 c 表明了该文件的类型,它表示该文件是一个字符设备。对于 普通文件,第1个字符是-;对目录则是 d;对块设备则是 b。通常情况下,所有设备文件 都存在,即使设备本身可能还没有安装。所以,如果系统有一个文件/dev/sda,并不意味着 真的有一块 SCSI 硬盘。拥有所有设备文件是为了使安装程序更为简单,便于增加新的硬件 设备。

16.2 各 类 设 备

Linux 系统把它管理的硬件设备分为字符设备和块设备。

字符设备指的是能够从它那里读取成字符序列的设备。例如,从键盘输入的按键序列 或通过串行线路发送的字节序列。块设备指的是用来存储数据并对它的各部分内容提供同 等访问权的设备。例如软盘和硬盘就是块设备。

利用字符设备必须按照数据发送的顺序从串行线路上获取数据。利用块设备可以从硬 盘的任何随机位置获取数据。字符设备有时又被称为顺序访问设备;块设备有时又被称为 随机访问设备。

当对某个文件执行一定的操作时,内核可以根据此文件的文件模式辨别出它是否是一 个设备。每个设备节点可以通过它具有的不同主次设备号选行区分。主设备号用来向内核 表明这一设备节点所代表的驱动器的类型。如主设备号为3的块设备是 IDE 磁盘驱动器; 主设备号为8的块设备是 SCSI 磁盘驱动器。硬件实例由次设备号的值来表示。如次设备号 为0的 SCSI 磁盘代表整个也可以说是第1个 SCSI 磁盘,次设备号为1~15的磁盘代表此 SCSI 磁盘上的15个分区。利用 ls 命令可以显示出磁盘的主次设备号,如下所示:

#ls -l /dev/sda*

在每一行中最左端的b表明这些设备都是块设备。主设备号和次设备号位于时间字段的 前面,它们之间由逗号分开。在对一般文件的ls-l输出中,这一位置通常显示的是文件的大 小。

如果只有一个数据文件要存储,可以直接把它存放在某个块设备上供以后读取。块设备在大部分方面的表现类似于普通文件,所不同的是普通文件具有由它包含的数据量所确定的文件长度,而块设备的"长度"则是它的整个容量。如果在一个100MB的块设备上写入了1MB的数据,那么在读取它的内容时,将会得到1MB数据,并且还有它以前的99MB内容。需要注意的是,有些Linux工具是针对于数据文件中可用数据的数量而不是文件的整个长度进行编码的,因此,适合于直接在块设备上存储数据。例如,:tar和 cpio 命令适合于所有人使用,而 dump 命令需要对备份数据底层的块设备进行读取,所以只适合于系统管理员。

16.2.1 硬盘

每个硬盘都是以一个独立的设备文件来表示的。一台计算机上通常可以有 2~4 个 IDE 硬盘。它们分别命名为/dev/hda、/dev/hdb、/dev/hdc 和/dev/hdd。SCSI 硬盘则命名为/dev/sda 和/dev/sdb 等等。硬盘容量一般比较大,这使得它能够在不同位置存放不同的文件系统。

对硬盘进行分配的方案称为分区。每个硬盘上一般具有多个不同的分区。在硬盘的分 区表中,保存有硬盘如何分区的简要信息。可通过硬盘的设备文件来访问整个硬盘,这与 分区无关。在访问设备文件时要格外小心,注意保护分区和分区内的数据。

第1个IDE硬盘上的分区具有以/dev/hda开头的名字,跟在这些前缀后面的是分区的编 号。例如,第1个分区叫做/dev/hda1,其他分区类推。在每一个扩展分区内有另一个分区表, 这使得扩展分区能够以同样的方式被划分为4个逻辑分区。扩展分区之外的分区有时又被称 为主分区。 16.2.2 软盘

软盘是可移动的低容量的存储介质。作为存储设备,它比硬盘要慢得多,但它具有可 移动和便于传输数据的优点。

相应于软盘的块设备都以字母 fd 开始,/dev/fd0 是第1个,其他软盘的编号逐步增大。 软盘有许多可能的格式,内核需要知道软盘的格式才能够正确地读取它。所以软盘需 要同时运行几个设备文件,每个设备文件都对应于驱动器及特定软盘类型的一个组合。例 如,/dev/fd0 H 1440 表明是第1个软盘驱动器(fd0),它是一个 3.5 英寸的驱动器,使用 3.5 英寸的软盘(H),软盘容量为 1440KB(1440)。

Linux 设计了一种特殊的软盘设备类型,它能自动侦测驱动器内插入的软盘属于何种类型。这些自动设备叫作/dev/fd0和/dev/fd1等等。自动设备会试着读取一张新插入的软盘的 第1个扇区,并与不同的软盘类型对照,直到找到相符的那一种为止。

除全部标准类型之外,Linux 也有能力支持许多非标准的软盘格式。其中有一些要求使用特殊的格式化程序。可以参考/etc/fdprm 文件,该文件指定了 setfdprm 程序能够识别的设置。

格式化软盘的过程将彻底破坏软盘上的数据,因为它需要对设备本身进行处理。格式 化软盘还需要具有 root 权限,它的执行方法如下所示:

#fdformat /dev/fd0H1440

在对一张软盘格式化后,请不要忘记利用 mkfs 在它上面创建一个文件系统。当系统中运行 X Window 时,可以使用 mount tool 通过点击按钮来完成同样的任务。

16.2.3 CD-ROM

和硬盘驱动器相比,CD-ROM 驱动器的速度一般要慢一些。在安装新软件的时候, CD-ROM 是一个很好的载体,因为安装文件的长度一般都很大,而且速度在安装时似乎并 不是最重要的。

CD-ROM 驱动器从根本上讲是另一种类型的只读设备,它与其他块设备的安装方式相同。CD-ROM 一般包含标准的 ISO 9660 文件系统和一些可选的扩充。一旦安装了 CD-ROM 后,它的表现就同其他任何只读文件系统完全一样。

要安装 CD-ROM 驱动器,可利用下面的 mount 命令:

mount /dev/cdrom

在 Red Hat Linux 中,目录/mnt/cdrom 是安装 CD-ROM 通常用的位置,因为有些软件 包将在这一位置查找所需的内容。

设备名/dev/cdrom 通常被用作为一个符号链接相应于 CD-ROM 的实际设备名,这是因为在 CD-ROM 驱动器变得可以被 PC 机使用时,还没有可供它们使用的廉价标准接口,各个厂商之间选择和设计的接口方案也互不兼容。正因为如此,在 Linux 内核中有大约十几个不同的驱动程序供 CD-ROM 驱动器使用。

16.2.4 磁带

磁带必须顺序读写,不能像软盘和硬盘一样随机读取。所以它的读取速度比后者慢得 多。用于在磁带上读取数据的设备是磁带机或磁带驱动器。但磁带的价格相对便宜,而且 保存的时间更久,其中可保存大量的数据,所以磁带非常适合于归档和备份这些不过分追 求速度但需要低廉而大存储能力的任务。

16.2.5 并行端口

并行端口一般用来和打印机进行通信,另外它还支持其他一些设备,例如 Zip 驱动器、 CD-ROM 驱动器,甚至网络。

这一硬件本身提供一次一个字符式的通信。并行端口可以提供一个中断来通知内核它 正准备输出一个新的字符。但由于打印机在多数 PC 机上通常不是关键,因此这一中断经 常被借给其他一些硬件使用。这便会并生一个不良的结果:内核经常需要轮询并行硬件, 因此驱动一个并行打印机所需的 CPU 工作量经常要多于它本该需要的工作量。

如果并行打印机的中断没有被其他硬件使用,可以利用打印机驱动程序的配置程序 tunelp 来配置它。tunelp 的-t 选项可用来设置每个打印机设备所使用的 IRQ。例如可以像下 面这样把打印机端口的 IRQ 设置为 7:

#/usr/sbin/tunelp /dev/lp1 -i 7

在 Red Hat Linux 下测试打印机端口最好的方法是:利用控制面板上的打印机配置工具。在这一工具中的 Test 菜单中,提供了把测试页直接打印到设备而不通过一般打印系统的选项,这是使用打印机的一个很好的起点。

16.2.6 CD-ROM刻录机

CD 只能够被普通 CD-ROM 驱动器读取,而 CD-ROM 刻录机可以向 CD 写入数据。

CD 的写入过程分为两步:CD 映像的生成和映像向介质的写入过程。

一般的 CD-ROM 的表面只能被写入一次。因此如果 mkisofs 像其他 mkfs 工具那样工作,便总会产生一个表示空 CD 的映像文件。由于这一原因, mkisofs 在生成映像文件时,把文件组装在文件系统中。

CD 映像文件由 mkisofs 程序生成, mkisofs 创建 ISO9660 文件系统的结构并把来自目 录树的文件安装在这一结构中。CD 与块设备不同, 它是不可写的, 这正是它们为什么不是 真正的块设备的原因。映像文件必须利用一个特殊的程序 cdwrite 写入到 CD-ROM 中。 cdwrite 能够理解用来驱动 CD 刻录机的各种专有模式。

当刻录 CD 时,向刻录机写入的数据流必须要跟得上刻录机运行的速度,否则,如果 刻录机用完了要写入的数据,这张 CD-ROM 就会被浪费。因此,通常使用 mkisofs 生成一 个映像文件,然后单独地利用 cdwrite 把这一映像文件写到 CD 刻录机中。

利用一个管道把数据从 mkisofs 直接发送到 cdwrite 也是可以的。这通常需要满足下面 两个条件之一才能正常工作:要么机器的速度非常快,能够确保 mkisofs 提供的数据足以 满足 CD 刻录机的需求;要么是 CD 刻录机对欠载运行不太敏感。不过并不建议使用这一 技术,因为中间映像的产生有其他一些好处,它使用户能够在数据最终写入之前测试 CD 映像。

16.3 磁 盘 管 理

磁盘管理过程中的基本任务是格式化磁盘、分区和管理文件系统。

格式化磁盘为磁盘的使用做好准备。如检查坏磁道(目前许多硬盘都不需要格式化)。 硬盘需要进行分区。其原因之一是要在同一个硬盘上保存不同的操作系统。另一个原 因是把用户文件和系统文件分开,从而简化备份,并有利于防止系统文件受损。

在每个硬盘的分区上要制作恰当的文件系统。如果没有文件系统,对于 Linux 硬盘是 毫无意义的。有了文件系统,才能在它的基础上创建并访问文件。

装入不同的文件系统形成单一的树形结构,根据需要自动或手动装入文件系统(手动 装入的文件系统通常也需要手动卸载)。

16.3.1 格式化

格式化是指为了使磁盘能够存储数据而对其进行初始化的过程,它用于标记磁道和扇 区。在格式化磁盘之前,其表面完全只是一些磁信号。格式化之后,一切就开始井然有序了。

格式化磁盘通常需要一个特殊的程序,在 BIOS 上的格式化程序和以 MS-DOS 程序的 方式提供格式化程序通常不能用于 Linux。

软盘的格式化是用 fdformat 进行的,软盘设备文件是作为参数给出的。fdformat 还将 校验软盘,也就是说,检查它是否有坏扇区,它将多次尝试修复坏扇区。如果软盘只有少 许损伤,fdformat 是可以应付的;如果是真的损坏,校验进程就会被终止,内核将打印出 它找出的各个 I/O 错误的日志消息,这些消息将发送到控制台。如果正在使用 syslog(系统 日志),这些消息就会被发送到/usr/log/message 文件。

badblocks 命令可用来搜索任何磁盘或分区上的坏块。它不格式化硬盘,所以还可用于 检查现成的文件系统。下面的示例是对一张 3.5 英寸的软盘进行检查,这张盘有两个坏块。

badblocks /dev/fd0H1440 1440

718

719

badblocks 输出找出的坏块编号。许多文件系统都可消除这样的坏块,它们维护了一个 已知坏块列表,该列表是在制定文件系统时被初始化的,而且后期还可修改。最初的坏块 搜索可以用 mkfs 命令来进行,这个命令对文件系统进行了初始化,但后期的检查则应该由 badblocks 来进行,对新增块的检查由 fsck 来进行。

16.3.2 分区

分区是指内存或存储设备中一个逻辑上不同的部分,其功能类似于物理上的独立的单 元。

一个硬盘可分为若干个分区。每个分区可用做独立的用途,也就是说,如果只有一个 硬盘,但又想装两个或更多的操作系统,就可以把硬盘分为两个或更多的分区。每个操作 系统可在同一个硬盘上和平共处。如果没有分区,就必须为每个操作系统提供一个硬盘。 软盘和光盘通常不能进行分区。

1. 主引导记录、引导扇区和分区表

与硬盘分区方式有关的信息保存在其第一个扇区内(也就是第一个硬盘上的第一个磁 道上的第一个扇区)。第一个扇区就是硬盘的"主引导记录"(MBR),这是计算机初次启 动时,BIOS读取并启动的扇区。主引导记录中包含一个小程序,这个小程序读取分区表, 查看哪个分区是活动的,并读取该活动分区的第一个扇区,也就是该活动分区的引导扇区。 这个引导扇区中包含另一个小程序,该程序读取存储在这个分区上的操作系统中的第一部 分并开始启动操作系统。

分区方案不是内置于硬件中的,也不是内置于 BIOS 中的。它只是许多操作系统都遵 循的一种约定。并非所有的操作系统都遵循这一约定。有些操作系统支持分区,但它们会 在硬盘上占用一个分区,并在这个分区内采用自己的内部分区方案。这种操作系统能和其 他操作系统和平共处(包括 Linux 在内),而且不需要任何特殊的标准,但对一个不支持分 区的操作系统来说,它是不能和任何操作系统同处于一块硬盘上的。

2. 扩展分区和逻辑分区

扩展分区允许把一个主分区分为若干个子分区,下分的主分区就被称为扩展分区,子 分区就是逻辑分区。子分区的行为和主分区的行为类似,只是创建方式不同而已。

硬盘的分区结构如图 16.1 所示。图中硬盘被分为 3 个主分区,其中的第 2 个主分区被 分为两个逻辑分区。硬盘上的某些部分根本就没有被分区。这个硬盘和每个主分区各自都 有一个引导扇区。



图 16.1 硬盘分区结构

3. 设备文件和分区

每个分区和扩展分区都有自己的设备文件。这些文件的命名约定就是在整个硬盘名称 后面再加一个分区号。命名约定中,1~4 预留给主分区 5~8 预留给逻辑分区。例如 /dev/hda1 指的是第 1 块 IDE 硬盘上的第一个主分区,而/dev/sdb7 指的是第 2 块 SCSI 硬盘上的第 3 个扩展分区。

4. 分区类型

分区表内有一个标识分区类型的位元组,每个分区一个,借此标识使用该分区的操作 系统。其目的是尽可能避免两个不同的操作系统争用同一个分区。虽然没有一个固定的标 准来指定位元组每一位的含义,但有些定义已经得到大家的认可,如表 16.1 所示,它取自 Linux fdisk 程序。

标志	分区类型	标志	分区类型	标志	分区类型	标志	分区类型
0	Empty	17	Hidden	5c	Priam Edisk	a5	BSD/386
			HPFS/NTF				
1	FAT12	18	AST Windows	61	SpeedStor	аб	OpenBSD
			swa				
2	XENIX root	1b	Hidden Win95	63	GNU HURD or	a7	NEXTSTEP
			FA		Sys		
3	XENIX usr	1c	Hidden Win95	64	Novell Netware	b7	BSDI fs
			FA				
4	FAT16 <32M	1e	Hidden Win95	65	Novell Netware	b8	BSDI swap
			FA				
5	Extended	24	NEC DOS	70	DiskSecure Mult	c1	DRDOS/sec
							(FAT-
6	FAT16	3c	PartitionMagic	75	PC/IX	c4	DRDOS/sec
							(FAT-
7	HPFS/NTFS	40	Venix 80286	80	Old Minix	сб	DRDOS/sec
							(FAT-
8	AIX	41	PPC PReP Boot	81	Minix / old Lin	c7	Syrinx
9	AIX bootable	42	SFS	82	Linux swap	db	CP/M / CTOS
							/.
a	OS/2 Boot	4d	QNX4.x	83	Linux	e1	DOS access
	Manag						
b	Win95 FAT32	4e	QNX4.x 2nd part	84	OS/2 hidden C:	e3	DOS R/O
c	Win95 FAT32	4f	QNX4.x 3rd part	85	Linux extended	e4	SpeedStor
	(LB						

表 16.1 Linux 可以识别的分区类型

(徳美)

							(沃代)
标志	分区类型	标志	分区类型	标志	分区类型	标志	分区类型
e	Win95 FAT16	50	OnTrack DM	86	NTFS volume set	eb	BeOS fs
	(LB						
f	Win95 Ext'd	51	OnTrack DM6	87	NTFS volume set	f1	SpeedStor
	(LB		Aux				
10	OPUS	52	CP/M	8e	Linux LVM	f4	SpeedStor
11	Hidden FAT12	53	OnTrack DM6	93	Amoeba	f2	DOS secondary
			Aux				
12	Compaq	54	OnTrackDM6	94	Amoeba BBT	fd	Linux raid auto
	diagnost						
14	Hidden FAT16	55	EZ-Drive	9f	BSD/OS	fe	LANstep
	<3						
16	Hidden FAT16	56	Golden Bow	a0	IBM Thinkpad hi	ff	BBT

5. 对硬盘进行分区

用于建立和删除分区的程序有许多,大部分操作系统都有自己的、用于建立和删除分 区的程序,建议采用各操作系统自己的程序。这类程序大多被称为 fdisk,Linux 及其变体 也不例外。cfdisk 命令类似于 fdisk,但前者有一个更好的用户界面。

在使用 IDE 硬盘时,引导分区必须完全包含在前 1024 个柱面内。这是因为硬盘是在 启动期间通过 BIOS 来使用的,而 BIOS 处理的柱面不能超过 1024 这个限制。有时,也可 能采用有一部分在前 1024 个柱面内的引导分区。但前提是 BIOS 读取的所有文件都必须在 1024 个柱面内。但这是很难做到的,所以不是上上之策。因此,请保证自己的引导分区完 全在前 1024 个柱面内。

事实上, BIOS 和 IDE 硬盘的有些版本能够对多于 1024 的柱面进行处理。如果有这样的系统,可以不管前面提到的限制;如果不能保证,最好把它存放在前 1024 个柱面内。

6. 分区方案

传统做法是:拥有一个相对较小的 root 文件系统,该文件系统内包含/bin、/etc、/dev、 /lib、/tmp 以及其他启动并运行系统所需要的部分。在这种作法中,root 文件系统就是启动 系统所需要的一切。这种做法的依据是:如果 root 文件系统较小,而且用得不是很多,那 么在系统崩溃时,它受损的可能性就越少,用户就能迅速地查出系统崩溃的原因。然后, 再为/usr(用户的根目录,通常在/home 下面)下面的目录树和交换空间建立独立的分区, 或者采用单独的硬盘。把根目录单独保存在其分区内有利于备份,因为通常情况下,没有 必要备份程序。

采用多个分区有一个问题:它把未使用的硬盘空间分成许多更小的部分,在硬盘和操 作系统可靠的情况下,可能只用一个分区来容纳他们的所有文件,而且不采用多个分区还 可减少备份的麻烦。

对于小硬盘,最好的办法是只用一个分区;对于大硬盘,最好有少数几个大分区。如

果有若干个硬盘,就可以把 root 文件系统(包括/usr 在内)装入一个硬盘,把用户根目录 装入另一个硬盘。

16.4 文件系统简介

本节将介绍一般的文件系统的特点和 Linux 文件系统的特点。

16.4.1 文件系统的概念

在操作系统中,文件命名、存储和组织的总体结构就称为文件系统。一个文件系统内包括文件、目录以及定位和访问这些文件和目录所需的信息。它也可以表示操作系统的一部分,把应用程序对文件操作的要求翻译成低级的、面向扇区的、并能被控制磁盘的驱动程序所理解的任务。它还可以用来指代用于保存文件或文件系统类型的分区或磁盘。

文件系统中包含一些重要的数据,少数程序直接在磁盘或分区的原始扇区上操作,如 果这些扇区上有现成的文件系统,这个文件系统就会遭到破坏或严重损伤。大多数程序都 是在文件系统上操作的,因此,不能在没有文件系统的分区上运行。

分区或磁盘可用作文件系统之前需要进行初始化,而且管理操作数据结构也需要写入 磁盘。这个过程就叫作"制定文件系统"。

16.4.2 文件系统的种类

Linux 支持许多不同的文件系统,使用最多的文件系统是 ext2 文件系统,因为它是专 门为 Linux 而设计的,并且非常高效,支持其他文件系统是为了与其他系统兼容。在软盘 上一般使用 msdos 和 vfat 文件系统。在 Red Hat Linux7.0 中,一些文件系统被编译在内核 中。

还有一些文件系统可以作为可装载的模块被使用。

几个重要的Linux支持的文件系统及其特点如表16.2所示。

文件系统	特点
Minix	
Xia	Minix 文件系统的修订版,对文件名和文件系统长度的限制放宽,但没有引入新的特性。
	应用不广,但表现相当不错
ext2	Linux 文件系统中最富有特性的一个 , 也是当前最流行的文件系统之一。它的向上兼容
	性很好 , 所以有些文件系统代码的新版本不需要重新制定现有的文件系统
Ext	ext2 的早期版本,它不能向上兼容
MSDOS	与 MSDOS、OS/2 以及 WindowsNT 兼容的 FAT 文件系统
umsdos	对 Linux 下面的 msdos 文件系统驱动程序进行扩展,从而获得长文件名、拥有者、访问
	许可、链接和设备文件。这样一来,就允许把一个普通的 MSDOS 文件系统用作 Linux
	文件系统,从而无需再为 Linux 分配一个独立的分区

表 16.2 Linux 支持的主要文件系统及其特点

	(续表)
文件系统	特点
iso9660	标准的光盘文件系统;常见的光盘标准文件系统扩展是 RockRidge,它允许自动支持长
	文件名
Nfs	网络文件系统,允许多台计算机共享一个文件系统,以便能通过其中一台连网计算机轻
	松地访问文件
Hpfs	OS/2 文件系统
Sysv	SystemV/386、Coherent 和 Xenix 文件系统

选择什么样的文件系统,要根据具体情况而定。选择依据是管理操作结构的开销、速度、可靠性、兼容性和其他各种因素。文件系统的选择需要具体情况具体决定。目前,ext2 是最常用的文件系统,而且它也可能是最明智的选择。如果能够自由选择文件系统,请使 用 ext2 文件系统,因为它具有丰富的特性,而且不会对性能产生影响。如果由于兼容性等 原因,在不得已的情况下再使用其他的文件系统。

另外,还有 proc 文件系统,它通常作为/proc 目录供人们访问。该文件系统实际上根本 不是文件系统。proc 文件系统能够使访问特定的内核数据结构更为方便。它使这些数据结 构看起来像一个文件系统,而这个文件系统可以利用所有的常用文件工具来进行处理。

proc 文件系统虽然被称为文件系统,但它并不存在于磁盘上。它只存在于内核的映像中。只要想查看 proc 文件系统的某个部分,内核就会令这部分看起来就像保存在磁盘上的 某个地方一样。所以,即使有一个长达几 GB 的/proc/kcore 文件,它也不占用任何磁盘空 间。

16.4.3 文件类型

Linux 文件系统中包括有以下类型的文件:文本文件、二进制文件、目录文件、连接文件、设备文件和管道文件。

1. 文本文件

Linux 文件系统中的文本文件同其他操作系统的文本文件相同,是由一些在计算机内的 ASCII 码组成的。通常这类文件在各个操作系统中是兼容的。

2. 二进制文件

二进制文件中保存的是数据的二进制表示。一般情况下用普通的文本阅览器所见到的 是一堆乱码,需要专用的软件才能查看文件内容。例如:可执行文件、图像文件、声音文 件等都是这类文件。

3. 目录文件

Linux 中对目录的管理也是通过文件进行的。顾名思义,目录文件就是一类特殊的文件, 其中保存的信息不是普通的数据,而是文件名和与文件有关的文件信息。例如文件名、文件创建日期、文件类型和读取权限等。目录文件中还可以包含下一级的目录。

4. 连接文件

连接文件实际上不是文件,它们仅是指向同一索引节点的目录条目,是一个索引节点 表。该表记录了一个文件有多少连接。这种连接不能跨越设备,因为不同设备上的 i 节点 号不同,而另一种连接类型符号链接则没有这种限制。

5. 设备文件

设备文件是 Linux 系统中较特殊的文件。Linux 中引入设备文件的目的是实现设备独立 性。用户访问外部设备就是通过设备文件进行的。操作系统对外设的支持程度直接影响到 操作系统使用的广泛程度,一个成功的操作系统不会由于自身的问题而限制了访问硬件的 能力。面对越来越多、越来越高级的外设,Linux 将外设看作一个文件来管理,这样就避免 了由于外设的增加而带来的问题。当需要增加新设备时,只要在操作系统内核中增加相应 的设备文件即可。因此,设备文件在外设与操作系统之间提供了一个接口,使用外设就像 使用普通文件一样。设备文件存放在/dev 目录下,它使用设备的主设备号和次设备号来区 分指定的外设:主设备号说明设备类型,次设备号说明具体指哪一个设备。例如:/dev/fd0 指系统中的一个软盘驱动器,主设备号 fd 是软盘驱动器的缩写,次设备号0是软盘驱动器 编号,意思是指系统中的一个软盘驱动器,也就是通常所说的 A:盘。

6. 管道文件

管道文件是 Linux 中较特殊的文件类型,这类文件多用于进程间的通信。进程可通过 管道文件实现互相的信息传输,一个进程将数据写在管道的一端,另一个进程从管道的另 一端将数据读出,这样就能实现进程的相互协作。管道文件又可分为无名管道和有名管道 两种。

16.5 使用文件系统

文件系统的使用和文件的操作是不同的。本节介绍 Linux 文件系统的使用,主要介绍 几个文件系统的工具。

16.5.1 创建文件系统

在安装 Red Hat Linux 时,安装过程将创建一些新的文件系统并使用它们。

Linux 提供了一个普通的命令 mkfs, 它可以用来在块设备上创建文件系统。由于 Linux 利用同一套操作来管理几乎所有的资源,因此 mkfs 也可用于在普通文件中生成文件系统。

实际上,每个文件类型都有一个独立的程序来建立系统。mkfs 根据不同的文件系统类型运行相应的程序。文件系统类型的选择是利用-t fstype 选项来完成的。

· -t fstype 选项

选择文件系统类型。

• -c 选项

搜索坏块并相应初始化坏块列表。

· -l filename 选项

从同一个文件内读取最初的坏块列表。

为了在一张软盘上建立一个 ext2 文件系统,可以使用下面这些命令:

#fdformat -n /dev/fd0H1440
#badblocks /dev/fd0H1440 1440 \$>\$ bad-blocks
#mkfs -t ext2 0l bad-blocks /dev/fd0H1440

首先,格式化软盘(-n 选项用来避免校验,也就是避开坏块检查)。然后,badblocks 命令对坏块进行搜索,其输出重新定向到一个 badblocks 文件内。最后,再利用由 badblocks 初始化的坏块列表建立文件系统。

-c 选项也可以随 mkfs (而不是 badblocks)和一个独立的文件一起使用。和单独使用 badblocks 相比,使用-c 选项更为方便,但对于文件系统已经建立好之后的坏块检查, badblocks 又是必不可少的。

在硬盘或分区上制订文件系统的过程和软盘是一样的 , 只不过不需要格式化。

可以向高层的 mkfs 传递选项。例如,-V 选项使它显示所执行的命令,-c 选项使它检 查设备的坏块。mkfs 程序也使用户可以向文件系统特定的 mkfs 传递选项。与文件系统相 关的选项有很多,但它们大多数都有自己合理的默认值,并且一般也不需要改变。用来创 建 ext2 文件系统的 mke2fs 需要传递的选项只有-m 和-i。-m 选项用来指定文件系统为 root 所保留的空间大小(例如,当系统盘被其他方面的内容完全塞满时,它可用来作为工作空 间);-i 选项很少被用到,它用来保持 inode 和磁盘块之间的平衡,它与预期的文件的平均 大小有关。正像前面所说的那样,默认值对于大多数要求都比较适合,因此这些选项只用 在一些特殊情况中。这里,可以看到在软盘上创建并安装了一个 ext2 文件系统,由 mkfs 程序的默认选项所确定的文件系统的结构也被显示出来。此文件系统没有卷标,每个 inode 有 4096 字节(360×4=1440),块的大小是 1KB,有 5%的磁盘空间供 root 用户使用,这 些都是 mke2fs 的默认值。当创建了某个文件系统后,可以利用 dumpe2dfs 命令来显示此 ext2 文件系统的信息。

当在软盘上创建了文件系统之后,可以把它包含在文件系统表/etc/fstab 中,这一任务 可以通过把文件系统表中已有的用来把/dev/fd1 作为 vfat 文件系统来安装的条目修改为如 下内容来实现:

/dev/fd1 /mnt/floppy ext2 user,sync,errors=continue 0 0

前3列分别是设备、安装点和文件系统的类型。这里的 user 选项表明允许用户安装这 一文件系统, sync 选项表明向此文件系统进行写入的程序必须等待到每个写操作完成之后 才能继续进行下去,这看起来似乎是很合理的,但这并不是通常所采用的方式。内核一般 以另外一种方式来管理文件系统的写入从而提高系统的性能,这一方式对于像硬盘这样的 固定设备是十分理想的,但对于像软盘这样的低容量可移动的设备却没有多大的好处。一 般在软盘上只是写入少量的文件,然后利用 umount 命令卸载它并把它拿走,umount 操作 必须等到所有数据写到设备上后才能完成。必须等待是很令人讨厌的,因此当向软盘中拷 贝文件时会有一些危险。为了避免它们,一定要耐心地等待磁盘的指示灯熄灭后再把它拿 走。如果利用异步写入,一些被缓冲的数据有可能不会被写入到磁盘中,因此对于可移动 的介质,同步写入是比较安全的。 ext2 文件系统具有一个针对于错误的配置选项。如果某个 ext2 文件系统遇到错误,可以采用 3 种不同的应付方式:

· 只读方式

对于主要包含非关键性数据的文件系统(例如,/tmp、/var/tmp 或消息假脱机),把它重 新安装为只读方式以便利用 fsck 对它进行修复通常是一种最好的解决方法。

・ 应急方式

不顾系统的配置文件可能已经遭到破坏而继续运行系统是不明智的,因此内核的应急 方式(也就是在控制下停止系统运行或紧急关闭系统)在有些情况下十分合适。

• 对错误忽略的方式

如果因为软盘有一个坏扇区就使系统关闭显然不合理,因此在这种情况下可利用 continue 选项通知内核"忽略错误继续执行"。如果真要这么做时,最好利用-c 选项运行 e2fsck 命令。例如,fsck –t ext2 –c /dev/fd1 将会调用 badblocks 命令对坏的磁盘块进行检测。 当执行这一命令后,e2fsck 将会尽量使错误恢复过来。

16.5.2 装入和卸装文件系统

在使用文件系统之前,必须先装入它,操作系统才能执行管理操作,以保证一切正常运行。由于 Linix 内的所有文件都在一个单独的目录树内,因此,装入操作将令新文件系统中的内容类似于一个现有子目录的内容。这个子目录位于某个已装入了的文件系统中。

要把一个块设备安装到文件系统,可使用 mount 命令。使用 mount 命令需要指定包含 文件系统的设备、类型以及把它安装在目录结构中的位置。

1. mount/umount 命令的使用

mount 命令的格式如下:

mount [-t type] [-o options] device mount-point

device 必须是一个块设备。如果它包含一个冒号,可能是从它那里安装文件系统的另一台计算机的名字。mount-point(装入点,也称为装入目录)应该是一个已经存在的目录, 文件系统将出现在这一位置(这一目录中现有的内容将被隐藏)。文件系统的类型和选项是 可选的,并且选项的种类和含义依赖于被安装文件系统的类型。如果想要安装的文件系统 被指定在/etc/fstab 文件中,那么只需要指示安装点或设备名,其他的内容将从/etc/fstab 中 读取。

装入目录不要求一定是空的,尽管它必须存在。但对其中的任何文件,在文件系统装 入期间将不能按其文件名访问它们。

执行完装入命令之后,文件系统中的内容和装入目录中的内容一样。同样,要想查看文 件系统就应该查看目录内容。注意设备文件和装入目录之间的区别:设备文件提供的是对 磁盘原始内容的访问:装入目录提供的则是对磁盘上文件的访问。

可像下面的示例一样,装入文件系统:

mount /dev/hda2 /home

Linux 支持许多文件系统类型。mount 命令可尽力猜测文件系统的类型。另外也可用-t fstype 选项直接指定文件系统的类型。例如,要装入一个 MS-DOS 软盘,可采用下面的命 令:

mount -t msdos /dev/fdo /floppy

如果不打算在文件系统内写入任何东西,可采用-r 选项进行只读装入。这样一来,就 会防止内核在文件系统内进行写操作尝试,并防止内核更新 inode 内的文件访问次数。只 读装入对不能写的媒体,如光盘是必要的。

root 文件系统通常先以只读方式装入。然后启动脚本运行 fsck,验证其有效性。如果 没有问题,它们就会重装这个文件系统,以便允许对它进行写操作。fsck 禁止运行于一个 已装入的文件系统上,因为 fsck 运行期间,任何对该文件系统的改动都将带来麻烦。由于 root 文件系统是只读装入的,在它接受验证期间,fsck 可解决任何难题。因为重装入操作 将刷新文件系统保存在内存中的任何数据。

在许多操作系统上还有其他的文件系统也应该在启动时自动装入。这些文件系统是在 /etc/fstab 文件内指定的。

当一个文件系统不再需要被装入时,就可以用 umount 来卸装。要卸装前一个示例中的 目录,可像下面这样:

umount /dev/hda2

无论如何,总是需要卸装已装入的软盘。不要以为把软盘从驱动器中取出就万事大吉! 由于磁盘缓冲的原因,在卸装软盘之前,没必要把数据写入软盘,所以过早把软盘从驱动 器中取出可能导致系统内容混乱不堪。如果只能通过软盘读取系统,这样做或许没太大关 系;如果在系统上进行写操作,后果将难以设想。

装入和卸装要求有超级用户权限,也就是说只有 root 才能执行。

2. mount 命令的选项和出错可能

如表 16.3 所示是可用于 mount 的选项及其说明 , 表中选项的顺序是以字母的顺序排列 的。

选项	说明
async	只有在数据到达硬件后,文件系统的写请求才被接受。利用这个选项,程序很快就能 继续执行。这意味着,万一系统崩溃,系统比较容易丢失数据。但从另一方面讲,系 统崩溃很少发生在 Linux 上。这一选项大大加快了 NFS 文件系统,与它相对的选项是
auto	sync 当给定-a 标志时,这一标志向 mount 表明它应该安装此设备。-a 标志一般被启动脚本 使用,以确保在引导时安装所有需要的文件系统。与此选项相对的是 noauto
defaults	打开选项 rw、suid、dev、exec、auto、nouser 和 async
dev	允许使用系统上的设备节点。对设备的访问完全由对磁盘上设备节点的访问权决定

表 16.3 mount 选项

1	(歩主)	`
C	头化)

选项	说明			
exec	它向内核表明,内核应该允许程序在文件系统上运行。此选项更多的是以 noexec 形式			
	出现,它向内核表明程序不允许在系统上运行			
noauto	与 auto 相对的选项,参见 auto			
nodev	与 dev 相对的选项,参见 dev			
noexec	与 exec 相对的选项,参见 exec			
nosuid	与 suid 相对的选项,参见 suid			
nouser	与 user 相对的选项,参见 user			
remount	这一选项允许 mount 命令在不被中断的情况下为已被安装的文件系统改变特征			
ro	把文件系统安装为只读方式。它是与 rw 相对的选项			
rw	把文件系统安装为可读/可写方式。它是与 ro 相对的选项			
suid	允许使 setuserID 和 setgroupID 文件模式位生效。与这一选项相对的是 nosuid。nosuid			
	选项比较常用,其原因与使用 nodev 的原因相同			
sync	所有写操作将使调用程序等待到数据被写入硬件为止。这一操体方式比与它相对的异			
	步 I/O 要慢但却比较可靠。异步 I/O 由选项 async 来表明			
user	允许普通用户安装文件系统。当/etc/fstab 文件中有 user 时,普通用户可以安装或卸载			
	相应的文件系统,用户只要提供设备名或安装点即可,而其他相关的信息将从/etc/fstab			
	文件中获取。由于安全性的原因,user 包含 noexec, nosuid 和 nodey 选项			

下面任何一种情况都有可能导致 mount 命令失败:

• 不正确的设备名

如果指定的是一个不正确的设备名。如设备文件不存在,在内核中没有可用于它的驱动程序,或者对应于它的硬件不存在。

• 不可读的设备

设备不可读的原因包括设备不正确。如空的软盘驱动器或损坏的介质和权限不够。除 通过在/etc/fstab文件中利用选项user授权的文件系统之外,mount命令对于普通用户来说一 般是禁止使用的。

• 不正确的安装点

试图在一个不存在的安装点安装设备也不能使安装成功。

• 其他原因

引起mount失败还有一些可能但未必会发生的原因。如超出了编译内核时设定的可以被 安装的文件系统的数量限制。

3. 使用 user mount tool 加载文件系统

在命令行执行如下命令:

#usermount

可以看到 user mount tool 的使用界面如图 16.2 所示。

使用这一图形化工具可以迅速加载和卸载文件系统。特别是对软盘和 CD-ROM 文件系统更为有效。图 16.2 显示了软盘和 CD-ROM 没有被加载。

User Mount Tool							
Directory	Device	Туре					
/dev/pts	none	devpts	Unmount	Format			
/proc	none	proc	Unmount	Format			
/mnt/floppy	/dev/fd0	auto	Mount	Format			
/mnt/cdrom	/dev/cdrom	iso9660	Mount	Format			
/boot	LABEL=/boot	ext2	Unmount	Format			
1	LABEL=/	ext2	Unmount	Format			
Exit							

图 16.2 user mount tool 的使用界面

user mount tool 从文件/etc/fstab 中得到有关加载的信息。缺省时软盘的文件类型为 ext2。 这个设置对 Linux 是有利的,但却不支持在 DOS 中使用。为了可以在不同的系统中使用软 盘中的内容,可以将/etc/fstab 中软盘的文件类型更改为 msdos,这样在系统切换时,使用 软盘有更大的灵活性。

16.5.3 设置文件系统

至少有两种方法可以更改文件/etc/fstab 中的内容。第一个方法是使用编辑器手动编辑 /etc/fstab。第二个方法是通过使用 fsconf 工具 (/bin/fsconf) 来更改/etc/fstab 文件。下面将 分别介绍。

1. 手工编辑/etc/fstab 文件

文件系统表/etc/fstab 只是一个文本文件,由制表符或空格分为许多列,可以利用任何 文本编辑器对它进行编辑。不过,通过手工修改时一定要特别小心,因为删掉和破坏某一 条目将使系统在下次引导时不能安装该文件系统。

下面是/etc/fstab 的一个例子:

LABEL=/	/	ext2	defaults	11
LABEL=/boot	/boot	ext2	defaults	12
/dev/cdrom	/mnt/cdrom	iso9660	noauto,owner,ro	0 0
/dev/fd0	/mnt/floppy	auto	noauto,owner	0 0
none	/proc	proc	defaults	0 0
/dev/hda10	swap	swap	defaults	0 0
前两个条目是构成 Linux 系统的 ext2 文件系统。当 Linux 引导时,首先安装的是根文件系统,接下来的是所有其他本地文件系统。文件系统在/etc/fstab 中出现的顺序就是它们 被安装的顺序。

接下来的两个文件系统都是可移动的文件系统(一个光驱和一个软驱),它们都具有 noauto 选项设置,因此在引导时不能被自动安装。光驱具有的文件类型是 iso9660,这是光 驱的标准文件类型,软驱具有的文件类型是 auto。

接下来的两个文件系统是特殊的文件系统。第一个文件系统/proc 是由内核提供的一个 特殊文件系统,它是向用户程序提供系统信息的一种方式。文件系统/proc 中的信息用来使 ps、top、xload、free 和 netstat 等这样的实用工具能够正常工作。/proc 中有些文件是十分庞 大的(例如/proc/kcore),但不必担心磁盘空间。文件系统/proc 中的所有信息都是在读取它 时由 Linux 内核在运行时产生的,它们并不消耗磁盘空间。通过一些方法,可以很容易地 辨别出它们不是真正的文件。例如,root 不能利用 chown 命令来处理它们。

最后一个"文件系统"实际上根本不是一个文件系统,这一条目用来指示一个作为交换空间使用的磁盘分区,交换分区用来实现虚拟内存。交换空间也可采用文件,交换文件的名字位于第一列,这通常是设备名的位置。

右边的两个数字列分别与 dump 命令和 fsck 命令的操作相关。dump 命令把第 5 列中的 数字同文件系统从上次备份到现在的天数相比较,从而通知系统管理员对文件系统进行备 份。其他备份软件如 Amanda,也能够以同样目的使用这一字段。没有这个字段的文件系统 被认为是 0,表示"从不转储"。

第 6 列是 fsckpass,它表明在引导时哪些文件系统能够被同时检查。根文件系统总是首先被检查,在它之后,分离的驱动器能够被同时检查。因为 Linux 是一个多任务操作系统,然而在同一时刻检查同一硬盘上的两个文件系统是毫无意义的,因为这将导致过多的磁头移动从而浪费很多时间。具有 1 以上的同样编号的所有文件系统将被同时检查,编号为 0 或没有编号的文件系统根本不被检查。

2. 使用 fsconf 在图形界面中创建和编辑文件系统

fsconf 工具允许编辑、添加或删除加载项。fsconf 的使用界面如图 16.3 所示。

Filesystem configurator
This package allows you to configure which partitions and network volumes your workstation can access
Access local drive
Access nfs volume
Configure swap files and partitions
Set quota defaults
Check some file permissions
Quit Help

图 16.3 fsconf 的使用界面

下面对 Access local drive 选项进行详细介绍。

单击工具中的第一个选项 Access local drive, 弹出的界面如图 16.4 所示, 它显示了当前的本地文件系统。

Local volume						
You can edit, add, Select [Add] to add	or delete I a new de	mounts finition				
Source	1ount poir	FsType	Size	artition typ	Status	
LABEL=/	1	ext2			Mounted	
LABEL=/boot /dev/cdrom /dev/fd0	/boot /mnt/cdrc /mnt/flopp	ext2 iso9660 auto			Mounted	
	0	Quit	Add	Help		

图 16.4 显示当前的本地文件系统

单击 Add 按钮,如图 16.5 所示的 Volume specification 窗口的 Base 选项卡就会显示出来。

Volume specification		_ = ×
You must enter the specificatio or partition and the position (m where you want to install this v in the directory structure of this	n of a volume sunt point) olume workstation	
Base Options Dos options	vlisc.	
Partition	·	7
Туре	[
Mount point	[
Accept Cancel	Del Mount Unmount Help	

图 16.5 Volume Specification 窗口的 Base 选项卡

单击 Partition 和 Type 文本框之后的下三角按钮,就可以在一个可能的列表中选择相应的值。从列表中看到,可以很容易地从一组选项中选取合适的值,而且也并不需要记住是MS-DOS 还是 DOS。

因为 DOS 和 OS/2 是单用户操作系统,而 Linux 是多用户系统,所以在这里有 DOS 选项。在 DOS 和 OS/2 系统中对文件设置拥有权是没有意义的。为了解决这个问题,使得所有用户可以看到其他人的 DOS 文件,Linux 允许用在这里缺省定义的用户和组 ID 来加载整个文件系统。

单击 DOS options 标签,弹出如图 16.6 所示的 Dos options 选项卡。像其他的窗口一样, 这里也提供下拉列表供选择缺省的用户 ID、缺省组 ID 和传输模式,这里提供了 3 种传输 模式的选项: binary、auto 和 text。

Volume specification		
You must enter the specification or partition and the position (mo where you want to install this w in the directory structure of this	n of a volume runt point) olume workstation	
Base Options Dos options N	Aisc.	
default user id		
default group id		
default permission		
translation mode		7
Accept Cancel	Del Mount Unmount Help	

图 16.6 Dos options 选项卡

这些选项为二进制(binary),自动(auto)和文本(text),在 DOS 和 Linux 中对文本 文件的存储是不同的。在 Linux 中行尾的标记符为一个单行符。在 DOS 中行尾的标记符为 一个回车符并跟着一个单行符。这 3 种传输模式使得文件可以从一个类型的系统传输到另 一个类型的系统。二进制方式说明不进行转换。而 Auto 意味着转换那些不是可执行文件、 程序代码、图像、TeX 文件或归档文件的所有文件。可执行文件一般带有 BIN、COM、EXE 或 SYS 后缀。程序代码一般有 APP、DLL、DRV、LIB、OBJ、OVL、OVR 或 PIF 后缀。 图像文件有 BMP、GIF、GL、JPG、PCX 或 TIF 后缀。TeX 文件则以 DVI、GF、PX、PXL、 TFM 或 VF 结尾。归档文件带有 ARC、ARJ、DEB、GZ、LHA、LZH、TAR、TAZ、TPZ、 TZ、TZP、Z、ZIP 或 ZOO 后缀。text 模式表明不分文件类型所有文件都做转换。在该选 项中把一个不需要转换的文件进行转换将会出现问题。

16.5.4 文件系统的管理工具

1.fsck 命令

文件系统非常复杂,而且容易出错。一个文件系统的正确性和有效性是可以检查出来 的,具体做法是利用 fsck 命令。对于找到的任何一个小错误,它都可以对其进行修复,并 提醒用户文件系统是否存在不可修复的错误。

许多系统都被设置为启动时自动运行 fsck,所以在使用系统之前,需要侦测错误。使用损坏了的文件系统将产生不可预料的后果。但是,fsck在一个大型的文件系统上运行时, 花的时间要多一些。如果没有错误,可采用以下两个方法来避免错误检查。

(1) 如果 etc/fastboot 存在,就不执行检查。

(2) ext2 文件系统在其超级块内有一个特殊的标记,用以说明这个文件系统在前一次 装入之后是否正确卸装。这样,如果这个标记指出卸装已完成,就允许 e2fsck(用于 ext2 文件系统的 fsck 版本)避免对文件系统的检查。

自动检查只适合启动时自动装入的文件系统。利用 fsck 手动检查其他文件系统,如软 盘。

fsck 只可运行于未装入的文件系统上,绝不能运行于已装入的文件系统上。这是因为 它访问的是原磁盘,因而可以在操作系统不注意的情况下,修改文件系统。

2. 修复文件系统的命令

有些磁盘数据由于性能方面的原因,在写入到磁盘前暂时被保存在内存中。如果内核 没有机会把这些数据真正地写到磁盘上,文件系统就会遭到破坏。这种情况的发生有如下 几个原因:

在内核完成对数据的写入之前,存储设备(例如软盘)被移走。

系统遭受停电。

· 用户错误地关闭电源或意外地单击 reset 按钮。

作为引导过程的一部分, Linux 将运行 fsck 程序, 它的任务是检查和修复文件系统。 大部分时间里, 引导是在一个正常的系统关闭之后进行的,此时文件系统将在重新引导系 统之前被正确卸载。在这种情况下, fsck 认为这些文件系统是完好的。fsck 之所以知道这 一情况是因为:在卸载所有文件系统之前,内核在文件系统上写入了一个特殊的记号来表 明数据是完好的。当文件系统为了写入数据而再次被安装时,这一记号便被删除。

相反,如果上面所列举的某种灾难发生,文件系统将不被标记为是完好的。这时当 fsck 像平常一样被激活时,它将注意到这一问题并开始对文件系统执行一个完整的检查。如果 对 fsck 指定-f 标志,这种检查也会发生。为了防止一些错误在不知不觉中产生,fsck 也定 期执行一些检查工作。即使文件系统每次都被正确地卸载,也将按照文件系统本身所指定 的时间间隔定期地进行完整的检查。

引导过程首先对根文件系统进行检查,然后再把它加载为可读/可写的方式。 将首先利用下面这一命令检查根文件系统:

fsck -V -a /

接着,将通过执行下面的命令对所有其他文件系统进行检查:

#fsck -R -A -a

当文件系统的破坏很严重时,这一方法将会失败。因为如果没有得到用户的指示,有些操作便无法对文件系统执行。在这种情况下,fsck 将会向它的调用者启动脚本返回一个错误值,启动脚本时将会产生一个 shell,从而使管理员能够以交互的方式运行 fsck。当这种情况发生时,将会产生出错信息并询问 root 口令。

当自动的 fsck 失败时,需要通过指定 root 口令进入系统并手工运行 fsck 程序。在输入 root 口令后,便会出现下面的提示:

(Repaire filesystem)#

可以手工为 fsck 提供一套选项使它检查所有的文件系统。这是一个非常好的方法,如 下所示:

fsck -A -V ; echo==\$?==

fsck 处于"交互的"模式。 echo= =\$?= =命令用来明确地解释 fsck 操作的输出内容。 如果在等号之间打印出的值小于 4,说明一切都还正常;如果这一值是 4 或更大,说明需 要采取进一步的恢复措施。所返回的各个值的意义如下:

- 0 没有错误。
- 1 文件系统的错误已经被纠正。
- 2 系统应该被重新引导。
- 4 文件系统的错误未被纠正。
- 8 操作性的错误。
- 16 用法或语法错误。
- 128 共享库的错误。

如果这一方法还不起作用,很可能是因为超级块被破坏。当 fsck 执行磁盘检查时,如 果超级块被破坏,它便不能执行。由于系统设计非常合理,ext2 文件系统有许多备用超级 块规律地散布在整个文件系统中。假定 fsck 命令宣布它不能清理某一文件系统,例如 /dev/fubar,则可以利用下面的命令通过使用超级块的备份再次运行 fsck 命令:

fsck -t ext2 -b 8139 /dev/fubar

8193 是第一个备用超级块所在块的编号,这一备用超级块位于块组1(第一块组编号为0)的开端。在块组2(16385)和块组3(24577)开端还有一些备用超级块,它们以8192块的间隔被分开。

如果利用与缺省值不同的其他设置创建了一个文件系统,那么这些备用超级块的位置 将有所变化。mke2fs 命令将在运行时列出它所创建的超级块,因此如果不使用缺省值,这 是注意这一问题的一个极好时间。如果 fsck 仍然不能成功地修复文件系统,可以进一步试 用其他方法。不过这些方法很少用到,并且通常说明硬件问题已经非常严重从而妨碍了 fsck 的正常操作。

这些比较极端的措施一般是不可能用到的。手工操作 fsck 在一些确实需要它的特殊情况下,基本上也都能够达到修复的目的。在手工操作 fsck 起作用后,启动脚本所提供的由

管理员使用的 shell 也就完成了它的任务,因此可输入 exit 命令退出这个 shell。此时,为了确保整个系统能够按计划运行,引导过程将再次从头开始启动。在这次运行时,文件系统将不会再有什么差错,系统也就能够正常引导了。

3. badblocks 命令

周期性地检查磁盘中的坏块是个好习惯,这是通过 badblocks 命令来执行的。该命令输 出一份列表,上面列出找到的所有坏块。该列表可以传给 fsck,记录到文件系统结构内, 以便操作系统不再尝试用坏块保存数据。下面的示例解释了整个过程:

badblocks /dev/fd0H1440 > bad-blocks

fsck -t ext2 -l bad-blocks /dev/fd0H1440

Parallelizing fsck version 1.18

e2fsck 1.18 11-Nov-1999 for EXT2 FS 1.18. 11-Nov-1999

Pass 1: Checking inodes, blocks, and sizes

Pass 2: Checking directory structure

Pass 3: Check directory connecrtivity

Pass 4: Check reference counts

Pass 5: Checking group summary information.

/dev/fd0H1440:****FILESYSTEM WAS MODIFIED*****

/dev/fd0H1440:11/360 files,63/1440 blocks

如果 badblocks 报告一个块已被采用, e2fsck 就会试着把这个块移到另一个地方。如果这个块真的坏了,就说明文件的内容也可能被损坏了。

4. 适用于所有文件系统的其他工具

还有一些工具可以用于管理文件系统。df 展示一个或多个文件系统上的剩余磁盘空间, du 展示一个目录及其所有文件占用了多少磁盘空间。它们可用于查找浪费磁盘空间的原因。

sync 把缓冲区内所有没有写入的块写入磁盘。这个过程很少需要手动进行,后台进程 会自动进行此类的更新。它特别适合于灾难性场合,如必须立刻关掉电源和不能等待运行 更新时。

5. 适用于 ext2 文件系统的其他工具

除了可直接或通过不依赖于前端的文件系统类型访问文件系统创建程序(mke2fs)和 检查程序(e2fsck)外, ext2文件系统还有一些有用的其他工具。

tune2fs 用于调整文件系统参数。下面列出其中一些较为有用的参数:

最大装入数

e2fsck 在文件系统已被装入多次时进行检查,即使在清除标记已设定的情况下也是如下。对于用于开发或自行测试的系统,这个数目越小越好。

· 两次检查之间的最大装入次数

e2fsck 也可在两次检查期间实施一个最大装入次数,即使在清除标记已设定而且文件 系统未曾经常性地装入的情况下也是如此。这个参数是可以禁用的。

· 为 root 预留的块数目

ext2 为 root 保留了一些块,以便在文件系统填满的情况下,仍有可能在无须删除任何 东西而进行系统管理。默认情况下,保留的块数大约是总数的 5%,对于大多数磁盘,这样 的设置不会太浪费。但是,对软盘没有预留任何块。

debugfs 是一个文件系统调试程序。它允许直接访问保存在磁盘上的文件系统数据结构,因此,对 fsck 不能自动修复的磁盘错误,就可以用它来修复。它还可用于恢复被删除 了的文件。但是使用 debugfs 时,要求清楚自己正在干什么。如果贸然行事将导致数据受损。

16.6 文件系统的组织

本节将讨论标准 Linux 目录树,它们是基于 FSSTND (文件系统分层结构标准)的文件系统标准。该标准描述了划分目录树的普通方式并指出这种划分方式的动机。该标准尽力为如何组合 Linux 系统内的目录树建立一个标准。这样的标准有它的优点:即便于为 Linux 系统编写或移植软件,又便于管理配置 Linux 的机器,因为 Linux 需要的一切东西都将放在约定俗成的某个地方。这个标准不要求每个人都必须遵从它,但它正越来越多地得到许多 Linux 版本的认可。

FSSTND 尽力遵循 Unix 的习惯和当前的发展趋势,使 Linux 系统更加接近其他的 Unix 系统。完整的目录树将被分为若干部分,各部分在自己的磁盘或分区内,更适应于磁盘空间,更易于进行备份和其他系统管理。目录树中主要的几个部分是/root、/usr、/var 和/home。 各部分用途不同。具体描述如下:

- · /root 文件系统 是每台机器特有的,其中包含启动系统和引导装入文件系统所需的文件。因此,对于单用户模式,root 文件系统的内容就够了。另外,该文件系统内还包含一些工具,用于修复损坏的系统和从备份中恢复丢失的文件。/root 文件系统一般比较小,因为严格的文件和一个小的不经常改变的文件系统通常都不容易损坏。如果根文件系统发生损坏,则意味着除非用特定的方法(例如从软盘)引导系统,否则系统将无法引导,所以要格外小心。
- · /usr 文件系统 包含所有的命令、库、手册页和其他没有变动过的文件,这些文件 是普通操作期间所需要的。/usr 文件系统内没有专门针对哪台机器的文件,也没有 普通操作期间应该修改的文件。
- · /var 文件系统 包含有变动的文件 ,如假脱机目录、日志文件、手册页和临时文件。
- /home 文件系统 包含用户的根目录,也就是系统上的所有真实数据。把根目录和
 用户的其他目录树或文件系统区分开的目的是为了方便备份。其他部分通常是不
 需要备份的,或者说至少不需要经常备份。

虽然前面的各部分被称为文件系统,但并不要求它们真的在一个独立的文件系统上。 如果系统是一个小型的单用户系统,而且如果想一切简单明了,以上各部分统统可合成一 体。

Unix 文件系统结构根据文件的用途把它们分为各个组,也就是所有的命令归入一个地

方,所有的数据文件归入另一个地方,文档又另行存放等等。另外,就是根据文件所属的 程序来分组。后一种分组方法的不足就是很难实现文件的共享,而且有时还不方便文件的 查找。

16.6.1 根文件系统

通常情况下,根文件系统所占空间一般比较小,因为其中的绝大部分文件都不需要经 常改动,而且包括严格的文件和一个小的不经常改变的文件系统不容易损坏。

除了一个叫/vmlinuz 标准的系统引导映像之外,根目录一般不含任何文件。所有其他 文件在根文件系统的子目录中。

· /bin 目录

/bin 目录包含了引导启动所需的命令或普通用户可能用的命令。这些命令都是二进制 文件的可执行程序,它们大多是系统中重要的系统文件。

· /sbin 目录

/sbin 目录类似/bin 目录,也用于存储二进制文件。因为其中的大部分文件都是系统管 理员使用的基本的系统程序,所以虽然普通用户必要且允许时可以使用,但一般不给普通 用户使用。

· /etc 目录

/etc 目录存放着各种系统配置文件,其中包括用户信息文件/etc/passwd 和系统初始化文件/etc/rc 等。Linux 正是靠这些文件才得以正常地运行。

・ /root 目录

/root 目录是超级用户的目录。

・ /lib 目录

/lib 目录是根文件系统上的程序所需的共享库,存放了根文件系统程序运行所需的共享 文件。

・ /dev 目录

/dev 目录存放了设备文件,即设备驱动程序。通过这些文件访问外部设备。

• /tmp 目录

/tmp 目录存放程序在运行时产生的信息和数据。

・ /boot 目录

/boot 目录存放引导加载器使用的文件,如 LILO,核心映像也经常放在这里,而不是 放在根目录中。

・ /mnt 目录

/mnt 目录是系统管理员安装文件系统的安装点。程序并不自动安装到/mnt。/mnt 下面 可以分为许多子目录。

· /proc、/usr、/var 和/home 目录

其他文件系统的安装点。

16.6.2 /etc文件系统

/etc 目录包含各种系统配置文件,下面介绍其中的一些。

• /etc/rc、/etc/rc.d 和/etc/rc?.d

启动、或改变运行级时运行的脚本或脚本的目录。

/etc/passwd

用户数据库,其中的域给出了用户名、真实姓名、用户起始目录、加密口令和用户的 其他信息。

/etc/fdprm

软盘参数表,用以说明不同的软盘格式。可用 setfdprm 进行设置。

/etc/fstab

指定启动时需要自动安装的文件系统列表。包括用 swapon -a 启用的 swap 区的信息。

/etc/group

类似/etc/passwd,但说明的不是用户信息而是组的信息。包括组的各种数据。

/etc/inittab

init 的配置文件。

/etc/issue

包括用户在登录提示符前的输出信息。通常包括系统的一段短说明或欢迎信息。具体 内容由系统管理员确定。

/etc/motd

motd 是 Message Of The Day 的缩写,成功登录后自动输出,内容由系统管理员确定。 常用于通告信息,如计划关机时间的警告等。

/etc/mtab

当前安装的文件系统列表。由脚本初始化,并由 mount 命令自动更新。当需要一个当前安装的文件系统的列表时使用。

· /etc/shadow

在安装了 shadow 口令软件的系统上的影子口令文件。shadow 口令文件将/etc/passwd 文件中的加密口令移动到/etc/shadow 中,而后者只对超级用户可读。这使破译口令更困难,以此增加系统的安全性。

/etc/login.defs

login 命令的配置文件。

/etc/printcap

打印机配置文件。

/etc/profile、/etc/csh.login、/etc/csh.cshrc

登录或启动时 Bourne 或 Cshells 执行的文件。这允许系统管理员为所有用户建立全局 缺省环境。

/etc/securetty

确认安全终端,即哪个终端允许超级用户登录。

• etc/termcap

终端性能数据库。说明不同的终端用什么转义序列控制。写程序时不直接输出转义序 列,而是从/etc/termcap 中查找要做的工作的正确序列。 16.6.3 /dev文件系统

/dev 目录包括所有设备的设备文件。设备文件用特定的约定命名,这在设备列表中说明。设备文件在安装时由系统产生。下面简要介绍/dev下的一些常用文件。

/dev/console

系统控制台,也就是直接和系统连接的监视器。

/dev/hd

IDE 硬盘驱动程序接口。

- /dev/sd
- SCSI 磁盘驱动程序接口。
- /dev/fd

软驱设备驱动程序。

SCSI 磁带驱动器驱动程序。

/dev/tty

提供虚拟控制台支持。

/dev/pty

提供远程登录伪终端支持。在进行 Telnet 登录时就要用到/dev/pty 设备。

/dev/ttys

计算机串行接口。

/dev/cua

计算机串行接口与调制解调器一起使用的设备。

/dev/null

所有写入该设备的信息都将消失。

16.6.4 /usr文件系统

/usr 是个很重要的目录,通常这一文件系统很大,因为所有程序都安装在这里。/usr 里的所有文件一般来自 Linux 发行版。本地安装的程序和其他一些在/usr/local 下,因为这 样可以在升级新版系统或新发行版时无须重新安装全部程序。/usr 目录下的许多内容是可 选的,但这些功能会使用户使用系统更加有效。/usr 可容纳许多大型的软件包和它们的配 置文件。下面列出一些重要的目录。

• /usr/X11R6

包含 X Window 系统的所有可执行程序、配置文件和支持文件。为简化 X 的开发和安装, X 的文件没有集成到系统中。

• /usr/bin

集中了几乎所有用户命令,是系统的软件库。另有些命令在/bin 或/usr/local/bin 中。

• /usr/sbin

包括根文件系统不必要的系统管理命令,例如多数服务程序。

/usr/share/man、/usr/share/info、/usr/share/doc

 $^{5 \}cdot /dev/st$

这些目录包含所有手册页、GNU 信息文档和各种其他文档文件。每个联机手册都有两 个子目录。例如:/usr/man/man1 中包含联机手册的源码,/usr/man/cat1 包含已格式化的内 容。

/usr/include

包含 C 语言的头文件。这些文件多以.h 结尾 ,用来描述 C 语言程序中用到的数据结构、 子过程和常量。

/usr/lib

包含了程序或子系统的不变的数据文件。编程的原始库也存在/usr/lib 里。

/usr/local

本地安装的软件和其他文件放在这里。这与/usr 很相似。

16.6.5 /var文件系统

/var 包含系统一般运行时要改变的数据。通常这些数据所在的目录的大小是要经常变 化或扩充的。原来/var 目录中有些内容是在/usr 中的,但为了保持/usr 目录的相对稳定,就 把那些需要经常改变的目录放到/var 中了。下面列出一些重要的目录。

/var/lib

存放系统正常运行时要改变的文件。

/var/local

存放/usr/local 中安装的程序的可变数据。

/var/lock

锁定文件。许多程序遵循在/var/lock 中产生一个锁定文件的约定,以支持它们正在使 用的某个特定的设备或文件。其他程序注意到这个锁定文件时,就不会再使用这个设备或 文件。

/var/log

各种程序的日志文件,尤其是 login (/var/log/wtmplog 记录所有到系统的登录和注销) 和 syslog (/var/log/messages 记录存储所有核心和系统程序信息)。

/var/run

保存在下一次系统引导前有效的关于系统的信息文件。例如 , /var/run/utmp 包含当前 登录的用户的信息。

/var/spool

放置假脱机(spool)程序的目录。如 mail、news、打印队列和其他队列工作的目录。 每个不同的 spool 在/var/spool 下有自己的子目录。

• /var/tmp

临时文件。

16.6.6 /proc文件系统

/proc 文件系统是一个伪文件系统,它是一个实际上并不存在的目录。它并不存在于某 个磁盘上,而是由核心在内存中产生。这个目录用于提供关于系统的信息。下面说明一些 最重要的文件和目录。 · /proc/进程号

关于进程的信息目录。每个进程在/proc 下有一个名为自己进程号的目录。

/proc/cpuinfo

存放 CPU 的信息。

/proc/devices

当前运行的核心配置的设备驱动的列表。

/proc/dma

显示当前使用的 DMA 通道。

/proc/filesystems

核心配置的文件系统信息。

/proc/interrupts

显示被占用的中断信息和占用者的信息,以及被占用的数量。

/proc/ioports

当前使用的 I/O 端口。

/proc/kcore

系统物理内存映像。

/proc/kmsg

核心输出的消息。

/proc/ksyms

核心符号表。

/proc/meminfo

各种存储器的使用信息,包括物理内存和交换分区。

/proc/modules

当前加载的核心模块信息。

/proc/net

网络协议状态信息。

/proc/stat

系统的状态。例如系统启动后页面发生错误的次数。

/proc/uptime

系统启动的时间长度。

- /proc/version
- 核心版本。

16.7 文件权限

16.7.1 文件权限的概念

由于 Linux 是一个多用户的操作系统,所以就要考虑安全性的问题。文件权限是操作

系统安全性的一个重要因素。

与 Unix 一样, Linux 系统也是多用户系统。这样就有一个用户的文件会不会暴露在另 一个用户面前的问题。为了保护用户的私人文件不被其他用户所侵犯,就出现了文件权限 的概念。这种文件权限的概念允许文件和目录归一个特定的用户所拥有。用户权限除了对 他所拥有的文件或目录进行存取或其他操作外,也包括限制其他用户在这些文件上的存取 和操作。

例如,用户A在自己的主目录中建立了某些文件,那么用户A就成为这些文件的所有者,就有权对这些文件进行存取,也有权限制其他用户对这些文件的存取访问。Linux 允许用户之间以及用户组之间共享文件。如果用户A愿意,他可以切断一切到他所拥有的文件的通路,即使得任何人都不能存取这些文件。当然在大多数情况下是只给其他用户分配只读权限,而不给他们对文件的修改和删除的权限。

为了方便管理,每个文件都被一个特定用户所拥有的同时还被系统指定的用户组所拥 有。一个用户至少在一个用户组中,有时系统管理员还可以将一个用户编进多个用户组中。

用户组通常是根据使用计算机的用户的种类来划分的。例如,在学校中根据用户的身份可以分为 student、staff 和 guest 等组。系统使用这些组来控制对资源的访问。

16.7.2 文件权限的构成

文件的权限有 3 种:读、写和执行。读权限允许用户阅读文件的内容,对目录则允许 用户列出目录中的内容;写权限允许用户写和修改文件,对目录写权限允许用户在这个目 录中建立新文件或删除文件;执行权限允许用户运行文件,对目录执行权限允许用户进入 和退出该目录。下面举例说明上面所说的文件权限。使用 ls-1 命令来显示文件列表。

```
#cd /bin
```

```
#ls -l
```

```
total 5828
```

-rwxr-xr-x	1 root	root	2828 Aug 31	2000 arch
-rwxr-xr-x	1 root	root	64924 Aug 31	2000 ash
-rwxr-xr-x	1 root	root	384232 Aug 31	2000 ash.static
-rwxr-xr-x	1 root	root	10460 Jul 12	2000 aumix-minimal
lrwxrwxrwx	1 root	root	4 Apr 23	14:18 awk -> gawk
-rwxr-xr-x	1 root	root	5780 Jul 13	2000 basename
-rwxr-xr-x	1 root	root	512540 Aug 22	2000 bash
lrwxrwxrwx	1 root	root	4 Apr 23	14:18 bash2 -> bash
lrwxrwxrwx	1 root	root	3 Apr 23	14:18 bsh -> ash
-rwxr-xr-x	1 root	root	9588 Aug 31	2000 cat
-rwxr-xr-x	1 root	root	14972 Aug 25	2000 chgrp
-rwxr-xr-x	1 root	root	16732 Aug 25	2000 chmod
-rwxr-xr-x	1 root	root	15676 Aug 25	2000 chown

•••••

从上面所列出的文件的长格式信息中可以看出,文件信息共有9列。在第1列中由长 度为10的字符组成的一串字符的就是文件权限的表示。每个文件的权限都是由一个十位的 二进制数组成。

这个十位二进制数分为4部分:第1部分是第1位代表文件的类型,文件类型的种类 和标志符号如表16.4 所示;第2部分是2~4位,代表文件所有者对文件的权限;第3部分 是5~7位,代表该文件所有者所在组对该文件所拥有权限;第4部分是8~10位,代表其他 用户对文件拥有权限。第2列代表文件的链接数;第3列是文件的所有者;第4列是文件 所属的用户组;最后一列是文件名。

	表 16.4	文件类型和标志符号
--	--------	-----------

文件类型	标志符号
普通文件	-
块设备文件	b
字符设备文件	c
连接文件	1
目录文件	d

表示文件权限的 3 个字符,依次代表读、写和执行权限。系统用 r 代表读权限,用 w 代表写权限,以及用 x 代表执行权限。当没有相应的权限时,该权限对应的位置用短线(-) 表示。如上所示的第 10 行,文件名为 cat,第 1 个短线代表该文件是普通文件,后面的 3 个字符 rwx 表示 root 用户对该文件有读、写权限和执行权限。接下去的 3 个字符 r-x 表示 与用户 root 同属于 root 组的用户对该文件有读权限和执行权限,但没有写权限。最后的 3 个字符 r-x 表示组外的其他用户也有该文件的读权限和执行权限,但不能对该文件进行写操 作。

16.7.3 文件权限的依赖性

要特别注意的是,用户对文件所拥有的权限依赖于他对文件所属的目录所拥有的权限。 虽然一个文件的权限为-rwxrwxrwx,但其他用户也只有在对该文件所属目录拥有读、写和 执行权限时,才能够对该文件进行读、写和执行。

这就是说,如果用户A希望限制他人对其拥有文件的访问,那么用户A只需将他的目录/home/A的权限设置成-rwx-----。这样,其他任何用户都不能访问他的目录,包括其中的所有文件和子目录。这样用户A就不必一个一个地修改他的文件的权限了。如果用户要访问某个文件,这个用户必须具有对该文件路径名中所有目录的执行权限以及对该文件自身的相应的访问权限。

通常,用户都是非常开放的,文件的权限一般都设置成-rw-r--r--,这种权限的设置允 许其他用户读文件,但不能写和执行。而目录的权限一般都设置成-rwxr-xr-x,这种权限的 设置允许其他用户查看目录,但不能在其中建立或删除文件。

16.7.4 改变权限

在系统应用中有时需要让其他用户使用某个他本来不能使用的文件的时候,就需要改 变文件的权限。在 Linux 中,使用 chmod 命令进行有关权限的设置。但只有当文件的拥有 者对某个文件有写的权限时,他才能够改变该文件的权限(超级用户可以对所有文件进行 权限设置)。

chmod 的命令格式主要有两种,下面是直接设置文件权限的第一种格式:

chmod [-R] mode filename1 filename2...

其中-R 是指递归地改变所指定的目录参数及其子目录和它们所组成的文件的访问权 限。mode 是一个八进制的数,直接描述一个文件的权限。文件权限是一个十位的二进制数。 Linux 中规定:十位数中每个位上如果出现了字母,则该位记为1,如果是-,则记为0。这 样将十数位都设置完后将得到一个十位的二进制数。由于 UNIX 系统最开始是采用八进制 数系统,所以至今 Linux 上仍然需要把这十位二进制数转化为八进制数,这样就得到一个 四位的八进制数,也就是 mode 参数。实际上对于普通文件第1位始终为0,输入 mode 参数的时候写3位就可以了。

例如:将某文件权限设置为对文件所有者可读、可写和可执行,同组用户可读,其他 用户可读。首先写出文件权限字段:-rwxr--r--,然后转化为二进制:0111100100,接着转 化为八进制:0744,最后用 chmod 命令进行设置即可。chmod 的第二种命令格式相对比较 直观,更容易掌握。命令格式如下:

chmod {a, u, g, o} {+-} {rwx} filename1 filename2...

该命令首先指明权限所要赋予的用户类,可以是 a、u、g 和 o 中的一个或多个,其中 a 代表所有的用户, u 代表文件的所有者自身,g 代表同组的用户, o 代表组外的其他用户。 接着指明是增加权限还是删除权限,用+选项表示增加权限,用-选项表示删除权限。最后 指明所要增加或删除的权限,这些权限可以是 r、w 和 x 中的一个或多个。

例如,假设在根目录下有一个名为 nyp.gif 的文件,要更改该文件权限,先用如下命令: #ls-l

查看其权限为-rw-r--r--,将其转化为二进制表示 0110100100,相应八进制表示为 644, 然后依次用第一种命令方式改变其权限为-rw-r-xr-x,其八进制表示为 655,所以用 chmod 655 nyp.gif 即可,第二种方式改变其权限为-rw-rwxr-x 和-rw-rwxrwx。

16.8 习题

1.使用 ls -l 命令查看/dev 下的文件,注意文件属性的第一位,并区分字符设备和块设备。

2. 阅读第 16.5 节中介绍的命令,在自己的 Linux 系统上试用。

3. 阅读第 16.6 节中对文件系统组织的分类,加深对 Linux 文件组织的理解。

第17章 常用用户管理

本章主要介绍常用的用户管理手段。内容包括用户管理简介、超级用户管理、帐号配 置文件、普通用户管理、使用 Linuxconf 管理用户以用户查询命令简介等。

17.1 用户管理简介

用户管理的基本概念涉及用户账号和账号口令,下面将介绍这两个概念。

17.1.1 账号

当一台计算机供多个用户使用时,有必要区分各个用户,以便他们的私人文件只供各 用户私人使用。即使计算机一次只供一名用户使用,这样做也同样重要。因此,每个用户 都有一个独一无二的用户名,这个用户名供他们登录时使用。但是对用户来说,更重要的 是账号。所谓账号,就是属于某个用户的所有文件、资源和信息。

17.1.2 账号口令

账号口令是计算机系统防止从其中盗取信息和资源的一道重要的屏障。如果 Linux 计 算机可以通过因特网或者调制解调器被访问,就必须认真地设置好口令。对于系统管理员 的超级用户口令和用户的因特网服务,更是如此。

Linux 系统虽然不能够控制用户选择什么样的口令,但可以帮助用户选择更好的口令。 Linux 系统可以使用工具软件查找出哪些口令选择得不好,然后分别通知各个用户,让用户 修改口令,并指导用户更安全地进行选择。

是否需要对口令进行检查取决于系统的易受攻击性、系统的知名度以及希望达到的安 全强度。当为某些人分配了新口令,或者当某个用户修改了自己的口令的时候,passwd 程 序可以指出哪些是比较差的口令。使用可以被 passwd 程序轻易检查出来的口令会对系统安 全性造成潜在的危害。

在命令行和 GUI 状态下都有管理用户的工具软件。具体使用哪一类工具软件只是个人 爱好问题。在 GUI 界面下,linuxconf 程序通常是使用最多的选择。本章将介绍在命令行模 式下的工具软件以及如何在 linuxconf 程序中完成这类任务。

17.1.3 shadow口令

保存在/etc/passwd 文件中的口令是经过加密编码的,因此不像普通的文本那样容易被 读出来。这就意味着不能通过查看这个文件来找出别人的口令,这也是为加强系统安全性 迈出的一步。但是这种加密编码的方法并不是很保险。一旦它们从这个文件中被获取,解 开口令的加密编码并不是很困难的事。 /etc/passwd 文件对任何人都是可读的。一个侵入者需要做的只是在设法闯入任何一个 用户账号之后通过使用黑客程序处理某个字典单词集破译口令,通过掌握某个用户设置口 令的习惯就可以简单地拷贝一份/etc/passwd 文件并把其中的加密编码解开。这个漏洞看起 来好像只需要通过修改该文件的存取权限的方法就可以弥补,但是这个方法行不通。因为 在/etc/passwd 文件中包含着重要的数据,如用户信息、分组信息,还有 shell 信息等等,有 很多程序需要对这个这些信息进行存取,因此不能把它设置为只对超级用户可读的权限。

解决这个问题最常用的一个办法是:安装 shadow 口令软件。这个软件将把全部口令都 转移到只对超级用户可读的某个文件中,这样就避免了某些人简单地通过盗取/etc/passwd 文件而破译口令。

17.2 超级用户管理

root 也被称为超级用户,它是每个 Linux 系统上都具有的特殊用户账号,这一特殊用 户对系统具有完全的访问权限。当利用 root 时,系统忽略所有的权限,它把系统上每个文 件和设备的读、写和执行的权限都提供给 root 用户。这意味着,如果以 root 身份运行 rm -rf /命令,将删除整个系统,另外还意味着 root 能够访问所有的数据。这对于备份和恢复数据、 系统的维护以及系统的安全性都是很重要的。

root 账号是全能的,它既可以用来维护系统,使系统在稳定的环境中运行,也可以破 坏系统以及系统中的数据。总之,root 账号是一个威力极大的工具。每个人都有可能犯错 误,作为一个普通用户可能只是引起很小的一场混乱,而作为超级用户,犯一个大错误就 有可能毁坏系统。

下面是应该知道的关于超级用户和 Internet 的一些常识。

许多 Internet 服务在处理超级用户的情况时都有一些特殊的考虑。如在缺省的情况下用 户无法作为超级用户使用 telnet 功能进入一台 Linux 计算机。如果允许这样做,就会被认为 是一个相当大的安全漏洞,因为别人只要能够猜出超级用户口令就可以对整个系统进行存 取。如果要求至少需要先进入某个普通用户账号,情况就会好得多,因为这样会使攻击进 入系统的过程时间长一些,用户也就更容易发现它。

以超级用户身份进入 IRC (网上聊天) 也有可能会遭到拒绝或禁止。超级用户权限允 许通过某台计算机运行一系列攻击性的程序功能 , 因此许多 IRC 不接受超级用户。

以超级用户身份发送电子邮件或者在新闻组中发布消息一般也不受欢迎,除非正在讨 论的是一个系统管理员关心的问题。

正因为 root 具有这样的控制系统的能力,因此作为一个系统管理员,在使用 root 账号 时应该特别小心。系统管理员不仅需要在操作系统时小心,而且在处理口令时也应该十分 小心。口令是计算机确定试图使用某个账号登录的用户是否合法的唯一证明。因此,不论 是谁,只要他获得 root 口令,就能够控制系统。

由于 root 账号的特殊功能,系统上具有 root 口令的用户不应该直接以 root 身份进入系统。当某个任务需要有 root 权限才能执行时,可以使用 su 命令转为 root,然后执行这一任务,并在任务完成后重新返回到自己的正常账号中。这种做法可以确保以下两件事情:

- 可以避免用户意外地执行他们不打算执行的操作,而这些操作是可以被 root 执行 的。
- 提供日志。/etc/login.defs 文件是一个用来定义 su 记录;注册失败的记录和注册到
 系统的用户的记录以及其他一些内容的文件。虽然做这样的日志并不能阻止这一
 操作,但至少有助于确定是谁执行了这一操作。

17.3 账号配置文件

Linux 中的和用户管理有关的文件包括/etc/passwd、/etc/group 和/etc/shadow 文件,下面着重介绍/etc/passwd 和/etc/group 文件。

17.3.1 /etc/passwd文件

Linux 系统中,基本的用户数据库是一个文本文件,名为/etc/passwd,它列出了所有有 效用户名及其相关信息。在该文件内,每个用户名都占一行,它被分为7个字段,中间用 冒号定界,如下所示:

[用户名]:[密码]:[用户 ID]:[组 ID]:[注释]:[主目录]:[登录 shell]

系统上的任何一名用户都可能读取这个密码文件,所以一个用户能够从中得知另一个 用户的用户名。这意味着人人都能看见密码,密码文件对密码进行了加密。从理论上讲, 以加密形式保存的密码是可靠的,但是,加密也是很容易被解密的,特别是密码比较薄弱 的时候。因此,现在的 Linux 系统都有影子密码。这是另一种保存密码的方案:加密的密 码被保存在一个独立的文件内,该文件名为/etc/shadow,只有 root 才能读取这个文件。 /etc/passed 文件只在第二个字段内包含一个特殊的标记。任何需要验证用户是 setuid 的程序 都可因此而访问影子密码文件。只能使用密码文件内其他字段的普通程序是不能看到这个 影子密码文件的。

下面对这些/etc/passwd 文件中各字段内容进行更详细的解释。

1. 用户名

用户名是一个字符串,通常由不多于 8 个的字符组成。用户名唯一标识用户,并且应 易于为用户所识别和记住。系统通过用户名来标识用户。用户名一般是由名和姓的字母组 合而成。

虽然有关如何指定用户名存在着某些惯例,但是只要用户名是唯一的,计算机并不关 心是怎样的用户名。实际上,在用户名中可以有下划线、句号、数字和某些特殊字符,大 小写字母是有区别的。

2. 口令

系统把用户的加密口令存放在本字段中。当系统利用 shadow 口令系统时,在本字段置入的值将为*。由于*并不是口令段的有效字符,所以值*阻止登录访问账号。这个字段不能进行手工编辑,应使用 passwd 之类的程序,以使正确的加密可以进行。手工改变这个字段后,原来的口令不再有效,因而必须利用 root 来改变。

当系统利用隐蔽口令系统时,存在一个含有口令的单独文件,它叫作/etc/shadow。只 有 root 用户可以读这个文件。

3. 用户 ID

每个用户名都有一个与其相关联的编号,这个编号也叫做 UID,系统利用它来标识用 户所拥有的任何信息。与用户有关的所有进程和文件等都以这种方式标识。用户 ID 的有效 范围为大于或等于 0。

4.组ID

每个用户都属于一个用户组,每个用户组都有一个编号叫做组 ID。

5. 注释

本字段是由其他程序标识用户的。通常,用户的实际名存放在本字段中。有关本字段 需要注意的一点是:任何人都能读出它。

除用户对这一字段具有访问权限外,某些实用程序也使用这一字段作为一个标识符。

6. 主目录

主目录字段通知系统把用户转储到哪里。如果登录成功,此目录即是用户的主目录, 当然也不一定必须如此。用户只要能够进入系统,系统并不关心目录指向哪里。所有用户 主目录存放的标准目录为/home 之下。

7. 登录 shell

登录 shell 是用户第一次登录时所执行的命令。在大多数情况下,这是一个 shell 命令。 而在另一些情况下,它可能是前端的界面或简单的应用程序。如果本字段保持空,则系统 设置为/bin/bashshell。

下面是一个/etc/passwd的示例:

root::0:0:root:/root:/bin/bash

bin:*:1:1:bin:/bin:

daemon:*:2:2:daemon:/sbin:

adm:*:3:4:adm:/var/adm:

lp:*:4:7:lp:/var/spool/lpd:

sync:*:5:0:sync:/sbin:/bin/sync

shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown

halt:*:7:0:halt:/sbin:/sbin/halt

mail:*:8:12:mail:/var/spool/mail:

news:*:9:13:news:/usr/lib/news:

uucp:*:10:14:uucp:/var/spool/uucppublic:

operator:*:11:0:operator:/root:/bin/bash

games:*:12:100:games:/usr/games:

man:*:13:15:man:/usr/man:

postmaster:*:14:12:postmaster:/var/spool/mail:/bin/bas

nobody:*:-1:100:nobody:/dev/null:

ftp:*:14:50::/home/ftp:/bin/bash

17.3.2 /etc/group文件

每个用户至少要与一个组关联。组是针对特定用途的一组用户的集合。

在系统内,组的数目是没有限制的。实际上,缺省的/etc/group 文件含有很多个组。每 行含有4个段,和/etc/passwd 文件一样用冒号分隔。假如字段中未输入任何信息,则字段 保持空,字段之间仍用冒号分隔。如下所示:

[组名]:[口令]:[组 ID]:[用户]

组名字段是唯一的组标识符,口令字段通常置空或 * , 用户可以指定口令。组 ID 字段 是标识操作系统内组的唯一编号 , 用户字段是属于该组的全部用户 ID 的列表。

下面是一个 group 文件的示例:

root::0:root bin::1:root,bin,daemon daemon::2:root,bin,daemon sys::3:root,bin,adm adm::4:root,adm,daemon ttv::5: disk::6:root lp::7:daemon,lp mem::8: kmem::9: wheel::10:root mail::12:mail news::13:news uucp::14:uucp man::15: games::20: gopher::30: dip::40: ftp::50: nobody::99: users::100: xfs:x:102: flopy:x:103: gdm:x:42: pppusers:x:230: popusers:x:231: slipusers:x:232: postgres:x:233: slocate:x:21: squid:x:234:

编辑这个文件有两种方法。第一种方法是使用一个脚本,例如 addgroup 或 groupadd;

第二种方法是利用文本编辑器来手工编辑文件。如果要把组添加到此文件中,只须依照其 他项的格式。添加唯一的组时,要给出组的唯一组 ID。如有必要,对组指定一个口令,然 后列出与该组关联的用户,用逗号分隔这些用户。如果以不正确的格式或其他不正确的方 式输入上面的行时,用户就可能无法使用这个组 ID。

当系统利用隐蔽口令系统时,口令段被移到/etc/shadow.group 中,并给此字段赋以 x。

在完成/etc/group 文件的编辑后,需要核实其权限,它应该由 root 所拥有,而且它的组 应该是 root 或 sys,组 ID 为 0。这些权限对拥有者为读和写,而对其他人为读。

组的列表不必按任何特定的顺序。每个组中的用户列表顺序也是无关的。Red Hat Linux 将搜索整个文件,直到查找到要找的行为止。

虽然用户可以属于几个组,但是 Linux 只允许他们在任何给定的时间内在单个组中是 被激活的。起始组是在/etc/passwd 文件中所标识的组,当用户需要切换到另一个组时,必须 用 newgrp 命令进行切换。

删除一个组或从组中删除一个用户,只需编辑/etc/group 文件删除整行或删除特定用户。检查/etc/passwd 文件以确保删除的组中没有定义的用户。

17.4 普通用户管理

日常的用户管理包括添加新用户;临时禁用用户;删除用户和修改用户属性等工作。 下面分别进行介绍。

17.4.1 添加用户

通常有两种办法可以在系统中添加用户:手工创建或使用工具创建。

第一种方法是手工编辑/etc/passwd 文件和/etc/group 文件。第二种方法是使用很多工具中的一种(命令行或图形方式)来增加用户。使用第二方法是比较好。因为,这种方法可以减少出错而且用户不必去理解过程或熟悉编辑器。

许多 Linux 版本都带有用于创建账号的程序,有几个此类的程序是可以采用的。这里, 有两个命令供选择,它们是 adduser 和 useradd,另外还有一个 GUI 工具 Linuxconf。

下面分别介绍手工添加用户和使用工具添加用户,以及可以改变新用户的缺少设置的 一些参数。

1. 手工添加用户

按照下面的步骤手工创建一个用户:

(1)利用 vipw 编辑/etc/passwd,为新账号增加一个新行,令密码字段为*,以便用户 不可能进行登录。

(2) 如果需要创建一个新组,采用类似的做法用 vigr 编辑/etc/group。

(3)利用 mkdir 创建用户根目录。

(4) 把文件从/etc/skel 复制到新建的根目录。

(5)利用 chown 和 chmod 确定拥有权和访问许可,-R 选项是最有用的。通常使用下 面的命令: #cd /home/newusername
#chown -R username.group .
#chmod -R go=u,go-w .
#chmod go= .

(6)利用 passwd 设置密码。设置密码之后,账号就开始发挥作用了。

2. 使用命令行添加用户

有两种不同的命令可以用来添加用户:adduser 和 useradd 命令。这两个命令工作的方 式完全相同。使用者必须以根用户登录进入系统才能建立新用户。

请输入 adduser username 开始添加一个新用户。例如,如果想添加新用户账户 mary, 输入 adduser mary 命令即可。然后为这个新用户设置一个口令,就可以使用这个账户了。

3. 改变新用户的缺省设置

在建立新用户的时候,如果需要改变关于新用户的缺省设置,可以使用 adduser 和 useradd 命令的对应参数。如表 17.1 所示是用来修改缺省用户设置值的各个参数。

用户设置值	参数	数据格式
GROUP	- g	来自/etc/group 文件的分组编码
HOME	- d	该账户上一级用户目录的完整路径
INACTIVE	- f	天数
EXPIRE	- e	天数
SHELL	- S	到 shell 的完整路径
SKEL	- k	到框架文件的完整路径

表 17.1 使用 adduser 和 useradd 命令修改新用户缺省设置值参数

下面介绍其中的一些选项。

当用户不修改过期失效口令时,需要把这类用户的账户冻结起来。这就需要修改 INACTIVE 项目的设置值。办法是:输入一个以按天数计算的数字。这个数字表示在用户 口令失效之后和彻底关闭该账户之前系统将要等待的时间。如果使用缺省的数值0天,就 表示不希望自动冻结那些账户。

如果系统比较容易受到攻击,定期使原来的口令失效可以提供一定的保护。请把 EXPIRE 项目修改为一个数字,这样就可以通知系统每循环经过多少天,原来的口令就将 失效;而在此之前,系统必须强迫用户修改口令。

缺省的 SHELL 项目的设置值一般都是/bin/bash。输入新的缺省 shell 的完整路径名可以改变该设置。

通过使用 skeleton 目录可以定制新用户的目录。缺省的 skeleton 目录通常是/etc/skel 目录,这个框架目录中包括准备安排到每一个新建账户的目录中的文件。如果想把框架目录 设置为缺省值以外的某个位置,就需要把 SKEL 项目的设置值修改为新的框架目录的完整 路径名。

例如,设置口令失效期为90天,如果失效口令在30天内没有修改,就冻结该账户,

把缺省的 shell 改为/bin/ksh。

使用 adduser -e 90 -f 30 -s /bin/ksh 命令即可达到该要求。

17.4.2 修改用户属性

下面几条命令用于更改账号属性,也就是/etc/passwd内的相关字段。

chfn 用于更改注释字段; chsh 用于更改用户 shell; passwd 用于更改用户密码。

例如, root 账户拥有为其他用户添加或者修改口令的权限,使用 passwd username 命令 来完成口令的添加和修改工作。即使用户当前登录进入的是一个非 root 账户,也可以通过 passwd 命令改变这个账户的口令。

超级用户以这些命令来更改任何一个账号的属性。普通用户只能更改自己账号的属性。 其他的用户属性可以手工修改。例如,要更改用户名,可以用 vipw 直接编辑/etc/passwd。 同样,要增加或删除用户可以用 vigr 编辑/etc/group。

下面介绍 usermod 命令的使用方法。使用 usermod 命令可以修改现有用户的信息,该 命令加上必要的参数可以指定需要修改的项目。如表 17.2 所示是可以使用的参数。

参数	名称	说明
с	说明信息	用在 finger 文件中的说明信息。这个数据域最好是用 chfn 命令来修改
d	用户目录	为这个用户输入一个新的用户目录位置,新位置将自动被建立。如果在 d 参数
		后加上一个 m 参数,那么原来的用户目录中的项将会被转移到新的目录中
e	失效日期	如果想把用户账户的失效日期设置为某个特定的日子 ,请按照 MM/DD/YY 的格
		式输入该日期
f	冻结期	口令失效之后至冻结该账户之前需要等待的天数。只有在该用户没有修改其口
		令的时候这个选项才被激活。如果不想使用这个选项,请输入数值-1(缺省值)
g	原始分组	分配给用户的分组名称或者编号。这个分组必须是当前存在的
G	分组	使用这个参数可以给该用户分配额外的分组。如果需要分配不止一个分组,请
		使用逗号而不是空格来分隔它们。例如,如果想把 wheel、root 和 admin 等几个
		分组分配给该用户,需要输入 whell, root, admin
1	登录名	使用这个参数可以修改用户的登录名。这么做的时候用户目录不会自动改变。
		如果用户已经登录进入了系统,就不能修改用户的登录名
S	shell	如果用户要求使用另外一个登录 shell,使用这个参数修改它。需要使用新 shell
		的名称
u	UID	改变用户 ID 的数值。除非了解这样做的后果和系统上当前可用的 ID 数值范围 ,
		否则不要这样做

表 17.2 usermod 命令中用来修改用户属性的参数

例如,如果用户 chris 要求把他的登录名修改为 c.adams,就需要以 root 身登录进入系统,然后输入 usermod -d /home/c.adams -m -l c.adams.chris 命令即可。

17.4.3 禁用用户

有时,系统管理员需要在不删除账号的情况下,临时禁用某个用户。注意不要禁用 root

账户。下面介绍两种禁用用户的方法。

修改用户密码。

按照下面的步骤可以在不删除的前提下冻结一个用户的账户:

- (1)以 root 身份登录进入系统。
- (2)编辑 password 文件:
 如果没有使用 shadow 隐藏口令功能,编辑/etc/passwd 文件。
 如果使用了 shadow 隐藏口令功能,编辑/etc/shadow 文件。
- (3) 在文件中找到这个账户。
- (4)把这个账户的口令替换为一个星号(*)。
- (5)保存并退出文件。
- · 改变用户的 shell。

这种方法可以给出提示,让用户知道发生了什么事。

将用户的 shell 改为一个特殊的程序,这个程序只打印一条消息。通过这种方式,任何 想采用这个账号登录的人都会失败,而且还可以知道失败的原因。这条消息能够要求用户 和系统管理员取得联系,以便解决存在的问题。

要创建这样的特殊程序,较简单的方法是编写一个 shell 脚本,如下所示:

#!/usr/bin/tail +2

This account has been closed.

Please call 62565533 for details.

前两个字符#!告诉内核该行中的其他部分是一条命令,需要运行该命令,以便翻译这 个文件。这里的 tail 命令将在标准输出内输出除了第一行以外的所有消息。

tail 脚本应该保存在一个单独的目录下,以便其命令不和普通用户命令发生冲突。并且 tail 脚本应该赋予被禁用的用户以可执行的权限。

17.4.4 删除用户

可以手工或使用命令删除一个用户。

如果手工删除一个用户,必须先删除他的所有文件、邮件别名、打印作业、cron 和 at 作业,以及其他对该用户的所有引用。然后,才能从/etc/passwd 和/etc/group 内删除相关的 行。要从已经添加该用户名的所有组中,删除这个用户名。较好的做法是在开始删除用户 名之前,先取消其账号,以防止该用户在删除期间仍然使用这个账号。

用户在其根目录外面可能还有文件,利用 find 命令可以找出这些文件:

find / -user username

还可以使用命令 userdel 或 deluser 删除用户。要把一个用户连同用户目录一起删除, 使用 userdel -r username 命令即可。

例如,如果想完全删除属于 paula 的账户,应该输入 userdel -r paula 命令。

如果因为某些原因删除一个账户但是又需要保留该用户的用户目录,可以使用不带-r 参数的 userdel 命令。 17.4.5 改变用户组和所有权

chgrp 命令能够改变用户组所有权。chgrp 命令接受一个用户组组名或者用户组 ID 编号 作为输入,如下所示:

#ls -l myfile

-rw-r--r-- 1 mxn mxn 0 Dec 26 15 : 26 myfile

这显示了 myfile 文件属于 mxn 用户和 mxn 用户组。如果想改变用户组的所有权使得 用户组中的其他成员都能够对它进行存取,请输入以下命令:

#groups mxn
mxn : mxn
#chgrp users myfile
#ls -l myfile
-rw-rw-r-- 1 mxn users 0 Nov 23 14 : 16 myfile

现在, users 用户组的其他成员也能够对这个文件进行存取了。在子目录/usr/bin 中还可 以找到功能与 chgrp 命令相类似的 newgrp 命令。使用 chgrp 命令可以把某个文件或者子目 录的用户组所有权转换到归属的另外一个用户组。如果是超级用户,可以把它转换到任何 一个用户组。

17.5 X Window系统中的用户管理

如果使用 X Window 系统, Red Hat Linux 就提供用户管理工具。有两个标准的基于 XWindow 的工具。第一个是 linuxconf 程序,第二个是 userconf 程序。这两个工具的使用 基本相同。如图 17.1 和图 17.2 所示分别是这两个工具的使用界面。



图 17.1 linuxconf 工具使用界面

User account configurator
This package allows you to add/delete and manage user accounts
Normal Special accounts Policies
User accounts
Group definitions
Change root password
Quit Help

图 17.2 userconf 工具使用界面

可以看出,这两个工具作用相同。下面以 Linuxconf 为例介绍用户管理的方法。

17.5.1 添加用户

请按照下面的步骤添加一个新用户:

- (1) 打开 linux conf 程序。
- (2)沿着目录树前进到 Users accounts 之下的 Normal 子目录树。
- (3) 单击 Users accounts 选项,出现如图 17.3 所示的 Users accounts 窗口。



图 17.3 Users accounta 窗口

(4) 单击 Add 按钮, 出现如图 17.4 所示的 User account creation 的 Base info 选项卡。

Apollo: Linuxconf 1.19 (subrev 2)		
Config Control Status	Users accounts User account creation	
	and the full name	
Change root password Special accounts Policies	Base into Params Privileges	The account is enabled
H File systems Miscellaneous services boot mode	Login name Full name	
	group (opt)	
	Supplementary groups	
	Home directory(opt)	
	Command interpreter(opt)	/bin/bash
QuitAct/ChangesHelp	Accept Cancel	J

图 17.4 linuxconf 程序的 User account creation 窗口

(5)在各文本输入框内填写有关的数据。Login name(账户的名称)是必须填写的数据项。如有必要,请单击 Params 或者 Privileges 标签并在相应的选项卡中选择适当的选项。

(6) 单击 Accept 按钮, 出现 Changing password 窗口。

(7) 在文本框输入用户的口令字。

(8)单击 Accept 按钮。如果口令字检查程序拒绝接受刚才设置的口令字,将出现一 个对话框,提示刚才选择的口令字不好,单击 OK 按钮。

(9) 根据提示再次输入口令字。如果还是使用刚才的口令字,就再把它输入一次。

(10) 单击 Accept 按钮,返回到 Users Accounts 窗口。

对于新的用户,在 Base info 选项卡中比较难办的是如何填写 Supplementary groups 文本框的内容,因为在这一项中没有可以选择的项。系统管理员必须手工填写补充组。可以下拉并选择 group 项中的选项,并将这些项抄写到 Supplementary groups 中。

Supplementary Groups 项的内容是对大小写敏感,并由空格界定。

如表 17.3 所示是 Base info 选项卡中可选项的缺省值。

可选项	缺省值
group	<login name=""></login>
Supplementary groups	None
Home directory	/home/ <login name=""></login>
Command interpreter	/bin/bash
User ID	从 500 开始递增

表 17.3 Base info 选项卡中可选项的缺省值

17.5.2 修改用户属性

请按照下面的步骤修改现有用户的设置信息:

- (1) 打开 linux conf 程序。
- (2)沿着目录树前进到 Users Accounts 下的 Normal 子目录树。

(3) 单击 Users accounts 选项,出现 Users accounts 窗口。

(4)沿着用户列表前进到需要修改的账户处,单击需要修改的项目,出现如图 17.5 所示的 User information 窗口。

Apollo: Linuxconf 1.19 (subrev 2)			_ 🗆 X
Config Control Status	Users accounts User information		
	You must specify at least the login name and the full name Base info Params Privileges	e □ The account is enabled	
ber Policies Brile systems ⊢Miscellaneous services Beboot mode	Login name Full name	mxn	
	group	mxn	
	Supplementary groups		
	Home directory(opt)	/home/mxn	
	Command interpreter(opt)	/bin/bash	
	User ID(opt)	500	
Guit Act/Changes Help	Accept Cancel De	el Passwd Tasks Help	

图 17.5 linuxconf 程序的 User information 窗口

(5) 根据需要修改有关的项目。

(6) 单击 Accept 按钮即可完成对用户属性的修改。

17.5.3 禁用用户

请按照下面的步骤禁用一个用户:

- (1) 打开 linux conf 程序。
- (2)在 linux conf 程序中打开 User information 窗口。
- (3) 单击 The account is enabled 复选框去掉其中的勾号。
- (4) 单击 Accept 按钮,即可禁用该用户。

17.5.4 删除用户

请按照下面的方法通过 linuxconf 程序删除一个用户:

- (1) 打开 linux conf 程序。
- (2)在 linux conf 程序打开 User information 窗口。

(3) 单击 Delete 按钮。

(4) 在 3 种删除账户数据方法的选项中选择一个。

(5) 单击 Accept 按钮,即可删除该用户。

17.6 用 户 查 询

有时系统管理员需要了解系统中有关其他用户的信息。例如某个用户当前是否登录进 入系统。下面介绍命令可以完成用户查询的工作。

17.6.1 who命令

有的用户登录到计算机后所做的事情之一就是查看有哪些用户已经登录到系统。GNU 的 shell 实用程序提供了 who 和 users 命令用来显示当前都有哪些用户登录到系统以及他们 正在执行何种操作的有关信息。users 命令只显示所有登录在系统中的用户的名字,而 who 命令则比较复杂。who 比 users 更有用的一个特性是 who 的输出结果每行只显示一个用户 条目,它可以与 grep 或 sed 之类的命令一起使用,以处理和格式化输出结果。

who 命令通过检查/var/run/utmp 文件来生成信息。/var/run/utmp 命令跟踪当前登录进入 系统的用户。

除了能够提供有哪些用户登录到系统的信息外, who 还可以显示用户已经空闲的时间、 注册的时间以及是否允许与其他人进行交谈。who 能够识别的一些选项如下:

· -H 或--heading 显示一个标题。

· -T,-w 或--mesg 显示用户的信息状态。

· -i,-u或--id1e 显示用户空闲的时间。

who 的-u 参数在输出中增加表示用户登录后存在了多长时间的列。在下面的例子中有 4 个用户,前3 个用户已停止操作9分钟。

duwei	tty1	Apr 25 20:37	00:09
root	tty2	Apr 25 20:37	00:09
mxn	tty3	Apr 25 20:37	00:09
root	:0	Apr 25 20:08	
root	pts/0	Apr 25 20:08	

输出为以空格分隔的信息。

17.6.2 finger命令

#finger

finger 命令检查一些系统和用户定义的文件,并报告查找到的信息。下面是 finger 命令 和其输出信息:

#illiger					
Login	Name	Tty	Idle	Login TimeOffice	Office Phone
duwei		tty1	1	Apr 25 20:37	
mxn		tty3	1	Apr 25 20:37	

root	root	tty2	1	Apr 25 20:37
root	root	*:0		Apr 25 20:08
root	root	pts/0		Apr 25 20:08 (:0)

下面是 finger 的另一种用法:

#finger duwei

Login: duweiName: (null)Directory: /home/duweiShell: /bin/bashOn since Wed Apr 25 20:37 (GMT-5) on tty159 seconds idleNo mail.No Plan.

首先,finger 命令读并且解释/etc/passwd 文件。它从这个文件中得到登录 ID、注释字 段、主目录位置以及所发出的登录命令。此外,它还检查/var/run/utmp。如果有用户登录在 系统中,即显示多长时间、在哪个终端上以及来自哪里。在给出这些信息后,接着检 查.forward 文件的存在。如果文件存在,即显示其信息。然后它查看用户是否有等待读的 邮件。最后检查在用户主目录中的.plan 文件。.plan 文件包含用户在其他用户查看时可以公 开的信息。

17.6.3 id命令

id 命令也是一个经常需要用到的命令。利用 id,不但可以了解自己的 uid 和 gid,并且 还能够确定其他用户的 uid 和 gid,这对于某些工作的执行十分有用。

id 除了包含 whoami 和 groups 命令的功能外还增加了返回系统上其他用户信息的能力。 在缺省情况下, id 命令打印出当前用户的所有标识信息。还可以利用其他用户名来运行 id, 例如:

#id mxn

此时将得到下面的输出结果:

uid=500(mxn) gid=500(mxn) groups=500(mxn)

上面所述的这些信息都可以通过查看/etc/passwd 确定,但利用 id 命令是完成这一任务的一个非常简单的方法。

除此之外, id 的输出结果可以利用下面的选项来定制:

- · g 或--group 只显示用户组的 ID。
- · G 或--groups 只显示扩充用户组的 ID。
- n 或--name 显示名字而不是 id 值。
- · r 或--real 显示实际的 ID 而不是有效的 ID。
- · u 或--user 只显示用户的 ID。

所以, groups 命令就是 id -Gn 命令, whoami 命令就是 id -un 命令。

下面介绍维护用户账号安全的要点。认真按照下面的建议操作,可以最大限度地保护 账号的安全。

1.保持口令的安全需要注意的事项如下:

- (1)不要将口令写下来。
- (2)不要将口令存于终端功能键或调制解调器的字符串存储器中。
- (3)不要选取显而易见的信息作口令。
- (4)不要让别人知道。
- (5)不要交替使用两个口令。
- (6)不要在不同系统上使用同一个口令。
- (7)不要让人看见自己在输入口令。
- 2. 不要让自己的文件或目录被他人写需要注意的事项如下:
- (1) 如果不信任本组用户,将 umask 设置为 022。
- (2) 确保自己的.profile 除自己外对他人都不可读写。
- (3) 暂存目录最好不用于存放重要文件。
- (4) 确保 HOME 目录对任何人不可写。
- (5) uucp 传输的文件应加密,并尽快私人化。

3.若不希望其他用户读自己的文件或目录,就要使自己的文件和目录不允许任何人读, 需要注意的事项如下:

(1) umask 设置为 006/007。

(2) 若不允许同组用户存取自己的文件和目录,将 umask 设置为 077。

(3)暂存文件按当前 umask 设置,存放重要数据到暂存文件的程序,就被写成能确保 暂存文件对其他用户不可读。

(4) 确保 HOME 目录对每个用户不可读。

4. 不要写 SUID/SGID 程序。

5. 拷贝和移文件需要注意的事项:

(1)用 cp 拷贝文件时,记住目的文件的许可方式将和文件相同,包括 SUID/SGID 许可。如目的文件已存在,则目的文件的存取许可和所有者均不变。

(2)用 mv 移文件时,记住目的文件的许可方式将和文件相同,包括 SUID/SGID 许可。若在同一个文件系统内移文件,目的文件的所有者和小组都不变,否则目的文件的所 有者和小组将设置成本用户的有效 UID 和 GID。

(3) 小心使用 cpio 命令, 它能复盖不在本用户当前目录结构中的文件, 可用-t 选项首 先列出要被拷贝的文件。

6. 删除一个 SUID/SGID 程序时,先检查该程序的链接数,如有多个链,则将存取许

可方式改为 000, 然后再删除该程序; 或先写空该程序再删除, 也可将该程序的 i 结点号给 系统管理员去查找其他链。

7.用 crypt 加密不愿让任何用户(包括超级用户)看的文件需要注意的事项如下:

(1)不要将关键词做为命令变量。

(2) 用 ed -x 或 vi -x 编辑加密文件。

8.除了信任的用户外,不要运行其他用户的程序。

9. 在自己的 PATH 中,将系统目录放在前面。

10. 不要离开自己登录的终端。

11. 若有智能终端,当心来自其他用户,包括 write 命令、mail 命令和其他用户文件的 信息中有换码序列。

12.用 Ctrl+D 或 Exit 退出后,在断开与系统的连接前等待看到 login:提示。 13.使用 cu 版本需要注意的事项如下:

(1)不要用 cu 调用安全性更强的系统。

(2)除非确信 cu 不会被诱骗去发送文件,否则不要用 cu 调用安全性更弱的系统。

17.8 习题

1. 在命令行的方式下创建一个用户,手工指定用户的主目录, shell 和注释等内容。

2. 禁用刚才创建的用户,删除该用户。

3. 在图形用户界面下重复上面的工作

第18章 常用网络管理

本章主要介绍一些常用网络管理工具。主要内容包括使用 ping 程序检查网络连通性、 使用 ifconfig 程序查看网络接口、使用 route 程序查看路由信息、使用 traceroute 程序查看 路由信息、使用 netstat 程序查看网络统计数据以及使用 tcpdump 程序查看网络数据包等。

18.1 ping 程 序

最基本的查找并排除网络故障的工具是 ping 程序。这个程序的命名是根据潜水艇声纳的微微震动的声音而来的。ping 使用网际控制报文协议 (ICMP), 这个协议运行在 IP 上并 控制路由和可到达信息等。通过这个协议, ping 程序将发出数据包到另一个主机并等待回 复, 然后根据回复统计网络的一些状况。

具体的, ping 命令会向目标主机发送一个请求, 然后等待响应。之后目标主机发回响应, 并显示到屏幕上。ping 命令会不断地发送请求, 直到使用者用停止命令 Ctrl+C 来停止 它。如果 ping 命令不能访问一台主机, 会显示一条信息指明该主机是不可到达的。

使用 ping 命令的方法是输入 ping, 然后再在后面输入目标主机的 IP 地址或者主机名称, 如下所示:

ping 159.226.43.5

PING 159.226.43.5 (159.226.43.5) from 159.226.43.4 : 56(84) bytes of data. 64 bytes from 159.226.43.5: icmp_seq=0 ttl=128 time=430 usec 64 bytes from 159.226.43.5: icmp_seq=1 ttl=128 time=401 usec 64 bytes from 159.226.43.5: icmp_seq=2 ttl=128 time=404 usec 64 bytes from 159.226.43.5: icmp_seq=3 ttl=128 time=391 usec 64 bytes from 159.226.43.5: icmp_seq=4 ttl=128 time=406 usec 64 bytes from 159.226.43.5: icmp_seq=5 ttl=128 time=405 usec

64 bytes from 159.226.43.5: icmp_seq=6 ttl=128 time=355 usec

--- 159.226.43.5 ping statistics ---

7 packets transmitted, 7 packets received, 0% packet loss round-trip min/avg/max/mdev = 0.355/0.398/0.430/0.033 ms

可以看到, ping 命令将报告出它发送的每一个数据包从对方计算机返回所需要的时间 (以千分之一秒为计时单位)。

ping 程序有许多选项,经常使用的选项如表 18.1 所示。

表 18.1 常用的 ping 程序选项

选项	功能
-c	只是发送一定数量的数据包而不是一直运行
-n	显示 IP 地址而不是主机名。当 DNS 服务器有问题时有用
-r	记录路由。在每一个数据包中加入该选项指示在源和目标之间的所有主机将它们的 IP 地
	址加到数据包中。这样就可以得知数据包通过了哪些主机。数据包的大小限制了只能记
	录9个主机
-q	禁止输出。只显示最终的统计结果
	"

通过 ping 程序,可以检查自己的主机是否可以与网络上其他主机进行连接。如果 ping 命令失败,则说明网络工作得不正常。可能是由于某个网络接口的问题、配置的问题、或者是物理连接的问题。

当检查网络问题时,请首先 ping 主机自己的 IP 地址。这可以检测始发的网络接口的 设置是否正确。在这之后,可以试着 ping 缺省网关、缺省网关的网关等等,直到到达目的 主机。这样可以容易地判断出问题的所在。当然,一旦检测到可以到达缺省网关,最好使 用 traceroute 程序自动进行这个进程。

所有的 TCP/IP 数据包包含名为 Time-To-Live 或 TTL 的项。每经过一个路由器这个值 就被递减一次。当这个值为 0 时,数据包就被丢弃了。ping 中使用的缺省 TTL 为 255 (这 是最大值),其他的很多程序如 telnet 或 ftp 使用更小的 TTL 值(通常为 30 或 60)。这也就 是说使用 ping 能到达的主机, telnet 或 FTP 未必能到达主机。可以使用 ping 命令的-ttl 选 项为它发出的数据包设置 TTL。

18.2 ifconfig命令

目录/sbin 中的 ifconfig 命令通常用来在网络中配置网络界面的,但是也可以使用 ifconfig 命令来查看各个接口连接的状态信息。

当运行某些网络应用程序时,虽然程序一直在发送和接收着数据,但是屏幕上的内容 根本不发生变化,这时,ifconfig 命令就十分有用,可以用它来检查接口状况。

18.2.1 查看接口配置

如果想使用 ifconfig 命令, 输入下面的命令即可:

#ifconfig

eth0

 Link encap:Ethernet HWaddr 00:00:1C:0A:1F:38
 inet addr:159.226.43.4 Bcast:159.226.43.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:345 errors:0 dropped:0 overruns:0 frame:0 TX packets:243 errors:0 dropped:0 overruns:0 carrier:0 collisions:1 txqueuelen:100 Interrupt:11 Base address:0xb000

lo Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 UP LOOPBACK RUNNING MTU:3924 Metric:1 RX packets:118 errors:0 dropped:0 overruns:0 frame:0 TX packets:118 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0

ifconfig 命令后面可以接很多参数,通常用到的参数是接口名称、IP 地址以及其他的选项。ifconfig 命令之后以该 IP 地址定义这个网络接口的地址,系统就会知道对应的接口以及这个接口的 IP 地址。

在 Linux 中,所有的网络接口名包含驱动程序名和一个数字。表 18.2 是在 Linux 中支 持的一些网络接口。

接口名称	设备类型
eth	以太网
tr	令牌环网
ррр	点对点协议
slip	串行线路接口协议
plip	并行线路接口协议

表 18.2 Linux 支持的网络接口

接口中的数字从 0 开始并按系统发现它们的顺序递增。缺省状态下, Linux 内核只会发现一个网络接口。如果有多于一个的网卡, 要将如下这行加入到/etc/lilo.conf 文件中并重新运行/sbin/lilo 命令:

append="ether=IRQ,I/O,eth1 ether=IRQ,I/Q,eth2"

这告诉内核要加入两个以太网设备 eth1 和 eth2,它们在所声明的 IRQ 和 I/O 地址上。 如果想让系统自动检测网卡的 I/O 地址和 IRQ,可以将 IRQ 和 I/O 都设为 0。

如果 ifconfig 在调用时只有接口名,它就会显示出该接口的配置信息。如果调用 ifconfig 时不带任何参数,它就会显示迄今为止已配置的所有接口,-a 选项也将强制它显示所有非 活动的接口。以太网接口 eth0 的调用示例如下:

#ifconfig eth0

athO	Link on conjecthormot IW addr 00:00:1C:0A:1E:29
etho	Link encap:Eulement H wadur 00:00:1C:0A:1F:58
	inet addr:159.226.43.4 Bcast:159.226.43.255 Mask:255.255.255.0
	UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
	RX packets:20 errors:0 dropped:0 overruns:0 frame:0
	TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:100
	Interrupt: 11 Base address:0xb000

MTU(最大传输单元)和 Metric(度量值)字段显示的是该接口当前的 MTU 和度量 值的值。按照惯例,度量值供某些操作系统所用,用于计算一条路由的成本。

RX 和 TX 这两行显示了已经准确无误地收发了多少数据包、发生了多少错误、丢失了 多少数据包,以及多少包是因为传输过载而遗失的。

使用 ifconfig -a 命令可以得到所有激活的接口的状态信息。所有的用户都可以使用这 种模式的 ifconfig 命令。它显示接口的所有配置信息,包括接口自己的 IP 地址、子网掩码、 广播地址和物理地址。它也显示接口的状态,如接口是否被使用和是否为回送接口。它也 可以显示其他的信息,如最大传输单元、网卡 I/O 地址和 IRQ 号、接收和发出数据包的数 量和冲突。也可以使用 ifconfig -a 命令得到一个接口的状态信息。这将把所有处在激活状 态的接口的参数显示出来。下面是 ifconfig -a 命令的输出示例:

ifconfig -a

eth0Link encap:EthernetHWaddr 00:00:1C:0A:1F:38inet addr:159.226.43.4Bcast:159.226.43.255Mask:255.255.255.0UP BROADCAST RUNNING MULTICASTMTU:1500Metric:1RX packets:117 errors:0 dropped:0 overruns:0 frame:0TX packets:38 errors:0 dropped:0 overruns:0 carrier:0collisions:0 txqueuelen:100Interrupt:11 Base address:0xb000loLink encap:Local Loopbackinet addr:127.0.0.1Mask:255.0.0.0UP LOOPBACK RUNNINGMTU:3924Metric:1RX packets:46 errors:0 dropped:0 overruns:0 frame:0TX packets:46 errors:0 dropped:0 overruns:0 frame:0TX packets:46 errors:0 dropped:0 overruns:0 frame:0

18.2.2 配置接口

一个接口可以在不进行重新配置的情况下临时地变为不可用和再变为可用,这可以用 于将服务器的网络连接临时变为不可用。使用下列命令实现该功能:

#ifconfig <interface> down
#ifconfig <interface> up

up 和 down 选项用于激活或取消这个接口。这种形式的 ifconfig 命令只有 root 用户才可以使用。

此外,还可以指定这个 IP 地址是一个主机地址还是一个网络地址:

#ifconfig interface -host_net_flag address options

其中 host_net_flag 可以是-net 或-host,应用来指定 IP 地址类型:是网络地址还是主机 地址。缺省的情况下使用-host。ifconfig 命令可以带有几个选项,用来设置接口的不同特性。 例如最大传输单元(MTU)或广播地址等等。

在下面的例子中, ifconfig 命令用于配置一个以太网接口:
#ifconfig eth0 204.32.168.56

对于这种简单的配置, if config 会自动创建一个标准的广播地址和子网掩码。标准的广 播地址是网络地址加入值为 255 的主机地址。标准的子网掩码是 255.255.255.0。但是, 如 果连接的网络是一个特殊的网,则需要在使用 if config 命令时指定这些值。指定广播地址 的选项是 broadcast,指定子网掩码的选项是 netmask。在下面的例子中, if config 命令包含 子网掩码和广播地址。

#ifconfig eth0 204.32.168.56 netmask 255.255.255.0 broadcast 204.128.244.127

对于点对点接口,例如并行 IP (PLIP), 串行线 IP (SLIP)以及点对点协议(PPP), 需要指定 pointtopoint 选项。PLIP 接口名称的 plip 开头后接一个数字,例如 plip0 是第一个 PLIP 接口; SLIP 接口使用 slip0; PPP 接口从 ppp0 开始。点对点接口通常用于连接两台主 机。例如,通过调制解调器连接的两台计算机,在指定 pointtopoint 选项的时候,需要指定 主机的 IP 地址。

在下面的例子中, PLIP 接口被配置为连接 IP 地址为 199.35.209.72 的主机,另一台主机的 IP 地址为 204.166.254.14。

#ifconfig plip0 199.35.209.72 pointopoint 204.166.254.14

还可以通过 ifconfig 命令来配置回送设备。回送设备的名称为 lo, 它的 IP 地址是固定的 127.0.0.1。下面是一个配置回送接口的示例:

#ifconfig lo 127.0.0.1

ifconfig 命令对显示接口的状态也是很有帮助的。如果输入 ifconfig 命令,后面接对应接口名称,会显示这个接口的有关信息。

要检查回送接口是否已被配置,只需在命令 ifconfig 后面接回送设备名称 lo 即可,如 下所示:

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0 UP LOOPBACK RUNNING MTU:3924 Metric:1 RX packets:46 errors:0 dropped:0 overruns:0 frame:0 TX packets:46 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0

18.2.3 设置IP别名

有时会有一个接口对应多个 IP 地址的情况。例如 , 使运行着多个服务的服务器 , 让客 户通过不同的 IP 地址来访问每个服务 , 可以使将来对接口的重新配置变得容易。

Linux 和大多数的 UNIX 一样提供了网络别名的特征。为了使用网络别名,首先要重新 配置和重新编译内核,并在网络选项配置部分启用网络别名和 IP:Aliasing Support 支持。这 些选项可以编译进内核也可以编译为模块。

如果 Linux 运行在支持别名的内核上,使用简单的 ifconfig 命令就可以创建一个别名。

所要做的就是追加一个冒号和针对接口名的别名号。下面是一个示例:

ifconfig eth0:0 10.1.1.1 netmask 255.255.255.0 broadcast 10.1.1.255

这个命令使用给出的参数为以太网接口 eth0 创建了 eth0:0 别名。

为了在系统每次启动时就自动产生别名,可以将建立别名的命令加入到 /etc/init.d/network 中。

18.2.4 ifconfig命令选项列表

如表 18.3 所示是 ifconfig 命令的所有选项,其对应的特性可以打开也可以取消,要只在选项名前加一个破折号(-)即可。

参数	作用
interface	接口的名称,通常位于/dev目录下。例如/dev/eth0
up	标志接口处于 up 状态,也就是说,IP 层可以对其进行访问。这个选项用于命令
	行上给出一个地址之时。如果这个接口已被 down 选项临时取消(与该选项对应
	的标记是 UPRUNNING) , 还可以用于重新启用一个接口
down	标志接口处于 down 状态,也就是说,IP 层不能对其进行访问。这个选项有效地
	禁止 IP 通信流通到这个接口。注意:它并没有自动删除利用该接口的所有路由
	信息。如果永久性地取消了一个接口,就应该删除这些路由条目,并在可能的情
	况下提供备用路由
netmask 标	分配子网掩码,供接口所用。要么给出一个前面是 0x 的 32 位十六进制号码,要
	么采用只适用于两台主机所用的点分四段式号码。 对于 SLIP 和 PLIP 接口 ,这个
	选项是必须配置的
pointtopoint address	用于只涉及两台主机的点到点链接。对于 SLIP 和 PLIP 接口 ,这个选项是必须配
	置的 (如果已经设置了一个点到点地址 , ifconfig 就会显示出 POINTTOPOINT
	标记)
broadcast ddress	广播地址通常源于网络编号,通过设置主机部分的所有位得来,有的 IP 采用的
	方案有所不同,这个选项可适用于某些奇怪的环境(如果已经设置了广播地址,
	inconfig 就会显示出一个 BROADCAST 标记)
metric number	可用于为接口创建的路由表分配度量值。路由信息协议(RIP)利用度量值来构
	建网络路由表。ifconfig 所用的默认度量值是 0。如果不运行 RIP 程序,就没必
	要采用这个选项;如果要运行 RIP 程序,就尽量不要改变这个默认的度量值
mtu bytes 标	用于设置最大传输单元,也就是接口一次能处理的最大字节数。对于以太网接口,
	MTU 的默认设置是 1500 个字节;对于 SLIP 接口,则是 296 个字节
arp	专用于以太网或包广播之类的广播网络。它启用 ARP(地址解析协议)来保护
	网络上各台主机的物理地址。对于广播网 , 默认设置是 on
-arp	取消在接口上使用 ARP

表 18.3 ifconfig 可以识别的选项

参数	作用
promisc	将接口置入 promiscuous(混乱)模式。在广播网中,这样将导致该接口接收所
	有的数据包 , 不管其目标是不是另一台主机。 允许利用包过滤器和所谓的以太网
	窥视技术对网络通信进行分析
-promisc	取消混乱模式
allmulti	多播地址,即向不在同一个子网上的一组主机广播数据,尚未获得内核支持,对
	应的标记是 ALLMULTI
-allmulti	取消多播地址
trailers	打开以太网帧中的跟踪位
-trailers	关闭以太网帧中的跟踪位
dstaddr address	点对点连接的目标 IP 地址
hw	设置硬件地址

18.3 route程序

数据包要到达目标地址必须经过一定的路由,路由决定了这个过程的开始以及要到达的目标主机,以及中间哪个计算机要进行数据包的转发。

在小型的网络中,路由信息可能是静态的,从一个系统到另一个系统的路由是固定的。 在大型的网络中,数据包要到达目标地址主机,中间要经过许多计算机,路由信息是动态 的。对于静态路由,系统必须清楚所有的网络连接;对于动态路由,系统不需要知道很多 的网络连接状况。路由信息在路由表文件/proc/net/route 中,使用 route 命令可以显示或者 修改路由信息。

18.3.1 显示路由信息

要显示路由表内容,可以直接在命令行上输入没有任何选项的 route 命令。这种形式的 命令所有用户都可以使用,如下所示:

route Kernel routing table DestinationGatewayGenmaskFlagsMetricRefUseIface loopback*255.0.0.0 U 0 0 12 lo pangol.train.com*255.255.255.0 U 0 0 0 eth0

可以看出,路由表中的每一项都是由几个域构成的,它提供了路由目标和接口类型等 信息。如表 18.4 所示是各个域及其说明。

(续表)

表 18.4 route 命令输出的域及其说明

域	含义
Destination	表明路由的终点。 如果在文件/etc/hosts 或/etc/networks 中包含有对应的项,这个名称将
	会被替换。default 表示缺省网关
Gateway	表明数据包传送到目标要经过的网关。星号 (*) 表明数据包直接发送到了目标主机上
Gateway	表明数据包传送到目标要经过的网关。星号 (*)表明数据包直接发送到了目标主机上
Genmask	表明应用于这个路由的掩码。这个掩码作用于 Destination 列中的值
Flags	可能含有多个值。常用的标志为:
	U 路由是可用的
	H 目标为一主机。这是到指定主机的静态路由
	G 使用网关。数据包不会被直接送到目标主机,而是使用网关代替
Metric	表明到目标的距离。由一些路由守护程序自动计算出到达目标的最佳路由
Ref	在 Linux 中并没有使用,在其他的 UNIX 系统中,它表示这个路由引用的数量
Use	内核寻找路由所使用的时间量
Iface	表明数据包进行传输的接口

18.3.2 添加和删除路由信息

修改路由信息,可以使用 route 命令在路由表中加入或删除路由。使用如下格式的命令 完成这种操作:

route add|del [-net|-host] <target> [gw <gateway>] [netmask <netmask>] [dev] <interface>

add 和 del 命令将分别表明增加还是删除一个路由。

可选的选项-net 或-host 选项表明是使操作在一个网络路由中进行还是对一个主机路 由。

target 参数可以是目标的主机地址或网络数。可以使用关键字 default 作为目标来设置 或删除缺省路由。

可选的 gateway 参数表明这个路由要使用的网关。如果省略这个参数,route 命令将假 设主机或网络是直接连到本机的。当使用 ifconfig 命令配置接口后,为本地网络增加一个 路由是很重要的,如下所示:

ifconfig eth0 10.1.1.1 netmask 255.255.255.0 broadcast 10.0.1.255 # route add -net 10.1.1.0

正像其名称所表明的,可选的 netmask 选项设置这个路由的掩码,这个掩码作用于 target 地址。如果省略这个选项,将使用 IP 地址的缺省掩码或接口的掩码。

可选的 dev 参数设置了数据包传送到目的地要使用的接口。如果省略, route 命令将在路由表中查找一个可以到达 gateway 的路由。如果没提供 gateway, 它会查找直接能到达 target 的接口。

例如,在下面的示例中,将回送接口的 IP 地址加入路由表。

route add 127.0.0.1

如果增加一个特定的静态路由,需要指定子网掩码、网关、接口设备或目标地址。但 是如果接口已经通过 ifconfig 命令启动,那么 ifconfig 命令能够产生大部分信息。例如,要 为一个已经由 ifconfig 命令设置过的以太网连接配置路由,只需输入-net 选项和目标的 IP 地址。ifconfig 命令通过 IP 地址定位接口,并用这个信息建立路由。下面的示例是一个以 太网接口的路由。

route add -net 204.32.168.0

如果系统连接到网络,应该至少有一项缺省路由在路由表中。当其他路由无法将数据 包发送给目标地址时,就会使用这个缺省路由。缺省路由的关键字是 default。

下面的命令将路由表中的某项路由信息删除:

route del -net 204.32.168.0

18.4 traceroute程序

traceroute 程序是 TCP/IP 网络查找并排除故障的的得力工具。它不断用更大的 TTL 发送 UDP 数据包并探测数据经过的网关的 ICMP 回应,最后得到数据包从源主机到目标主机的路由信息。

它的工作原理是:traceroute 首先发送 TTL 为1的数据包,数据到达一个网关,这个网 关可能是目标主机或不是。如果为目标主机,这个网关发送一个回应数据包;如果不是目 标主机,网关将递减 TTL。因为 TTL 现在为0,网关删除数据包并返还一个数据包声明此 事。不管发生了什么,traceroute 程序都会探测到回复数据包。如果数据已经到达目标主机, 它的任务完成了;如果没有到达目标主机,它将 TTL 递增1(这时为2)并发送另一个数 据包。这一次第一个网关递减 TTL(到1)并将数据包传送到下一个网关。这个网关将做 同样的事:确认是否为目标主机并递减 TTL。这个过程将一直进行,直到到达目标主机或 TTL 到达它的最大值(默认时为30,可以使用-mmax_ttl 选项进行修改)。

对于每个 TTL, traceroute 将发送 3 个数据包并报告每个数据包所花费的往返时间。这 个功能可以用来检测网络瓶颈。

traceroute 通常使用和 ping 一样的方式,将目标地址作为命令参数。

从 traceroute 的输出可以得到许多信息。可以使用 traceroute(8)的帮助手册页获得帮助。

18.5 netstat程序

18.5.1 使用netstat

从名称上可以看出 netstat 命令是一个检查网络活动、连接、路由表以及其他网络消息 和统计数字的命令。如果对网络中随时发生的事情有兴趣,可以使用这个命令。netstat 命

令显示了所有 TCP/IP	网络服务的状态。	根据所需要显示的内容	,它提供了一些参数。	例如
如果不提供任何参数,	, 将会看到下面的	内容(有节略):		

#netstat							
Active Internet connections (w/o servers)							
Proto Recv-Q Send-Q Local Address			Foreign Addre	ess	State		
Activ	e UNIX	domain socke	ets (w/o servers)				
Proto	RefCnt	Flags	Туре	State	I-Node	Path	
unix	1	[]	STREAM	CONNECTED	1024	@0000003f	
unix	0	[]	STREAM	CONNECTED	213	@0000001d	
unix	1	[]	STREAM	CONNECTED	1396	@00000061	
unix	1	[]	STREAM	CONNECTED	1324	@00000050	
unix	1	[]	STREAM	CONNECTED	1401	/tmp/orbit-root/	
						orb-1830525694495327456	
unix	1	[]	STREAM	CONNECTED	1397	/tmp/orbit-root/	
						orb-200047408823580672	
unix	1	[]	STREAM	CONNECTED	1335	/tmp/.X11-unix/X0	
unix	1	[]	STREAM	CONNECTED	1330	/tmp/orbit-root/	
						orb-2014894003960655160	
unix	1	[]	STREAM	CONNECTED	1004	/tmp/.ICE-unix/875	
unix	0	[]	DGRAM		885		
unix	1	[]	STREAM	CONNECTED	884	/tmp/.X11-unix/X0	
unix	1	[]	STREAM	CONNECTED	864	/tmp/.X11-unix/X0	
unix	0	[]	DGRAM		832		
unix	0	[]	DGRAM		780		
unix	0	[]	DGRAM		653		
		_					

不带选项的 netstat 命令会显示系统上的所有网络连接。首先是活动的 TCP 连接, 之后 是活动的域套接字, 域套接字包含配置系统间通信的过程。命令显示的 6 个字段如表 18.5 所示。

表 18.5 netstat 命令显示的字段

字段	含义
Proto	用于连接的协议 TCP 和 UDP
Recv-Q	系统接收但还没有使用的字节数
Send-Q	系统发送给远程主机的还未确认接收的字节数
Local Address	本地主机名和端口号
Foreign Address	远程主机名和端口号。端口号可以是连接类型,例如 telnet 或者 ftp
state	与远程主机连接的状态。可选项有:ESTABLISHED,连接已经建立;SYNSENT,
	尝试创建连接;SYN_REC,正在创建连接;FIN_WAIT1,关闭连接;CLOSE,连接
	已被关闭;LISTEN,监听远程的连接请求;UNKNOWN,未知状态域套接字
Proto	套接字的协议。通常是 unix

1	(赤王)	`
C	头化)

字段	含义
RefCnt	当前在套接字中进程的数目标记
Туре	访问的套接字的类型
State	套接字的状态,可选项有:FREE,套接字没有被使用; LISTEING,等待连接;
	UNCONNECTED,当前没有连接;CONNECTING,尝试创建连接;CONNECTED,
	当前连接已经建立;DISCONNECTED,关闭一个连接
Path	进程用的访问套接字的路径名

18.5.2 netstat选项

netstat 可以接受许多选项,这些选项决定了显示的内容。这些选项及其说明如表 18.6 所示。

表 18.6 netstat	接受的选项及其说明
----------------	-----------

选项	说明
-a	显示所有 Internet 套接字的信息
-I	显示所有网络设备的统计数字
-C	不断地显示当前网络状态,直到该程序被中断
-n	显示 IP 地址
-0	显示定时的状态和过期时间
-r	显示核心路由表信息
-t	显示 TCP 套接字的信息
-u	显示 UDP 套接字的信息
-V	显示版本信息
-W	显示原始套接字信息
-X	显示 Unix 域套接字信息

下面将详细介绍一些 netstat 常用的选项。

1. netstat -a 选项

netstat -a 列出所有打开的或监听的套接字,而非只是那些有连接的。它显示的套接字的信息内容如下(有节略):

Active Internet connections (servers and established)							
Proto Recv-Q Send-Q Local Address				ss l	Foreign Address	State	
tcp	0	0	*:X	;	*:*	LISTEN	
udp	0	0	*:sunrpc	;	*:*		
raw	0	0	*:icmp	;	*:*	7	
Activ	Active UNIX domain sockets (servers and established)						
Proto RefCnt Flags Type			State	I-Node	Path		
unix	0	[ACC]	STREAM	LISTENING	G 856	/tmp/.X11-unix/X0	

unix 0 [[ACC] STREAM	I LISTENING	1261					
/tmp/orbit-roo	ot/orb-16223567811	671505799						
unix 0 [[ACC] STREAM	I LISTENING	669	/dev/	gpmctl			
unix 1 [[] STREAM	I CONNECTED	1267	@00	00005d			
unix 1 [[] STREAM	I CONNECTED	861	@00	000031			
unix 1 [[] STREAM	I CONNECTED	1229	/tmp/	.ICE-unix	/873		
unix 1 [[] STREAM	I CONNECTED	862	/tmp/	.font-unix	/fs7100	С	
unix 1 [[] STREAM	I CONNECTED	864	/tmp/	X11-unix/	/X0		
unix 0 [[] DGRAM		832					
unix 0 [[] DGRAM		819					
unix 0 [[] DGRAM		780					
2 netstat -	e选顶							
2 . netstat				- · -				
netstat -e 列	」出当前使用套控	爱字的用户。它	显示的信	息如	ト(有节	5略)	:	
Active Interne	t connections (w/o s	servers)						
Proto Recv-Q	Send-Q Local Addr	ess Fore	ign Address		State		User	Inode
tcp 0	0 159.226.43.	4:ftp 159.	226.43.5:10	70	ESTABL	ISHED) root	1284
udp 0	0 159.226.43	4:1030 159.	226.43.4:do	main	ESTABL	ISHED) root	1400
udp 0	0 159.226.43	4:1030 159.	226.43.4:do	main				
Active UNIX	domain sockets (w/	o servers)						
Proto RefCnt I	Flags Type	State	I-No	de	Path			
unix 0 [[] STREAM	I CONNECTED	213		@000000)1d		
unix 0 [[] DGRAM		1286	5				
unix 1 [[] STREAM	I CONNECTED	1280)				
/tmp/orbit-roo	ot/orb-16223567811	671505799						
unix 1 [[] STREAM	I CONNECTED	1252	2	/tmp/.X11	l-unix/	X0	
unix 1 [[] STREAM	I CONNECTED	1239)				
unix 1 [[] STREAM	I CONNECTED	864		/tmp/.X11	l-unix/	X0	
unix 0 [[] DGRAM		832					
unix 0 [[] DGRAM		819					
3 . netstat -	r 选项							
netstat -r 列	出路由表。它显	显示的信息如下	:					
#netstat -r								
Kernel IP rout	ing table							
Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface	
159.226.43.0	*	255.255.255.0	U	0	0	0	eth0	
127.0.0.0	*	255.0.0.0	U	0	0	0	lo	
default	159.226.43.4	0.0.0.0	UG	0	0	0	eth0	

在 netstat -r 的输出结果中,第2列展示的是路由条目所指的网关。如果没有使用网关,就会出现一个星号(*)。第3列展示路由的概述。在为具体的 IP 地址找出最恰当的路由时,内核将查看路由表内的所有条目。在对找到的路由与目标路由比较之前,将对 IP 地址和

genmask 进行按位"与"计算。

第4列显示了不同的标记,这些标记的说明如下:

- · G 路由将采用网关。
- · U 准备使用的接口处于"活动"状态。
- H 通过该路由只能抵达一台主机。例如,回送接口的路由器条目 127.0.0.1 就如此。
- · D 如果路由表的条目是由 ICMP 重定向消息生成的,就会设置这个标记。
- · M 如果路由表条目已被 ICMP 重定向消息修改,就会设置这个标记。
- 4. netstat -i 选项

netstat -i 列出网络接口和每一个接口的统计信息。它显示了和 ifconfig 得出的同样的统计信息,但以表的形式出现,便于分析。

在随-i 标记一起调用时, netstat 将显示网络接口的当前配置特性。除此以外, 如果调用时还带-a 选项, 它将输出内核中所有接口,并不只是当前配置的接口。netstat -i 的显示结果如下:

#netstat -i

Kernel Interface table

eth0 Link encap:Ethernet HWaddr 00:00:1C:0A:1F:38

inet addr:159.226.43.4 Bcast:159.226.43.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:458 errors:0 dropped:0 overruns:0 frame:0

TX packets:127 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:100

Interrupt:11 Base address:0xb000

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

UP LOOPBACK RUNNING MTU:3924 Metric:1

RX packets:174 errors:0 dropped:0 overruns:0 frame:0

TX packets:174 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

MTU 和 Met 字段表示的是接口的 MTU 和度量值值。RX 和 TX 这两列表示的是已经 准确无误地收发了多少数据包(packets);产生了多少错误(errors);丢弃了多少包(dropped); 由于误差而遗失了多少包(overruns)。

最后一列展示的是为这个接口设置的标记。在利用 ifconfig 显示接口配置时,这些标记都采用一个字母。它们的说明如下:

- · B 已经设置了一个广播地址。
- · L 该接口是一个回送设备。
- M 接收所有数据包(混乱模式)。
- N 避免跟踪。

- · O 在该接口上禁用 ARP。
- P 这是一个点到点链接。
- R 接口正在运行。
- · U 接口处于"活动"状态。

18.6 tcpdump程序

tcpdump 是另一个有效的调试某些类型的网络问题的工具。它的基本工作模式为监听 网络,查看通过的数据包或对数据包进行操作。它可以将所看到的整个通信数据或其中的 令人感兴趣的部分存储起来,或对它们所包含的信息进行基本的分析。

tcpdump 程序将网卡设置为混杂(promiscuous)模式状态来实现自己的功能。一般网 卡只能看到与其相关的数据包。但是在混杂模式下,它可以看到通过网络的所有数据包, 并可以将它们传递到操作系统之上。操作系统将数据包传送给 tcpdump,而 tcpdump 就可 以过滤、显示或存储它们。因为它要改变对网卡的设置,所以必须应由 root 用户来运行。

使用 ifconfig 命令可以检测网络是否处在混杂模式。当接口为混杂模式时, PROMISC 标志就会显示出来。

使用没有参数的 tcpdump 命令将得到在网络中传输的所有数据包的列表。

在缺省情况下,tcpdump命令将把 IP 地址转换为主机名,并把端口号转换为服务名。

在一些情况下, tcpdump 所使用的查看字节数(68)不足以解码数据包。在这种情况下,可以使用-s选项来增加这一数值。

通常并不想看到所有的数据包,特别是在大地网络中。有时只是想看到在两个特定主机之间传输的数据包,或使用特定服务的数据包。tcpdump 命令提供一个可选的过滤表达式来选择一定的数据包。

tcpdump 中的过滤表达式是由关键字 and、or 和 not 连接起来的基本元素。每个元素包 含一个后面跟着 ID 的限定词。限定词包含一个或多个关键字。如表 18.7 所示是一些常用 的限定词。

限定词	匹配
src host	数据包发出主机的 IP 地址
dst host	数据包目标主机的 IP 地址
host	源或目标主机中端口的 IP 地址
src port	数据包发出的端口
dst port	数据包目标端口
port	源或目标端口

表 18.7 tcpdump 中经常使用的限定词

在 tcpdump 命令中也提供一些选项,其中常用的选项如表 18.8 所示。

表 18.8 常用的 tcpdump 选项

选项	作用
-c count	在接收了 conut 个数据包之后退出
-i interface	监听 interface。在缺省情况下,tcpdump 监听除了回送接口之外第一个被它找到的接口。
	使用 ifconfig -a 命令得到所找到的接口及其顺序
- n	不用将数字地址和端口号转换为主机和服务名
- N	只打印主机名 , 而不包含其域名
-r file	从 file 中读入数据。file 是由-w 选项创建的
-S	snaplen 从每个数据包中抓取 snaplen 字节。缺省值为 68,对于 IP、ICMP、TCP 和 UDP
	数据包足够用了。对特定的协议,这将会使一些协议信息被截断。当协议察觉到协议信
	息部分被截断时,它将会被标识为[protocol]
- V	详细模式。将显示更多的数据包信息
-w file	将数据包捕获到 file
- X	用十六进制打印数据包

18.7 习题

- 1. 使用 ping 程序及其参数测试本机和远端计算机的连通性。
- 2. 使用 ifconfig 程序查看本机的网络接口配置。
- 3. 使用 netstat 程序查看本机的网络统计信息。

第 19 章 shell 编程

本章主要介绍 Linux 环境下如何进行 shell 编程。内容包括 shell 编程简介、创建和运行 shell 程序以及 shell 编程规则等。

19.1 shell编程简介

shell 是一种命令解释器,它提供了用户和操作系统之间的交互接口。shell 是面向命令 行的,而 X Window 则是图形界面。在命令行输入命令, shell 进行解释, 然后送往操作系 统执行。

shell 可以执行 Linux 的系统内部命令,也可以执行应用程序。可以利用 shell 编程, 执行复杂的命令程序。从 Linux 终端命令行每次输入一个命令,就可得到系统的一次响应。 但可能会从事一些例行的重复性命令,或者需要一个接一个地连续执行命令,以便得到最 后的结果。这时,用户会希望将这些重复或连续的命令写成一种类似批处理的命令,只要 执行这个命令就等于执行这些命令。

Linux 中的 shell 脚本 (shell Script)就是这种批处理命令。shell 脚本是放在一个文件 中的一系列 Linux 命令或实用程序。执行 shell 脚本时,可以通过 Linux 一个接一个地解释 和执行每个命令。

shell 的功能强大,和一般的高级语言不相上下,可以像在 Linux 下执行任何其他命令 一样编写并执行 shell 脚本。可以在 shell 脚本内执行其他的 shell 脚本,只要这些 shell 脚本 在查找路径中就行。shell 脚本跟其他任何语言一样具有自己的语法。如定义变量并分配各 种值等。

shell 脚本是以文本方式存储的,而非二进制文件。所以 shell 脚本必须在 Linux 系统的 shell 下解释执行。不同 shell 脚本会有一些差异,所以不能将写给一个 shell 的脚本用另一 个 shell 执行。

在 Red Hat Linux 系统中经常使用是 Bourne again shell (bash),本章结合 bash 和其他常用 shell 来介绍 shell 编程。

19.2 创建和运行shell程序

Red Hat Linux 中包含了各种灵活而强大的 shell。如表 19.1 所示是 Linux 中 shell 的描述和它在文件系统中的位置。

shell 名称	描述	位置
ash	一个小 shell(和 sh 相似)	/bin/ash
ash.static	一个不依靠软件库的 ash 版本	/bin/ash.static
bash	Bourne Again shell	/bin/bash
bsh	ash 的一个符号链接	/bin/bsh
csh	Cshell 和 tcsh 的一个符号链接	/bin/csh
ksh	Public Domain Korn shell	/bin/ksh
	Korn shell	/usr/bin/ksh
pdksh	一个 ksh 的符号链接	/usr/bin/pdksh
rsh	受限制的 shell (针对网络操作)	/usr/bin/rsh
sh	bash 的一个符号链接	/bin/sh
tcsh	和 csh 兼容的 shell	/bin/tcsh
zsh	一个和 csh、ksh 和 sh 兼容的 shell	/bin/zsh

表 19.1 Red Hat Linux 中的 shell

下面是几个主要的 shell 提供的基本功能和其各自的特点。

- · Bourne shell 是由 Steven Bourne 编写的 ,是 Unix 的缺省 shell。Bourne shell 的 shell 的编程能力很强。但它不能处理命令的用户交互特征。bash 是 Bourne shell 的增强版。
- · C shell 是由加利福尼亚大学伯克利分校的 Bill Joy 编写的。它能提供 Bourne shell 所不能处理的用户交互特征,如命令补全、命令别名和历史命令替换等。很多人 认为, C shell 的编程能力不如 Bourne shell,但它的语法和 C 语言类似,所以 C 程 序员将发现 C shell 很顺手。tcsh 是 C shell 的增强版本,它和 C shell 完全兼容。
- Korn shell 是由 Dave Korn 编写的。Korn shell 融合了 C shell 和 Bourne shell 的优 点,并和 Bourne shell 完全兼容。Korn shell 的效率很高,其命令交互界面和编程 交互界面都很不错。
- Public Domain Korn shell (pdksh) 是 Korn shell 的增强版。
- Bash 是大多数 Linux 系统的缺省 shell。它克服了 Bourne shell 的缺点,又和 Bourne shell 完全兼容。在 Red Hat Linux 7.0 中, Bash 的版本是 2.04.11。 Bash 有以下特点:

1.补全命令行。

在 bash 命令提示符下输入命令或程序名时,不必输全命令或程序名,按 Tab 键, bash 将自动补全命令或程序名。

2. 可以使用通配符。

在 bash 下可以使用通配符* 和?。*可以替代多个字符,?则替代一个字符。

3.历史命令。

bash 能自动跟踪每次输入的命令,并把输入的命令保存在历史列表缓冲区。缓冲区的 大小由 HISTSIZE 变量控制。每次登录后,home 目录下的.bash_history 文件将初始化历 史列表缓冲区。也可以通过 history 和 fc 命令执行和编辑历史命令。 4. 可以使用别名。

在 bash 下, 可用 alias 和 unalias 命令给命令或可执行程序起别名和清除别名。这样用 户可以用自己习惯的方式输入命令。

5. 输入/输出重定向。

输入重定向用于改变命令的输入,输出重定向用于改变命令的输出。输出重定向更为 常用,它经常用于将命令的结果输入到文件中而不是屏幕上。输入重定向的命令是:<,输 出重定向的命令是:>。

6. 可以使用管道。

管道用于将一系列命令连接起来。也就是把前面命令的输出作为后面命令的输入。管 道的命令是:|。

7. 可以使用提示符。

bash 有两级提示符。第一级提示符就是用户登录 shell 时见到的,缺省为\$。可以通过 重新给 ps1 变量赋值来改变第一级提示符。当 bash 需要进一步提示以便补全命令时,会显 示第二级提示符。第二级提示符缺省为>,用户可以通过重新给 ps2 变量赋值来改变第二级 提示符。一些特殊意义的字符也可以加入提示符赋值中。

8.进行作业控制。

作业控制是指在一个作业执行过程中控制执行的状态。可以挂起一个正在执行的进程, 并在以后恢复该进程的执行。按下 Ctrl+Z 组合键将挂起正在执行的进程;用 bg 命令将使 进程恢复在后台执行,用 fg 命令将使进程恢复在前台执行。

19.2.1 创建shell程序

可以用任何编辑器编辑 shell 程序,只需将要执行的 shell 或 Linux 命令写入 shell 程序 即可。

例如每次进入系统时需要设置一些别名,可以用文本编辑程序如 vi 把这些别名放在一 个文件中,然后执行这个文件,而不需要每次进入系统时输入所有的别名。

为此目的而创建的示范文件 myalias 内容如下:

```
alias ll='ls –l'
```

```
alias dir='ls'
```

alias copy='cp'

在 Linux 下可用各种方法执行 myalias 脚本。

19.2.2 运行shell程序

在 Linux 下可以用如下几种办法运行已经写好的 shell 脚本。

可以把 shell 脚本的权限设置为可执行,这样就可以在 shell 提示符下直接执行。可以使用下列命令更改 shell 脚本的权限:
 chmod u+x filename 只有用户自己可以执行,其他人不能执行。
 chmod ug+x filename 只有用户自己以及同一个工作组的人可以执行,其他人不能执行。

chmod +x filename 所有人都可以执行。

这些命令打开了 myalias 的可执行许可。在执行之前,要确保另外一个条件,即 myalias 脚本必须在查找路径中。执行如下命令可获得查找路径:

#echo \$PATH

如果放置 myalias 脚本的目录不在当前查找路径中,那么就必须把这个目录名加到查找路径中。

现在,可以在命令行上像执行 Linux 命令一样执行 myalias 脚本:

#myalias

至于系统用哪一个 shell 来解释执行 shell 脚本,下面是几种约定的指定方式:

(1) 如果 shell 脚本的第一个非空白字符不是#,则使用 Bourne shell。

(2) 如果 shell 脚本的第一个非空白字符是#, 但不以#!开头时, 则使用 C shell。

(3) 如果 shell 脚本以#!开头,则#!后面所跟的字符串就是所使用的 shell 的绝对路径

名。如 Bourne shell 的路径名称为/bin/bash ,而 C shell 的路径名称则为/bin/csh。

例如,使用 Bourne again shell 可用以下方式:

#!/bin/bash

如使用 C shell,可用以下方式:

#!/bin/csh

执行 shell 脚本想要执行的 shell,其后跟随 shell 脚本的文件名作为命令行参数。
 例如,使用 pdksh 执行上面的 shell 脚本,如下所示:

#pdksh myalias

这里引用了新的 pdksh shell ,并把脚本作为参数传送来执行该文件。

pdksh shell 最初由 Eric Gisin 设计,是 ksh shell 的公共域版本,可以在/usr/bin 目录下 找到符号链接。在 Red Hat Linux 中, pdksh 称为 ksh。两个符号链接/usr/bin/pdksh 和/usr/bin/ksh 指向 pdksh。

在 pdksh 和 bash 下使用 "." 命令,或在 tcsh 下使用 source 命令。例如,在 pdksh
 和 bash 下执行上面的 shell 脚本,如下所示:

myalias

或在 tcsh 下执行上面的 shell 脚本,如下所示:

source myalias

执行 myalias 脚本后,就能从命令行上用 dir 来获得当前目录下的文件列表,并用-1 获得可显示各种属性的文件列表。当然,使用文件 myalias 中新的命令的最好方法是将它们放进 shell 的 login 或 profile 文件。对于 Red Hat Linux 用户,缺省的 shell 是 bash,所以在系

统中将这些命令放入/etc 目录中的 profile 文件可以使系统中的所有用户得以使用这些命令。

如果已经登录到 Red Hat Linux 系统想使用其他的 shell,可以使用 chsh 命令。用户将 被提示输入用户密码和新 shell 的名称和位置。这时新的 shell 将成为用户的缺省 shell。

19.3 shell编程规则

作为一种脚本语言, shell 有一套自己的编程规则。本节简单介绍一些 shell 的编程规则, 内容涉及 shell 的变量、运算、条件和循环语句以及函数等。

19.3.1 变量

Linux 的 shell 编程是一种成熟的编程语言。

shell 支持各种类型的变量。共有3种主要的变量类型:环境、内部和用户。

环境变量是系统环境的一部分,不必去定义它们,可以在 shell 脚本中使用它们,某些 变量还能在 shell 脚本中加以修改。

下面是几个环境变量及其含义:

- · \$HOME 用户自己的目录。
- · \$PATH 执行命令时所搜寻的目录。
- \$TZ 时区。
- · \$MAILCHECK 每隔多少秒检查是否有新的邮件。
- \$PS1 在 shell 命令行的提示符。
- · \$PS2 当命令尚未打完时, shell 要求再输入时的提示符。
- · \$MANPATH man 指令的搜寻路径。

内部变量是由系统提供的。与环境变量不同的是不能修改它们。这类变量在程序中用 来作出判定。

下面是几个内部变量及其含义:

- · \$# 存储 shell 程序中命令行参数的个数。
- · \$? 存储上一个执行命令的返回值。
- · \$0 shell 脚本的名称。
- · \$* 存储 shell 程序的所有参数。
- · "\$@" 存储所有命令行输入的参数,分别表示为 "\$1"、 "\$2"...。
- \$\$ 存储 shell 程序的进程号。
- · \$! 存储上一个后台执行命令的进程号。

变量是在编写 shell 过程时定义的,可以在 shell 脚本内任意使用和修改它们。

1. 给变量赋值

在 pdksh 和 bash 中给变量赋值的方法是一样的,即在变量名后跟着等号和变量值。例如,想要把 5 赋给变量 count,则使用如下的命令:

count=5

在 tcsh 中,可以使用如下的命令:

set count = 5

在 pdksh 和 bash 下必须确保在等号(=)的前后没有空格。 如要在变量中存放字符串,在 pdksh 和 bash 中可利用如下命令:

myname=Sanjiv

在 tcsh 中可利用如下命令:

set myname = Sanjiv

如果字符串没有嵌入的空格,可以利用上面的命令,如果字符串有嵌入的空格,在 pdksh 和 bash 中可以执行如下的赋值语句:

myname='Sanjiv Guha'

在 tcsh 中可利用如下命令:

set myname='Sanjiv Guha'

shell 编程和其他编程语言的主要不同是 shell 语言是一种不需要类型检查的解释语言, 所以在使用变量之前无须事先定义。也就是说,不必指定变量是数字类型还是字符串类型。 也可以使用同一个变量来存储字符串或整数。给字符串赋值的方法和给整数赋值的方法一 样。但是应该注意尽量不这样做。

2.访问变量值

可以通过在变量名前置以\$符来访问变量值。假如变量名为 var,利用\$var 就能访问这个变量。如要把 var 的值赋给变量 count,在 pdksh和 bash 中可以执行如下命令:

count=\$var

在 tcsh 中可利用如下命令:

set count=\$var

3. 位置参数

当编写一个带若干参数的 shell 脚本时,可从命令行或其他的 shell 脚本调用它。这些 选项是通过 Linux 作为位置参数来提供给 shell 脚本的。位置参数有由系统给出的专用名。 存放在变量中的第一个参数名为 1,在程序内可以利用\$1 来访问;存放在变量中的第二个 参数名为 2,在程序中可以利用\$2 来访问,依此类推。

例如有一个 shell 脚本 name,只带了一个参数(名字),在屏幕上显示这个名字。在 pdksh 和 bash 中,如果如下所示执行 name:

#.name

将得到如下输出:

Name not provided

但是,如果如下所示执行 name:

#.name Sanjiv

则得到如下的输出:

Your name is Sanjiv

下面的 shell 脚本 reverse 有两个变量。输出时,将两个变量的位置颠倒。

#program reverse , prints the command line parameters out in reverse order echo "\$2" "\$1"

在 shell 下执行此 shell 程序,如下所示:

reverse hello there

其输出结果,如下所示:

there hello

19.3.2 特殊字符

由于某些字符对 Linux shell 具有特殊含义,当字符串含有这类特殊字符时,必须利用转义字符(反斜杠)来指明这类特殊字符不应作为特殊字符来处理。

下面是这类特殊字符中的某些字符及其含义:

- · \$ 指出 shell 变量名的开始。
- · # 标记注释开始。
- · & 在后台执行进程。
- ? 匹配一个字符。
- · * 匹配 个或几个字符。
- · > 输出重定向操作符号。
- · < 输入重定向操作符。
- · ' 命令置换(反引号字符,许多键盘上在 Tab 键之上的键)。
- · >> 输出重定向操作符。
- [] 列出字符的范围。
- · [a z] 指 a~z 的全部字符。
- · [a, z] 指 a~z 字符。
- · .filename 执行 filename 文件。
- · 空格 两个字之间的间隔符。

有些特殊字符如单引号、双引号、反斜杠以及反引号在下面着重讨论。

1. 单引号

利用单引号(')把字符串括起来,可以阻止 shell 解析变量, shell 将忽视所有单引号中的特殊字符。例如,如果想把登录时的用户名也包括在 greeting 变量中,在 bash 和 pdksh环境下,应该使用下面的命令:

greeting="hello \$LOGNAME"

在 tcsh 环境下则使用下面的命令:

set greeting="hello \$LOGNAME"

如果是以 root 身份登录,将会把 hello root 存储在变量 greeting 中。但如果在上面使用 单引号,则单引号将会忽略\$符号的真正作用,而把字符串 hello \$LOGNAME 存储在 greeting 变量中。

2. 双引号

在字符串含有嵌入的空格时,应该用双引号括起来,让 shell 作为整体而不是作为几部 分来解释字符串。例如,把 abc def 的值赋给 shell 脚本中的 x 变量,在 pdksh 和 bash 中使 用如下命令:

x="abc def"

tcsh 中使用如下命令:

set x = "abc def"

双引号解析字符串内的所有变量。如下是一个有关 pdksh 和 bash 的示例:

var="test string" newvar="Value of var is \$var" echo \$newvar

下面是有关 tcsh 的相同示例:

set var="test string" set newvar="Value of var is \$var" echo \$newvar

执行上面 3 行 shell 脚本,可得到如下的结果:

Value of var is test string

3. 反斜杠

在某个字符前利用反斜杠 (\) 能够阻止 shell 把后面的字符解释为特殊字符。例如要 把\$test 的值赋给 ar 变量。如果用如下命令:

var=\$test

那么存放在 var 中的值将是一个空值。这是由于 shell 以变量 test 来解释\$test, 而 test 未赋给任何值, 所以 var 为空。应该用如下的命令把\$test 正确地存放住 var 中:

var=\\$test

在美元符(\$)之前的反斜杠用来告知 shell 同任何其他普通的字符一样来解释\$,因而 \$无任何特殊含义。

4. 反引号

利用反引号(')通知 shell 执行由反引号所定义的字符串。当需要把执行命令的结果存放在变量中时,就可以在 shell 脚本中利用反引号。例如要统计当前目录下一个文件中 test.txt 有几行并把结果存在 var 变量中时,可以用如下命令:

var='wc -l test.txt'

19.3.3 运算

1. expr 命令

使用 expr 命令可以处理数值运算。下面是数值运算符及其用法:

expr expression

expression 是由字符串以及运算符所组成的,每个字符串或运算符之间必须用空格隔 开。

如表 19.2 所示是运算符的种类及其功能。运算符的优先顺序以先后次序排列,可以利 用小括号来改变运算的优先次序。其运算结果输出到标准输出设备上。

运算符	作用
:	字符串比较
*	乘法
/	除法
%	取余数
+	加法
-	减法
<	小于
<=	小于等于
=	等于
!=	不等于
>=	大于等于
>	大于

表 19.2 运算符的种类及其功能

	(
运算符	作用
&	AND 运算
	OR 运算

字符串比较方式是以两字符串的第一个字母开始,以第二个字符串的最后一个字母结 束。如果相同则输出第二个字符串的字母个数,如果不同则返回0。

当 expression 中含有*、(或)等符号时,必须在其前面加上\,以免被 shell 解释成其他意义。例如: expr 2 * \(3 + 4 \)。

2. test 命令

在 bash 和 pdksh 环境中, test 命令用来测试条件表达式。其用法如下:

test expression 或者 [expression]

test 命令可以和多种系统运算符一起使用。这些运算符可以分为4类:整数运算符、字符串运算符、文件运算符和逻辑运算符。如表19.3、表19.4、表19.5和表19.7所示分别是 整数运算符、字符串运算符、文件运算符和逻辑运算符的使用和它们的返回值。

整数运算符	返回值
int1 –eq int2	如果 int1 和 int2 相等,则返回真
int1 –ge int2	如果 int1 大于等于 int2,则返回真
int1 -gt int2	如果 int1 大于 int2 , 则返回真
int1 -le int2	如果 int1 小于等于 int2,则返回真
int1 -lt int2	如果 int1 小于 int2 , 则返回真
int1 -ne int2	如果 int1 不等于 int2,则返回真

表 19.3 整数运算符的使用及其返回值

表 19.4 字符串运算符的使用其及返回值

字符串运算符	返回值
str1 = str2	如果 str1 和 str2 相同,则返回真
str1 != str2	如果 str1 和 str2 不相同,则返回真
str	如果 str 不为空,则返回真
-n str	如果 str 的长度大于零,则返回真
-z str	如果 str 的长度等于零,则返回真

表 19.5 文件运算符的使用及返回值

文件运算符	返回值
-d filename	如果 filename 为目录,则返回真
-f filename	如果 filename 为普通的文件,则返回真
-r filename	如果 filename 可读,则返回真

· / 生 士 丶

1	4金主	٦
C	纵衣)

文件运算符	返回值
-s filename	如果 filename 的长度大于零,则返回真
-w filename	如果 filename 可写,则返回真
-x filename	如果 filename 可执行,则返回真
	表 19.6 逻辑运算符的使用及返回值

逻辑运算符	返回值
! expr	如果 expr 为假,则返回真
expr1 -a expr2	如果 expr1 和 expr2 同时为真 , 则返回真
expr1 -o expr2	如果 expr1 或 expr2 有一个为真,则返回真

tcsh 中没有 test 命令,但它同样支持表达式。tcsh 支持的表达式形式基本上和 C 语言 一样。tcsh 表达式的运算符也分为整数运算符、字符串运算符、文件运算符和逻辑运算符 4 种。如表 19.7、表 19.8、表 19.9 和表 19.10 所示分别是整数运算符、字符串运算符、文件 运算符和逻辑运算符的使用和返回值。

整数运算符	返回值
int1 <= int2	如果 int1 小于等于 int2 , 则返回真
$int1 \ge int2$	如果 int1 大于等于 int2,则返回真
int1 < int2	如果 int1 小于 int2,则返回真
int1 > int2	如果 int1 大于 int2,则返回真
	表 19.8 字符串运算符的使用及返回值
字符串运算符	返回值
str1 == str2	如果 str1 和 str2 相同,则返回真
str1 != str2	如果 str1 和 str2 不相同,则返回真
	表 19.9 文件运算符的使用及返回值
文件运算符	返回值
-r file	如果 file 可读,则返回真
-w file	如果 file 可写,则返回真
-x file	如果 file 可执行,则返回真
-e file	如果 file 存在,则返回真
-o file	如果当前用户拥有 file,则返回真
-z file	如果 file 长度为零,则返回真

-d file

如果 file 为目录,则返回真

表 19.10 逻辑运算符的使用及其返回值

逻辑运算符	返回值
exp1 exp2	如果 exp1 为真或 exp2 为真,则返回真
exp1 && exp2	如果 exp1 和 exp2 同时为真,则返回真
! exp	如果 exp 为假,则返回真

19.3.4 条件语句

1. if 表达式

if 条件是可以嵌套的,也就是说, if 条件内可以包含另一个 if 条件。if 条件中的 elif 或 else 部分并非是必要的。如果在 if 语句和相继的 elif 语句中所指定的表达式可选的都不 为真时,则执行 else 部分。fi 用来指出 if 语句的结束,在嵌套 if 条件的情况下, fi 是很有 用的。在这种情况下应该让 fi 与 if 相匹配,以确保所有的 if 语句是正确的编码。bash 和 pdksh 的 if 表达式如下:

if [expression]

```
then commands
elif [expression2]
then
commands
else
commands
```

fi

elif 和 else 在 if 表达式中均为可选部分。elif 是 else if 的缩写。只有在 if 表达式和任何 在它之前的 elif 表达式都为假时,才执行 elif。fi 关键字表示 if 表达式的结束。

在 tcsh 中, if 表达式有两种形式。第一种形式如下所示:

if (expression1)

then commands

else if (expression2)

then commands

else commands

endif

tcsh 的第二种形式是第一种形式的简写,它只执行一个命令。如果表达式为真,则执行,如果表达式为假,则不做任何事。其用法如下:

if (expression) command

2. case 语句

case 语句是执行匹配指定变量值的语句。在大多数情况下,如果存在大量条件,就可 以用 case 语句来替代 if 语句。

```
pdksh和bash的case语句格式如下:
case string1 in
str1)
commands;;
str2)
commands;;
*)
commands;;
```

esac

可以对每个条件指定若干个值(例如 str1 和 str2 等),或用通配符指定值。最后的条件 应是星号(*),当其他条件都不满足时执行它。对所指定的条件中的每个条件,执行全部 关联的语句,直到双分号(;;)时终止。

tcsh 的 case 语句格式如下:

switch (string1)

case str1:

statements

breaksw

case str2:

statements breaksw

default :

statements

breaksw

endsw

可以对每个条件指定若干个值 (例如 str1 和 str2 等),或利用通配符指定值。最后的条件应是 default,当其他条件都不满足时执行它。对所指定的条件中的每个条件,执行全部 关联的语句,直到 breaksw 时终止。

19.3.5 循环语句

1. for 语句

bash 和 pdksh 中有两种 for 语句的表达式。 第一种形式是:

for var1 in list do

commands

done

在此形式时,对在 list 中的每一项, for 语句都执行一次。List 可以是包括几个单词的 由空格分隔开的变量,也可以是直接输入的几个值。每执行一次循环, var1 都被赋予 list

```
中的当前值,直到最后一个为止。
```

第二种形式是:

for var1

do

statements

done

使用这种形式时,对变量 var1 中的每一项, for 语句都执行一次。此时, shell 程序假 定变量 var1 中包含 shell 程序在命令行的所有位置参数。

一般情况下,此种方式也可以写成如下形式:

for var1 in "\$@" do statements done

在 tcsh 中, for 循环语句叫做 foreach。其形式如下:

foreach name (list)

commands

end

2. while 语句

当指定的条件为真时 while 语句可用来执行一系列命令。一旦所指定的条件判定为假时,循环就立即终止。如果所指定的条件一开始就判定为假,则不执行循环。利用 while 命令时应该注意,所指定的条件如果不判定为假,循环也就永不终止。

在 bash 和 pdksh 环境下,使用 while 语句的表达式形式如下:

while expression do statements

done

而在 tcsh 中, while 语句形式如下:

while (expression)

statements

end

3. until 语句

until 语句的作用和 while 语句基本一样,只是当给定的条件为假时,执行 until 语句。 until 语句在 bash 和 pdksh 中的写法如下:

until expression do

commands

done

until 语句不是很常用,因为 until 语句可以用 while 语句重写。tcsh 命令不支持 until 语句。

4. shift 命令

bash、pdksh 和 tcsh 都支持 shift 命令。shift 命令将存储在位置参数中的当前值左移一 个位置。例如当前的位置参数如下:

\$1=-r \$2=file1 \$3=file2

执行 shift 命令,如下所示:

shift

位置参数将会变,如下所示:

\$1=file1 \$2=file2

也可以指定 shift 命令每次移动的位置个数。下面的示例将位置参数移动两个位置:

shift 2

编写一个 shell 脚本并使用户能够在程序中传送不同的选项时, shift 命令是很有用的。 这些参数随所指定的选项而定,可以指不同的含义,或者可能根本不存在。

下面是一个应用 shift 命令的示例。此程序有两个命令行参数,一个代表输入文件,另 一个代表输出文件。程序读取输入文件,将其中的内容转换成大写,并将结果存储在输出 文件中,如下所示:

```
while ["$1"] do

if ["$1"="-i"]

then infile="$2"

shift 2

elif ["$1"="-o"]

then

outfile="$2"

shift 2

else

echo "Program $0 does not recognize option $1"

fi

done

tr a-z A-Z <$infile>$outfile

5. repeat 语句

repeat 语句使一个单一的语句执行指定的次数。repeat 语句形式如下:
```

repeat count command

下面是 repeat 语句的一个示例。它读取命令行后的一串数字,并根据数字在屏幕上分 行输出句号。

```
foreach num ($*)
repeat $num echo -n "."
echo ""
```

end

任何 repeat 语句都可以用 while 或 for 语句重写, repeat 语句只是更加方便而已。

6. select 语句

select 语句自动生成一个简单的文字菜单。其用法如下:

select item [in list_of_items] do

commands

done

其中方括号中是 select 语句的可选部分。

当 select 语句执行时 ,pdksh 为在 list_of_items 中的每一项创建一个标有数字的菜单项。 list_of_items 可以是包含几个条目的变量,例如, choice1 choice2, 或者是直接在命令 中输入的选择项,例如,

select item in choice1 choice2 choice3

如果没有 list_of_items, select 语句则使用命令行的位置参数,就像 for 表达式一样。 一旦选择了菜单项中的一个, select 语句将选中的菜单项的数字值存储在变量 item 中。可 以利用 do 语句来执行选中的菜单项要执行的命令。

下面是 select 语句的一个示例。

19.3.6 函数

shell 语言可以定义自己的函数。使用函数可以使程序具有更好的可读性。下面是如何 在 bash 和 pdksh 中创建一个函数的命令形式:

```
fname () {
```

shell commands

```
}
```

在 pdksh 中也可以使用如下的形式:

function fname {

```
shellcommands
```

}

使用函数时,只须输入以下的命令:

```
fname [parm1 parm2 parm3 ...]
```

可以传递任何数目的参数给一个函数。函数将会把这些参数视为位置参数。请看下面的示例,此例子包括4个函数:upper()、lower()、print()和 usage_error(),它们的任务分别是:将文件转换成大写字母、将文件转换成小写字母、打印文件内容和显示出错信息。 upper()、lower()和 print()都可以有任意数目的参数。如果将此示例命名为 convert,可以在 shell 提示符下如下使用该程序:

#convert -u file1 file2 file3.

```
convert 程序内容如下:
upper () {
     shift
     for i do
           tr a-z A-Z <$1>$1.out
           rm $1
           mv $1.out $1
           shift
     done:
}
lower () {
     shift
     for i do
           tr A-Z a-z <$1>$1.out
           rm $1
           mv $1.out $1
           shift
           done;
}
print () {
     shift
     for i do
           lpr $1
           shift
           done;
}
usage_error () {
     echo "$1 syntax is $1 <option> <input files>"
     echo ""
```

```
echo "where option is one of the following"
     echo "p ---to print frame files"
     echo "u ---to save as uppercase"
     echo "l ---to save as lowercase";
case $1 in
     p|-p) print $@;;
     u|-u) upper $@;;
     l|-l) lower $@;;
      *) usage_error $0;;
```

esac

}

颕 19.4 기

1. 编写一个叫做 compare1 的 shell 程序,用于比较 string1 和 string2 两个字符串。

2. 编写一个叫做 compare2 的 shell 程序,比较 number1 和 number2 两个数。

3. 写 shell 脚本,分别使用 while 和 until 语句累加 5 个最小的偶数。

4. 使用 case 语句编写 shell 脚本,该脚本提示用户输入月份并打印月份名。

5. 参照第 19.3.6 节中的示例程序,编写 shell 脚本,把一个输入文件中的字母改成大 写,存在另一个输出文件中。

第 20 章 Perl 编 程

本章主要介绍在 Linux 环境下如何进行 Perl 编程。内容包括 Perl 编程简介、Perl 程序 简单示例以及 Perl 编程规则等。

通过学习本章,读者可以了解Linux环境下的Perl编程的基本概念和知识。

20.1 Perl编程简介

Perl (Practical Extraction and Report Language) 是由 Larry Wall 于 80 年代开发的,它的设计目标就是"使容易的事情变得更容易,使困难的事情能够实现"。Perl 有一个座右铭:完成一个任务有多种方法(There is more than one way to do it)。因为该座右铭经常被提到,所以缩写为 TIMTOWTDI。

Perl 是一种解释性的语言,专门为搜索纯文本文件而做了优化。它也可以十分方便地 完成很多系统管理任务。它集成了 C、sed、awk 和 shell 语言的优点,包含了健壮的 awk, sed 和 shell 脚本的模块。

Perl 可以运行于 Linux、Unix、MVS、VMS、MS-DOS、Macintosh、OS/2、Amiga 以及其他一些操作系统上。大多数 Perl 程序无需修改就可以在其他系统上运行。对于 Unix 系统或 Linux 系统,可以从 http://www.perl.com/下载 Perl 源码并进行编译。

很多用户把精力放在程序能被正确地执行上,而 Perl 使用户无须考虑内存的占用或变 量的输入等问题,完成所希望的工作。系统程序员可以用 Perl 结合系统命令来处理数据和 过程,并且可以使用 Perl 的格式匹配函数进行系统信息的搜寻和总结。如果对其他的编程 语言比较熟悉,那么编写 Perl 程序是一件比较容易的事。

Perl 编译器随 Red Hat Linux 发布,在 Red Hat Linux 7.0 中带有 Perl 5.6.0 编译器。

20.2 Perl简单示例

一个有用的 Perl 程序可以很短。例如,使用下面的一条命令可以更换大量文件中的一 些相同内容:

perl -e 's/gopher/World Wide Web/gi' -p -i .bak *.html

下面是一个基本的 perl 程序:

#!/usr/local/bin/perl

#

Program to do the obvious

#

print "Hello world.\n";

Print a message

这就是一个 Perl 程序的全部内容。输入这一程序,把它存放到一个名为 hello.pl 的文件中,对这一文件执行如下命令:

chmod +x hello.pl

然后就可以运行这个程序了。下面对这个程序进行一些解释。

· #! 不是 Perl 程序的代码,它用来向操作系统指示到哪里去寻找 Perl 解释器,这 是 shell 脚本的标准过程。如果系统上/usr/bin/perl 不是 Perl 所在的正确位置,那么 可以通过在命令行输入下面的命令查找它所在的位置:

which perl

· print "Hello world.\n"用来显示引号中的文本,其中符号\n用来作为换行符。

Perl 的语句是以分号为结束符的。在屏幕上一个 Perl 语句可以占用很多行。当然,也可以把 Perl 语句写在一行中。Perl 对空格不敏感。

 # 表示一行中剩下的部分都为注释。就是说,从这个#到下一行的开始的这一部 分内容将会被 Perl 编译器忽略。当然,#在引用中或在一个正则表达式中充当定界 符时除外。

20.3 Perl编程规则

Perl 语言有特定的编程规则,本节就 Perl 的变量、Perl 语句和 Perl 进行文本操作的规则进行介绍。

20.3.1 Perl变量类型

Perl 是一种弱类型语言。也就是说,在编程中并不需要指定在特定的变量中存储的是 什么类型的数据。例如,C 语言要求必须声明变量为整数、字符、指针或其他类型。Perl 中的变量可以是它们所需要的任何类型,在需要时也可以方便地改变它们的类型。

在 Perl 中存在 3 种变量类型——标量、数组和联合数组。为了使每一个数据类型变得 更直观,每一种变量类型都对应一个符号。

1.标量

Perl 中最基本的变量类型是标量。标量既可以是数字,也可以是字符串,而且两者是可以互换的。具体是数字还是字符串由上下文决定。如果将一个数字当成字符串使用,那 它就是字符串。如果将一个字符串当成数字使用,如果有意义就会被转换为数字;如果没 有意义很有可能被置为 0。例如,字符串 15abc 会被转换为 15,而 abc 则被转换为 0。在 Perl 中,变量名大小写敏感。

标量变量是由\$符号来标志的,如:

\$priority = 9;

把9赋予标量变量\$priority,也可以将字符串赋予该变量:

\$priority = 'high';

Perl 中有 3 种类型的引用。双引号(") 括起来的字符串中的任何标量和特殊意义的 字符都将被 Perl 解释。如果不想让 Perl 解释字符串中的任何标量和特殊意义的字符,应该 将字符串用单括号括起来。这时,Perl 不解释其中的任何字符,除了\\ 和\ 。最后,可以 用'将命令括起来。这样,其中的命令可以正常运行,并能得到命令的返回值。请看下面的 示例:

#!/usr/bin/perl
\$folks="100";
print "\\$folks = \$folks \n";
print \\$folks = \$folks \n';
print "\n\n BEEP! \a \LSOME BLANK \ELINES HERE \n\n";
\$date = 'date +%D';
print "Today is [\$date] \n";
chop \$date;
print "Date after chopping off carriage return: [".\$date."]\n";

程序输出结果如下:

\$folks = 100
\$folks = \$folks \n
BEEP! some blank LINES HERE
Today is [03/29/96]
Date after chopping off carriage return: [03/29/96]

第3行显示\$folks的值。\$之前必须使用换码符\以便Perl显示字符串\$folks而不是\$folks的值 100。

第 4 行使用的是单引号,结果 Perl 不解释其中的任何内容,只是原封不动地将字符串 显示出来。

第 6 行使用的是',则 date +%D 命令的执行结果存储在标量\$date 中。

下面是一些上例中使用的特殊变量和一些其他特殊变量的含义:

- ・ \n 换行。
- ・ \r 回车。
- \t 制表符。
- · ∖a 蜂鸣声。
- \b Backspace.
- · \L \E 将\L 和\ E 之间的字符转换成小写。
- · \1 将其后的字符转换成小写。
- · \U \E 将\U 和\E 之间的字符转换成大写。

- · \u 将其后的字符转换成大写。
- · \cC 插入控制字符 C。
- · \x## 十六进制数##。
- · \0000 八进制数 000。
- 、、、反斜杠。
- · \ 按原样输出下一个字符。例如,\\$ 输出\$。

2.数组

Perl 数组也叫做列表,是由一系列标量组成的。在 Perl 中使用@来代表数组。如@fish。 数组为一组和索引数(从0开始)相对应的数值。数组中的每一个元素为标量。因为标量 是由\$表示的,所以在数组中的单个元素也可以用\$来表示。

例如,\$fish[2]表示数组@fish 中的第 3 个元素。这样好像会把一些项漏掉,其实它们 是完全一致的。

数组变量以@开头。请看以下的赋值语句:

@food = ("apples", "pears", "eels");

@music = ("whistle", "flute");

数组的下标从0开始,可以使用方括号引用数组的下标,如下所示:

\$food[2]

上例返回 peers。因为 peers 是一个标量,所以@已经变成了\$。 在 Perl 中,数组有多种赋值方法,如下所示:

@moremusic = ("organ", @music, "harp"); @moremusic = ("organ", "whistle", "flute", "harp");

还有一种方法可以将新的元素增加到数组中,如下所示:

push(@food , "eggs");

把 eggs 增加到数组@food 的末端。要往数组中增加多个元素,可以使用下面的语句:

push(@food , "eggs" , "lard"); push(@food , ("eggs" , "lard")); push(@food , @morefood);

push 返回数组的新长度。 pop 用来将数组的最后一个元素删除,并返回该元素。如下所示:

@food = ("apples", "pears", "eels");
\$grub = pop(@food);

此时\$grub = "eels"

3.联合数组

一般的数组允许通过数字下标存取其中的元素。例如数组@food 的第一个元素是

\$food[0],第二个元素是\$food[1],以此类推。但 Perl 允许创建联合数组,这样可以通过字符串存取数组。联合数组还有一个名字叫散列。

一个联合数组中每个下标索引对应两个条目。第一个条目叫做关键字,第二个条目叫 做数值。这样,就可以通过关键字来读取数值。联合数组中的每一个元素为标量,所以在 数组中的单个元素也可以用\$来表示。例如,\$employee{name}给出联合数组中的一个数值。 在对联合数组进行操作时,keys和 values这两个函数是很有用的。keys函数返回一个包含 联合数组中所有名称的数组,而 values函数返回一个包含联合数组中所有数值的数组。

联合数组名以百分号(%)开头,通过花括号({})引用条目。例如:

%ages = ("Michael Caine", 39, "Dirty Den", 34, "Angie", 27, "Willy", "21 in dog years", "The Queen Mother", 108);

可以通过下面的方法读取数组的值:

\$ages{"Michael Caine"};# Returns 39
\$ages{"Dirty Den"};# Returns 34
\$ages{"Angie"};# Returns 27
\$ages{"Willy"};# Returns "21 in dog years"
\$ages{"The Queen Mother"};# Returns 108

4.特殊变量

在 Perl 中,存在一大批特殊变量。例如, \$_、 \$!和\$]。对于这些变量, perl 都可以识别。 逐步熟悉 perl 后,会发现这些变量十分有用。

\$_是这些特殊变量中最有用处的一个。\$_是 Perl 中的"缺省"变量。当没有声明参数的时候,将使用变量\$_。例如,下面的两个循环是等价的:

```
for $cow (@cattle) {
    print " $cow says moo.\n ";
}
for (@cattle) {
    print " $_ says moo.\n ";
}
```

20.3.2 Perl 运算符

Perl 中的数字是以浮点形式存储的。如表 20.1 所示是有关的数字运算符的使用示例及 其结果。

使用示例	结果说明
\$a = 1+2;	1 加 2 , 结果存储在\$a 中
\$a = 3-4;	3 减 4 , 结果存储在\$a 中
a = 5*6;	5 乘 6 , 结果存储在\$a 中
\$a = 7/8;	7 除以 8, 结果存储在\$a 中
\$a = 9**10;	9 的 10 次方,结果存储在\$a 中
\$a = 5%2;	取 5 除 2 的余数,结果存储在\$a 中
++\$a;	\$a 加 1, 然后赋予\$a
\$a++;	先将\$a 返回,然后\$a 加 1
\$a;	\$a 减 1 , 然后赋予\$a
\$a;	先将\$a 返回, 然后\$a 减 1
\$a = \$b;	\$b 赋予\$a
\$a += \$b;	\$b 加\$a,然后赋予\$a
\$a -= \$b;	\$a 减\$b , 然后赋予\$a
\$a .= \$b;	把\$b 连接到\$a 的后面,然后赋予\$a

表 20.1 数字运算符使用示例及结果说明

如表 20.2 列出逻辑运算符的使用示例及其结果。

表 20.2 逻辑运算符使用示例及结果说明

使用示例	结果说明
$r = x \parallel y$	\$r = \$x 或\$ y
r = x &	r = x = y
\$r = ! \$x	\$r = 非\$ x

如表 20.3 所示是字符标量运算符的使用示例及其结果。

表 20.3 字符标量运算使用示例及结果说明

使用示例	结果说明
\$a = \$b.\$c;	将\$b 和\$c 连接,然后赋予\$a
\$a = \$bx\$c;	\$b 重复\$c 次,然后赋予\$a

如表 20.3 所示是比较运算符的使用示例及其结果。

表 20.4 比较运算符使用示例及结果说明

使用示例	结果说明
\$x == \$y	如果\$x 和\$y 相等,则返回真
\$x != \$y	如果\$x 和\$y 不相等,则返回真
\$x < \$y	如果\$x 比\$y 小,则返回真
\$x <= \$y	如果\$x 小于等于\$y , 则返回真

(续表)

使用示例	结果说明
x > y	如果\$x 比\$y 大,则返回真
$x \ge y$	如果\$x 大于等于\$y,则返回真
\$x eq \$y	如果字符串\$x 和字符串\$y 相同,则返回真
\$x ne \$y	如果字符串\$x 和字符串\$y 不相同 , 则返回真
\$x lt \$y	如果字符串\$x 比字符串\$y 小 , 则返回真
\$x le \$y	如果字符串\$x 小于等于字符串\$y , 则返回真
\$x gt \$y	如果字符串\$x 比字符串\$y 大,则返回真
\$x ge \$y	如果字符串\$x 大于等于字符串\$y , 则返回真
\$x cmp \$y	如果\$x 大于\$y , 则返回 1 , 如果\$x 等于\$y , 则返回 0 , 如果\$x 小于\$y , 则返回-1
\$w ? \$x : \$y	如果\$w 为真,则返回\$x;如果\$w 为假,则返回\$y
(\$x\$y)	返回从\$x 到\$y 之间的值

20.3.3 条件语句

Perl 提供了两种条件语句, if 语句和 unless 语句, 它们实现相反的功能。 if 语句是满足 一定条件的情况下执行一段程序,并且控制着程序的逻辑流程; unless 是在不满足特定条 件的情况下起作用。 if 语句的语法如下:

```
if (condition) {
  statements
  }
  elsif (condition) {
    statements
  }
  else {
    statements
  }
}
```

condition 可以是任何语句或比较。在 Perl 中除了数字 0、"0" (包含数字 0 的字符串) " "(空字符串)和没有定义的值,其他所有的量都是真。注意:甚至字符串"00"也为真, 因为它不属于上述 4 种情况之一。

如果该语句返回的值为真,那么将执行 statement 所表示的语句。 if 结构中也可以使用嵌套结构,如下所示:

```
if (!$a) # The ! is the not operator
{
    print "The string is empty\n";
}
elsif (length($a) == 1)# If above fails , try this
{
    print "The string has one character\n";
```
```
}
elsif (length($a) == 2)# If that fails , try this
{
print "The string has two characters\n";
}
else# Now , everything has failed
{
print "The string has lots of characters\n";
}
```

unless 和 if 的工作方式近似,只是功能相反。unless 在表达式为假的时候执行语句和程序块。

20.3.4 循环语句

Perl 有 4 种循环结构: for、foreach、while 和 until。

1. for 语句

for 语句在满足条件的情况下执行一条或一组语句 ,Perl 中的 for 结构和 C 语言中的 for 结构基本一样 ,如下所示 :

for (start condition; end condition; increment function) {
 statement(s)

}

在循环的开始,首先设置的是起始条件。当循环每执行一次时,增量函数也将执行一次,直至达到结束条件为止。下面是一个 for 循环的示例,用来显示从 0 到 9 的数字:

```
#!/usr/local/bin/perl
for ($i = 0; $i < 10; ++$i) # Start with $i = 1
# Do it while $i < 10
# Increment $i before repeating
{
    print "$i\n";
}
2...foreach 语句</pre>
```

foreach 语句用来对集合如列表或数组中的每个元素执行一条或一组语句,如下所示:

```
foreach $name (@names) {
    print " $name\n ";
```

}

循环变量(在示例中为\$name)不仅仅被赋予了数组变量中元素的值,而且它已经是 元素的别名。这意味着,如果改变循环变量的值,数组也会被修改。

请看下面的示例:

```
foreach $morsel (@food)# Visit each item in turn
# and call it $morsel
{
    print "$morsel\n";# Print the item
    print "Yum yum\n";# That was nice
}
```

每次要执行的命令用花括号括出。第一次执行时\$morsel 被赋予数组@food 的第一个元素的值,第二次执行时\$morsel 被赋予数组@food 的第二个元素的值,以此类推直到数组的最后一个元素。

3. while 语句和 until 语句

当某个特定的条件为 true 时, while 将执行一组语句, 语法如下:

```
while (conditon) {
```

statements;

}

condition 为任何可以返回 true 或 false 值的任何合法表达式。

until 与 while 语句正好相反。当某个特定的条件为 false 时, until 将执行一组语句, 直 到条件变成 true 时才终止对循环体的执行。

下面是一个 while 和 until 循环的示例。它从键盘读取输入直到得到正确的口令为止。

#!/usr/local/bin/perl

print "Password? ";# Ask for input

\$a = <STDIN>;# Get input

chop \$a;# Remove the newline at end

while (\$a ne "fred")# While input is wrong...

```
{
```

print "sorry. Again? ";# Ask again \$a = <STDIN>;# Get input again chop \$a;# Chop off newline again

}

当输入和口令不相等时,执行 while 循环。 在执行体的末尾处也可以使用 while 和 until,这时需要用 do 语句,如下所示:

#!/usr/local/bin/perl

do {

"Password? ";# Ask for input \$a = <STDIN>;# Get input chop \$a;# Chop off newline

}

while (\$a ne "fred")# Redo while wrong input

20.3.5 文件操作

通过下面的程序可以了解文件句柄的基本用法。程序的执行结果和 Linux 的 cat 命令一样:

```
#!/usr/local/bin/perl
$file = '/etc/passwd';
open(INFO , $file);
@lines = <INFO>;
close(INFO);
print @lines;
```

open 函数打开一个文件以供读取,其中第一个参数是文件句柄。文件句柄用来供 Perl 在以后指向该文件。第二个参数指向该文件的文件名。close 函数关闭该文件。

open 函数还可以用来打开文件以供写入和追加,只须分别在文件名之前加上>和>>, 格式如下。

- · 为写打开 open(INFO, \$file);
 - open(INFO, "<\$file");
- · 为读打开 open(INFO, ">\$file");
- 为追加打开 open(INFO, ">>\$file");

20.3.6 文本操作

Perl 最强的实力在于文本操作,这是通过利用正则表达式(regex)来完成的。正则表达式使得复杂的模式匹配和替换能够高效并容易地完成。

例如,下面的一行代码将把在文本行中出现的字符串 Bob 或 Mary 替换为 Fred。如表 20.5 所示是对代码进行的解释。

\$string= ~ s/bob mary/fred/gi;

代码	说明
\$string = ~	对 \$ string 变量中的文本执行这种模式匹配
S	替换
/	开始对文本进行匹配
bob mary	匹配文本 bob 或 mary。要注意的是:这是查找文本 mary 而不是单词 mary。也
	就是说,它也匹配 maryland 中的文本 mary
/	结束对文本的匹配,开始替换文本
fred	用文本 fred 替换所有匹配的内容
/	结束对文本的替换
g	全局进行替换。也就是说,不管在字符串中的哪个位置,只要有匹配的文本(无
	论有多少个),都将进行替换

表 20.5 代码说明

(续表)

代码	说明
i	匹配文本时忽略字母的大小写。也就是说 , bob 将匹配 bob、Bob 或 boB 等
;	表示代码行的结束

虽然用一个字符串替换另一个字符串似乎是一件相当简单的任务,但是当用其他语言, 例如 C 编写完成这一任务的代码时却相当繁琐。

下面再举个例子:

```
s = One if by land and two if by sea';
```

```
if (s = /if by la/) {print "YES"}
```

else {print"NO"}

结果将会显示 YES ,因为 if by la 在字符串\$s 中。 下面列出了一些在正则表达式中具有特殊意义的字符及其意义:

- · . 除了换行符(\n)的任何字符。
- · ^ 一行和一个字符串的开始。
- · \$ 一行和一个字符串的结束。
- · * 其前一个字符重复零次或多次。
- · + 其前一个字符重复一次或多次。
- · ? 其前一个字符重复零次或一次。

例如下面这个语句:

```
if ($x =~ /l.mp/) {print "YES"}
```

对于\$x =" lamp "、" lump "和" slumped "将显示 YES ,但对于\$x =" lmp "或" lessamperes " 将不会显示 YES 。

再看下面的例子:

- · /fr.*nd/ 匹配 frnd、 friend 和 frontandback。
- · /fr.+nd/ 匹配 frond、friend 和 frontandback,但不匹配 frnd。
- · /10*1/ 匹配 11、101、1001 和 100000001。
- /b(an)*a/ 匹配 ba、 bana、 banana 和 bananana。
- · /flo?at/ 匹配 flat 和 float,但不匹配 flooat。

方括号用来匹配其中的任何字符。方括号中的-符号用来表明在什么之间^符号表明非 的意思。

- · [0123456789] 匹配任何单个的数字。
- · [0-9] 匹配任何单个的数字。
- · [a-z] 匹配任何由小写字母组成的单词。
- · [^0-9] 匹配任何非数字的字符。

反斜杠还是用于转义。如果想匹配+、?、.、*、^、\$、(、)、[、]、{、}、\和/这些 字符,则其前面必须用反斜杠(\)。如下所示:

- · /10.2/ 匹配 10Q2、1052 和 10.2。
- · /10\.2/ 匹配 10.2,但不和 10Q2 或 1052 匹配。
- ・ ヘ*+/ 匹配一个或多个星号。
- ・ /A:\\DIR/ 匹配 A:\DIR。
- ・ //usr\/bin/ 匹配/usr/bin。

下面还有一些特殊意义的字符:

- ・ \n 换行。
- \t 制表符。
- · \w 任何字母和数字和[a-zA-Z0-9_]一样。
- · \W 任何非字母和数字和[^a-zA-Z0-9_]一样。
- · \d 任何数字和[0-9]一样。
- · \D 任何非数字和[^0-9]一样。
- · \s 任何空白字符,空格、tab 和换行等。
- · \S 任何非空白字符。
- · \b 单词边界,只对[]以外有效。
- · \B 非单词边界。

```
20.3.7 过程
```

1. 过程定义

Perl 语言可以定义自己的过程。一个过程的格式定义如下:

sub mysubroutine{

commands;

}

下面的几种方法都可以调用这个过程:

- &mysubroutine;# Call the subroutine
- &mysubroutine(\$_);# Call it with a parameter
- &mysubroutine(1+2 , \$_);# Call it with two parameters
- 2.参数

调用一个子过程时,所有的参数都传送到了数组@_中。下面子过程的示例显示出所有 的参数:

```
sub printargs{
  print "@_\n";
}
```

调用结果如下:

&printargs("perly", "king"); 结果将显示 perly king。 &printargs("frog", "and", "toad"); 结果将显示 frog and toad。

3.返回值

下面的示例返回两个输入参数的最大值:

```
sub maximum{
if ($_[0] > $_[1]){
$_[0];
}else{$_[1];
}}
调用$biggest = &maximum(37,24);
$biggest 的结果为 37。
```

20.3.8 命令行选项

Perl 中有各种各样的命令行选项,用它们可以灵巧地控制 Perl 的行为。这些命令行选项可以出现在命令中,也可以出现在 Perl 程序的#!行中。

下面是可用的命令行选项。

- · -a 当和-n 或-p 选项一同使用时,打开自动分割模式。这说明输入的每一行将自动地被分割为@F数组。
- · -c 让 Perl 对指定的 Perl 程序进行语法检验,且并不执行程序。这是很有益的, 它输出的错误信息是详尽的、可读的,并且告诉用户到哪里寻找错误。
- · -d 在 Perl 调试器下运行脚本。
- · -d: foo 在调试的控制下运行脚本或跟踪模块 Devel::foo。
- · -D flags 设置调试标志。
- · -e commandline 说明其后为一个 Perl 程序。这使得用户可以在命令行直接输入 Perl 代码,而不用运行一个文件中的代码。
- · -F pattern 当-a 有效时,声明要分割的模式。缺省为"",-F 选项使用户可以将 其设置为任何字符。
- · -h perl -h 将列出所有可用的命令行选项。
- · -i [extension] 声明在原文件中进行编辑。如果提供了 extension (后缀),原始文 件将被备份为带有这一后缀的文件。否则,原文件将被覆盖。
- · -I directory 由-I 声明的目录是模块(@INC)预先要搜索的路径。
- · -l [octnum] 允许自动的行结束进程。这说明输入的行尾符将被自动删除并将其放回输出。如果没有声明可选的八进制数,将使用新行符。
- · -m [-] module 或-M [-] module 在运行脚本之前调入指定的模块。
- · -n 当命令行中提供每个文件时循环运行脚本,不打印输出。
- · -p 功能和-n 一样,只是将每行都打印出来。
- · -P 在脚本被 perl 编译之间先进行 C 的预编译。

- · -s 做命令行参数解析并将参数放入到 perl 脚本中相应的变量中。
- · -S 使用 PATH 环境变量来搜索脚本。
- · -T 允许用户的输入检查。
- · -u 告诉 perl 在编译完脚本之后产生转储。
- · -U 允许在 Perl 程序中使用不安全的内容。
- · -v 打印 Perl 的版本号。
- · -V 打印 Perl 的主要配置参数值和@ INC 的当前值。
- · -V:显示指定配置变量的值。
- · -w 告诉 Perl 对程序中潜在的问题显示警告信息。
- · -x directory 告诉 Perl 脚本中嵌入了一些很大的内容,如 email 信息。Perl 将把在 以#!开头并包含'perl'的一行前的所有内容和在_END_之后的所有内容丢弃。如 果声明了目录, Perl 将在执行脚本之前改变到这个目录。

20.4 习题

- 1. 编写脚本,将文本文件中的字母变成大写,结果存入另一个文件。
- 2. 编写脚本,使用联合数组和循环语句实现目录文件的输出。
- 3. 编写脚本,删除目录中的 core 文件。

第 21 章 tcl/tk 编程

本章主要介绍如何在 Linux 环境下进行 tcl/tk 编程。内容包括 tcl/tk 简介、tcl 基础、tcl 编程规则以及 tk 入门等。

21.1 tcl/tk编程简介

tcl 脚本语言和 tk 工具箱是为 X Window 系统创建图形用户界面的编程环境 利用 tcl/tk 构造用户界面的速度要比利用传统的 X Window 编程方法快得多。

tcl 最初设计用来为交互式工具提供一种可复用的命令语言,但它的实际发展却远非如此,并且在众多的软件产品中被广泛应用。

利用 tcl 脚本语言几乎完全可以编写复杂的图形应用程序,因而避开了利用 C 语言编 写界面时所遇到的界面编程的许多复杂性。tcl/tk 也是一个用来设置查看的工具箱,它可以 利用 tcl 语法来创建按钮、滚动杆、对话框以及窗口等 GUI 组件。

tcl 和 tk 非常容易学习和使用。本章将介绍 tcl/tk 的基本知识和编程规则。

tcl 中的可执行程序 tcl、tclsh、wish 和 tclhelp 安装在 Red Hat Linux 7.0 的/usr/bin 目录中。本章以 tclsh 为例介绍 tcl/tk 的编程。

21.2 tcl编程基础

和前面介绍的 shell 和 Perl 等语言类似, tcl 也是一种解释性的语言。tcl 程序是以脚本的形式存在的。在某一点上, tcl 和 shell 的行为很相似, shell 可以在提示符下执行命令, 也可以将命令写成脚本一起执行。同样, tcl 也可以在自己的提示符下运行单个命令, 或者执行脚本。但是, 通常将 tcl 命令写成脚本执行, 因为其交互能力很有限。

21.2.1 交互使用方式

在 Linux bash 的提示符下输入 tclsh,就可以交互使用 tcl,这时将出现如下提示符:

%

在这一提示符后输入如下命令:

% puts " hello world ";

结果为在屏幕上输出 hello world。

这个命令使用的是 tcl 命令 puts(putstring)。再如:

% set x 10; % set y 10; % puts [expr x+y] ;

结果为在屏幕上输出 20。

21.2.2 非交互使用方式

在通常情况下, tclsh - 般是以非交互的方式来使用的。这就是说, 它们在 Linux 的提 示符(\$)下被调用并执行脚本, 例如:

#tclsh myprog.tcl

或者从一个脚本内来调用它们,这个脚本的第一行通常与如下内容类似:

#!/usr/bin/tclsh

在通常情况下,对脚本的每次安装都必须修改第1行,因为 tclsh 可能位于不同的位置。 为了避免在每次安装时都必须对脚本进行编辑, tclsh 的手册页推荐利用如下3行代码作为 所有 tcl/tk 脚本的头3行:

#!/bin/sh
#the next line restarts using tclsh \
exec tclsh "\$0" "\$@"

这意味着,只需在自己的路径中具有 wish 就可以使用脚本。利用这种方法产生的结果可能会根据系统上 sh 版本的不同而不同。

tcl 非交互使用方式允许把多个命令组合在一起,并且只要输入脚本的名字就可以执行 脚本中的所有命令,这种方式可以加速对大型程序的开发和调试。

21.3 tcl编程规则

本节将介绍 tcl 的语法及其在脚本中的使用方式。内容包括 tcl 数据类型, tcl 变量, tcl 对字符串和数字的操作、tcl 引用和置换、tcl 条件和循环语句、tcl 对文件的操作及使用过 程等内容。

21.3.1 命令

tcl 语句的使用格式如下:

command arguments

这里的 command 是 tcl 所要执行的命令; arguments 是提供给此命令的参数。整个行称为一个命令。

命令之间通过换行或者分号来分隔。如果在一行上只有一个命令,那么分号可以省略。 有时需要在表达式中使用另一个表达式的值。使用方括号来实现,如下所示: puts [expr 1000/4];

21.3.2 注释

tcl 的注释与 shell 和 perl 的用法一样,注释行以#号开头,如下所示:

#this is a comment

注释必须是独立的行,不能在一个命令行后使用注释,除非这个命令行以分号结尾, 因为 tcl 解析器总是认为一条命令应该以换行或分号结束。因此,用分号结束所有的命令通 常是一种很好的做法。

21.3.3 变量

同 shell 和 Perl 一样, tcl 是一种弱类型语言,变量在使用之前不需要声明类型,并且 在同一程序中,一个变量可以在不同的时刻分别当作数值、字符或字符串使用。

tcl 可以定义两种类型的变量:标量和数组。

1.标量

可利用 set 命令创建标量并对它赋值,例如:

set physics 1;

这一命令将创建变量 physics 并把它的值赋为 1。

同样,如果想在程序的另一处把变量 physics 设置为一个字符串,可以直接赋值,如下 所示:

set physics "My favorite class";

此时变量 physics 具有的值为 My favorite class,命令中的双引号用来通知 tcl 变量的值 由引号中包括空格在内的所有字符组成。

在使用变量时应使用\$符引用。例如要输出 physics 的值,可利用 puts 命令,如下所示:

puts \$physics;

2.数组定义

tcl 数组和前面介绍过的关联数组类似,下标和数组内容一起构成了一对值。对数组的 赋值可以像下面这样进行:

set dept(0) physics; set dept(1) maths;

这两个命令将创建一个数组变量 dept,并把它的两个元素分别赋值为 physics 和 maths。 由于是关联数组,所以数组的下标可以使用非连续的内容。例如下面命令在数组 dept 中创建了3个元素。

set dept { 100 } computer;

set dept { 2 } chemistry; set dept { 87 } art;

还可以在下标中使用字符串等内容,如下所示:

set food(koala) eucalyptus; set food(chipmunk) acorn;

数组的下标也可以是一个变量,例如:

set animal chipmunk;

set food(\$animal) acorn;

访问数组和访问标量一样,使用\$符号引用数组内容,例如:

puts \$food(koala);

这一命令将显示出存放在数组 food 中下标为 koala 的值。 多维数组是一维数组的简单扩充,它们的设置方式如下:

set myarray(1,1) 0;

多维数组的引用和上述一维数组相同。

3. 数组使用

由于数组下标使用随意,从数组本身不易获得数组的统计信息,所以 tcl 提供了用来获 取有关数组信息的命令 array,以及用来输出有关数组信息的命令 parray。

下面介绍 array 的使用方法,这个命令用来获取有关数组及其元素的信息。array 命令的基本语法如下:

array option arrayname

在本节后面将讨论它所支持的选项。

有关数组最常用的信息之一是数组的大小。如果已经提供了如下说明:

set dept(0) physics; set dept(1) computer; set dept(2) art; set dept(3) geology;

那么命令:

array size dept;

将返回4,这一数值通常在循环中非常有用。

由于数组可以具有非顺序的或非数字的下标,因此 array 命令提供一个用来从数组中获 取元素的选项。假设数组 food 已经像前面介绍的那样进行了定义,那么开始获取元素所须 做的第一件事是利用 startsearch 遍历数组。这是通过首先获取数组的一个搜索 ID 来完成的, 如下所示: set food_sid [array startsearch food]

在方括号中的命令 array startsearch food 返回一个字符串,这一字符串是搜索标志的名字。这个名字将在以后的引用中用到,因此需要把它赋给某个变量。在本例中,这一变量为 food_sid。

要获取 food 数组的第一个元素 (以及其后的每个元素), 可以利用如下命令:

array nextelement food \$food_sid;

当完成对数组的搜索时,可利用如下命令终止搜索:

array donesearch food \$food_sid;

array 命令的另一个选项是在遍历数组时经常用到的 anymore。当在搜索中还有元素时, 它将返回 true (也就是1),这一命令当与前面所说明的数组 food 一起使用时,在前两次将 返回 1。

array anymore food \$food_sid;

要清除标量或数组,可以利用 unset 命令,例如:

unset physics;

这个命令将取消 physics 变量。如果利用 unset \$physics 取代刚才的 physics, 那么会得到如下一条错误信息:

can't unset "0 ": no such variable

发生错误的原因是:当把\$放在变量名的前面时,在执行命令之前变量的值将被替换进去。

21.3.4 字符串操作

tcl 提供了方便实用的字符串操作函数命令,可以执行丰富的功能。

在这些字符串命令中,最常用的命令是 string 命令, sting 命令的基本语法如下:

string options string1 [string2]

这里的 string1 和 string2 可以是字符串,也可以是变量。option 可以有如下选项,如表 21.1 所示。

选项	意义
compare	按照词典的排序方式进行比较 ,根据 string1 小于、等于或大于 string2 ,分别返回 - 1、
	0或1
first	返回在 string2 中第一次出现 string1 的位置。如果 string1 没有出现在 string2 中,则
	返回 - 1

表 21.1 option 选项表一

				(4	读表)
-		 / - -			

 last
 返回在 string2 中最后一次出现 string1 的位置。如果 string1 没有出现在 string2 中 ,

 则返回 - 1

string 命令的如下选项将把 string2 解释为从 stringl 中删除的一个列字符,如表 21.2 所示。

表 21.2	option	选项表二
--------	--------	------

选项	意义
trim	从 string1 中删除开头和结尾出现在 string2 中的字符
trimleft	从 string1 中删除开头出现在 string2 中的字符
trimright	从 string1 中删除结尾出现在 string2 中的字符

string 命令的如下选项只利用 string1 作为变元,如表 21.3 所示。

表 21.3 option 选项表三

选项	意义
length	返回 string1 包含的字符数
tolower	返回 string1 中的所有字符被转换为小写字符后的新字符串
toupper	返回 string1 中的所有字符被转换为大写字符后的新字符串

现在给出几个使用 string 命令的示例。 创建一个字符串并获取其长度,如下所示:

set str "Here Is A Test String"; string length \$str;

这一命令给出的长度为 23 (length 选项对空格字符也计算在内)。下面的语句可以获取 在 \$ str 字符串中第一次和最后一次出现字符串 st 的位置:

string first "st" \$str; string last "st" \$str

选项

意义

因为大多数字符串比较函数是区分大小写和空格的,所以这两个命令给出第一次出现 st 的值为 13,最后一次出现 st 的值也为 13。

这是下面的代码将首先把 \$ str 转换为小写字体, 然后再执行查找:

string last "st" [string tolower \$str];

此时,这一命令给出的值为 16,它相应于 String 中的 st。最后,来删除\$str 字符串前 后的空格,然后获取这一字符串的长度,如下所示:

string length [string trim \$str " "],

该命令的返回值为 21,这意味着已经删除了第一个和最后一个空格。

还有一个常用的字符串操作,就是利用 append 命令把多个字符和变量连接在一起。 append 语法如下:

append string1 string2;

返回值是一个新字符串。

21.3.5 数字操作

tcl 提供两个操作数字变量和常数的命令: incr 和 expr。

tcl 中的 incr 命令等价于 C 语言中的操作符++, 它的基本语法如下:

incr variable integer

这里的 variable 必须是一个整数, incr 命令将把给定的 integer 添加到 variable 上, 因此 减法可通过提供负整数来执行。下面演示一下它的用法。

首先,创建一个变量,并对这个变量执行 incr,如下所示:

set a 81;

- incr a;
- puts \$a;

\$a 此时具有的值为 82。在缺省情况下, incr 与++相同。当没有提供 integer 变元时, incr 将把 1 添加到所指定的变量上。现在,如果要从\$a 上减 3,可利用如下命令:

incr a -3

puts \$a

可以看到, \$a 具有的值为 79。最后一点需要说明的是:这里的 integer 可以是某个变量的值,例如:

set a 6;

set b 9;

incr a \$b;

puts \$a;

此时\$a 具有的值为 15。

对于较复杂的数学操作,tcl提供了 expr 命令,这一命令对所有标准的 ANSI C 操作符 有效,操作符的优先级大部分与 ANSI C 中的优先级相同。

当需要执行算术操作时,必须把 expr 命令放在算术操作之前,例如:

set a 20;

set b 4;

set c \$a/\$b;

puts \$c;

上述命令产生的输出结果如下:

20/4

而不是所想要的结果 5。为了获得正确的答案,需要利用 expr 命令, expr 命令的基本 语法如下:

expr function number

例如, set c [expr \$a/\$b]。

除了标准操作符+、-、*和/之外, expr 还能够识别的一些函数及其返回的值如表 21.4 所示。

函数	说明
abs(x)	x 的绝对值
round(x)	x 舍入后所得到的整数值
cos(x)	x 的余弦(x 为弧度)
cosh(x)	x 的双曲余弦
acos(x)	x 的反余弦(0~π)
sin(x)	x 的正弦(x 为弧度)
sinh(x)	x 的双曲正弦(-p/2~p/2)
asin(x)	x 的反正弦 (-p/2~p/2)
tan(x)	x 的正切(x 为弧度)
tanh(x)	x 的双曲正切
atan(x)	x 的反正切 (-p/2~p/2)
exp(x)	e 的 x 次幕
log(x)	x 的自然对数
log10(x)	x 的底为 10 的对数
sqrt(x)	x 的平方根
pow(x,y)	x 的 y 次幂

表 21.4 函数及说明

21.3.6 引用和置换

引用和置换被大量应用于与变量有关的操作。本章的前面曾提到过引用(利用双引号 构成字符串)和置换的最基本形式。tcl 还支持另一种类型的引用,即花括号引用,以及另 一种类型的置换,即命令的置换。

双引号的主要用途是创建具有内嵌空格的字符串,例如:

set chemistry "I hate this class";

双引号也可以用来创建多行的字符串。

除创建多行字符串外,在 tcl 字符串中还可以使用 ANSI C 语言的标准转义序列,例如:

set chemistry "I hate this class\n\tDo you like it";

这一命令的输出结果如下:

I hate this class Do you like it

除此以外,在双引号之间的字符串可以应用两种类型的置换。第一种类型的置换,也 就是变量置换。在利用双引号引起来的字符串中,可以通过在变量名前添加\$来访问该变量 的值,因此下面的命令:

set dept chemistry; set first "I hate this class"; set next "Do you like it"; puts "\$first,\$next ";

输出的结果为:

I hate this class, Do you like it

另一种类型的置换是命令置换。命令置换块以左括号([)开始,以右括号(])结束。 例如:

set len_in 2; puts "\$len_in inches is [expr \$len_in*2.54] cm";

这一行代码的输出结果为:

2 inches is 5.08 cm

5.08 是如下命令:

expr \$len_in*2.54

产生的结果。

由于这一命令位于括号中,因此它的返回值将被替换进去。在本例中使用的是 tcl 命令 expr,不过任何 tcl 命令都可以放在括号之中。命令置换可用于大多数命令中,而不只限于 由双引号括起来的命令,例如:

set len_in 2; set len_cm [expr \$len_in*2.54]; puts "\$len_in inches is \$len_cm cm";

这几个命令产生的输出结果与下面一行代码相同:

set len_in 2; puts "\$len_in inches is [expr \$len_in*2.54] cm";

在 tcl 中还有另一种类型的引用,即花括号的引用,这种类型的引用类似于 shell 中单 引号的使用。花括号的引用将利用给定的字符来创建字符串集,其中不进行置换,并且也 不对特殊字符进行解释。例如:

puts "This\nis a \nmulti-line\nstring"

```
这一命令产生的输出结果如下:
```

This

is a

multi-line

string

而下面的命令:

puts {This\nis a \nmulti-line\nstring}

产生的输出结果如下:

This\nis a \nmulti-line\nstring

要想在花括号括起来的字符串中使用制表符、换行符以及其他的特殊字符,必须按原 样实际地输入它们,例如:

puts {This is a

multi-line

string}

这一命令将产生想要的输出结果。

21.3.7 条件语句

tcl 支持 if 语句和 switch 语句,并且支持有关字符串和数字的所有 ANSI C 的标准比较符。

1.if 语句

if 语句的语法如下:

if statement {

commds;

} else if statement {

commands

} else {

commands

}

tcl 对花括号和空格的使用是十分苛刻的。在 if、elseif 或 else 语句中的开始的花括号必须要和 if、elseif 或 else 在同一行中。而括号外必须有一个空格。且 else 或 elseif 语句必须 要在前一个 if 或 elseif 的后括号的同一行中。

可以根据需要添加任意多的 elseif 语句,如:

```
if {$x == 0} {
set x 10;
} elseif {$x == 10} {
```

```
incr x -1;
} elseif {$x == 100} {
       set x 50;
} else {
       set x 0;
}
```

2. switch 语句

switch 语句是复杂的 if 语句的简化形式。switch 把某一值(字符串或数字)与相应的 代码块对应起来。当利用 switch 语句编写代码时,上面的 if 语句将变,如下所示:

```
switch $x {
       0 \{ set x | 10; \}
       10 {incr x -1;}
       100 {set x 50;}
```

}

}

在缺省情况下,只有对应于匹配值的代码被执行。但如果代码块被指定为一个减号 (-), 那么略过该 switch 语句,执行后面的代码块,例如:

```
switch $x {
      0 -
       10 -
       100 {incr x -1};
```

该 switch 语句等价于下面的 if 语句:

```
if {(x = 0) || (x = 10) || (x = 100) || (x = 100) } {
        incr x -1;
}
```

21.3.8 循环语句

tcl 支持 for、foreach 和 while 循环命令。此外 tcl 还提供循环控制命令 break 和 continue。 在 for、foreach 或 while 语句中的开始的花括号必须要和 for、foreach 或 while 在同一 行中。而括号外必须有一个空格。

1. for 语句

for 循环是最常用到的循环,它的结构如下:

for {initialization} {condition} {increment} {body}

下面是一个计数到 10 的简单的 for 循环的示例:

for {set i 0} {\$i<=10} {incr i} {puts \$i;}

2. foreach 语句

foreach 将在一组变元中进行循环,并且每次都将执行它的循环体。foreach 循环具有如 下结构:

foreach variable {items} {block}

这里 variable 是变量的名字,位于 items 集合中的每个元素将依次赋给这个变量。下面 是 foreach 循环的一个示例:

```
foreach element {o c n p li} {
    switch $element {
        o -
        n {puts gas;}
        c-
        p-
        li {puts solid;}
    }
```

}

在这个示例中,item 集合是利用要查看的元素列表来指定的,其实这里也可以利用变量,例如:

```
set elements "o c n p li";
foreach element $elements {
switch $element {
o-
n (muts gas:)
```

```
n {puts gas;}
c-
p-
li {puts solid;}
}
```

}

如果给定的是一个变量而不是元素列表,则不应该使用花括号,因为花括号将被作为 用于引用的花括号对待。

3. while 语句

当测试条件为 true 时, while 循环将执行它的循环体。while 循环的结构如下:

while {statement} {commands}

下面的代码是一个简单的 while 循环,它共计要循环 10次:

```
set x 0;
while {$x<10} {
     incr x;
     puts$x;
}
```

330

4. break 和 continue 语句

break 命令用来中断循环并执行循环代码块之后的下一行代码。 continue 命令用来跳到循环的下一次执行。

continue 命令对于读入允许使用注释行的初始化文件非常有用。例如,下面的语句包 含在一个读入文件的循环中,所有以#开头的行都将被跳过。

if {[regexp {^#} [string trim \$line]]} {continue;}

21.3.9 文件操作

tcl 提供一个关于文件输入输出的简单而有效的方法,这一方法类似于 C 的标准 I/O 库 的方法。对于文件的输入输出,第一步是打开文件并获取文件句柄或文件 ID。这一命令将 以 r 方式打开文件/etc/passwd 并返回一个文件句柄,这一句柄将被赋值给变量 file。

setfile [open/etc/passwd r];

tcl 支持如表 21.5 所示的几种文件打开方式。

表 21.5 tcl 的文件打开方式

打开方式	具体意义
r	以只读方式打开文件。文件必须存在
r+	以可读可写的方式打开文件。文件必须存在
w	以可写的方式打开文件。如果文件不存在,它将被创建;如果存在,它将被修改
w+	同 w 一样。除此之外,还强调对文件可读
а	以追加文本的方式打开文件。文件不存在时将被创建
a+	同 a 一样。除此之外,还强调对文件可读

如果文件以可读的方式打开,那么就可以利用 gets 命令从这个文件读取信息。为了处 理文件的所有行,经常需要用到下面的 whlie 命令:

while {[gets f line] >= 0}

gets 命令将从文件句柄\$f 中读入一行,并把该值赋给变量 line。在循环体中,可以对 \$line 进行访问和操作。当到达文件结束时,gets 命令将返回 - 1。

如果以可写的方式打开了文件,那么就可以利用 puts 命令把输出写到这个文件中。如果文件句柄\$f 对应于一个以可写方式打开的文件,下面的命令将把字符串 This is a line of text 写到这个打开的文件中。

puts \$f "This is a line of text";

关闭文件的命令是 close。该命令利用文件句柄作为它的参数。要关闭前面打开的文件, 只需使用如下命令:

close \$f;

在 tcl 中,可以使用 file 命令来获取有关文件的信息。file 命令的语法如下:

file option filename

这里的 filemame 是需要获取信息的文件。option 是如表 21.6 所示的选项之一。

表 21.6 option 的可选项

选项	作用
Executable	如果当前用户对该文件可执行,则返回 true
Exists	如果文件存在,则返回 true
Isdirectory	如果文件是一个目录,则返回 true
Isfile	如果文件是一个普通文件,则返回 true
Owned	如果当前用户拥有该文件,则返回 true
Readable	如果当前用户对该文件可读,则返回 true
Writable	如果当前用户对该文件可写,则返回 true
Atime	返回自 1970 年 1 月 1 日以来最后一次访问文件的时间,以秒表示
Mtime	返回自 1970 年 1 月 1 日以来最后一次修改文件的时间,以秒表示
Size	返回文件的大小,以字节表示
Readlink	当给出的文件是符号链接时,返回符号链接的值
Туре	返回给出文件类型的字符串

21.3.10 过程

tcl 中的过程等价于 C 语言中的函数。要创建过程可利用 proc 命令。该命令的语法如下:

proc procedure_name {arguments} {body}

其中参数的数目是可变的,空的参数表由{}来表示; body 可以包含任何有效的 tcl 语 句并且长度不限。

如下所示是一个无参数的简单过程:

proc test_proc { } {puts "procedure test"; }

要调用该过程,只需给出它的名字即可,例如:

procedure test;

将得到输出结果。

下面是一个较实际的示例,它可以像 cat 命令一样输出文件。这个过程以文件名作为 自己的一个变元:

```
proc cat {filename} {
    set f [open $filename r];
    while {[gets $f line]>=0} {
```

```
puts $line;
}
close $f;
```

在过程的开始的花括号必须要和 proc 的名称在同一行中。而括号外必须有一个空格。 要想以/etc/passwd 为参数调用该过程,可利用如下命令:

cat /etc/passwd

该命令将显示出/etc/passwd的内容。

21.4 tk 入门

使用 tk 可以方便地创建一些简单的 X Window 图形用户界面。限于篇幅,本节将简单介绍 tk 的概念并给出一个简单的演示。

21.4.1 tk 构件

tk 是一个利用 tcl 脚本语言编写 X Window 图形用户界面(GUI)的工具箱。tk 为 tcl 语言添加了创建 GUI 组件的功能,这种 GUI 组件通常称为构件(widget)。tk 定义的构件 类型如表 21.7 所示。

构件类型名	
canvas	用于绘制对象
entry	用于单行文本的输入
frame	用来包含其他构件
listbox	显示一组字符串并允许对其中的一个或多个字符串进行选择
menu	显示菜单栏和菜单项
text	显示多行文本
label	显示单行静态文本
button	显示一个可点击按钮的构件
Checkbutton	显示一个可选的方框
Radiobutton	显示多个互相排斥只能单选的方框
scale	类似滚动条,用来设置一个值

表 21.7 tk 构件类型

要想创建和操作构件,必须使用 wish (windowing shell)。当需要交互地调用 wish 时,在 Linux 提示符下输入 wish,此时将会出现如下 wish 提示符:

%

与此同时,将弹出一个空的窗口。这一窗口是 wish 的根窗口(称为.),以后创建的所

}

有构件将随此窗口一起显示。

21.4.2 创建构件

创建构件的基本方法如下:

widget_type path option

这里的 widget_type 是表 21.7 中列出的构件类型之一 ;path 是窗口的路径名(一般是.); option 是构件能够识别的任一选项。

下面的语句创建了一个按钮:

button .button;

该行中指定的构件类型为 button,因此创建一个按钮。设定的路径为.button,所以 tk 将在根窗口.中创建按钮,并把它命名为 button。

创建按钮后并没有立即显示。需要用 pack 命令显示按钮并给出所要显示的构件的路径, 如下所示:

pack.button;

显示的按钮是空的。下面介绍使用构件的选项来控制构件的外观和功能。

21.4.3 构件的选项

所有构件都可以使用标准的选项来控制它们的外观和功能。大多数构件可以识别如表 21.8 所示的选项。

选项	意义
-background color	构件的背景色。有效值的格式为#RRGGBB 和#RRRGGGBBB,或者是
(-bg color)	/usr/lib/X11/rgb.txt 所定义的名字之一
-foreground color	构件的前景色。有效值的格式为#RRGGBB 和#RRRGGGBBB,或者是
(-fg color)	/usr/lib/X11/rgb.txt 所定义的名字之一
-height pixels	构件的高度(以像素为单位)
-width pixels	构件的宽度(以像素为单位)
-borderwidth pixels	构件的边缘宽度(以像素为单位)
(-db pixels)	
-padx pixels	构件在 x 方向上所要求的附加空距
-pady pixels	构件在 y 方向上所要求的附加空距
-relief type	构件的 3D 效果, type 是下列字符串之一: flat、raised、grove、ridge 和 sunken
-text string	在构件中显示的字符串

表 21.8 构件选项及其意义

选项	意义	
-font font	在构件中显示文本所使用的字体。	有效的字体定义由命令 xlsfonts 给出

-command command

在构件中显示文本所使用的字体。有效的字体定义由命令 xlsfonts 给出 当利用构件时所执行的 tcl 命令,通常是一个过程名或 exec 语句

pack 命令可以识别的一些选项如表 21.9 所示。

表 21.9 pack 命令可以识别的选项及意义

选项	意义
-side type	控制构件排放的位置。有效排放类型为 left、right、top 或 bottom。例如,left
	表示新构件应排放在现有构件的左边
-fill type	控制构件是否充满窗口所打开的空间。有效值为 none、x、y 或 both。例如, both
	表示构件应充满被打开的整个空间
-expand value	控制构件是否随着窗口大小的增加而扩大。value 或者是 0 或者是 1 表示构件的
	大小随窗口的变化而变化

21.4.4 编程示例

在学习了有关构件和 pack 选项的知识后,就可以着手使用它们。构件具有一个特性叫做 relief,即构件的三维效果。为了了解每种 relief 类型的外观表现,可以利用如下命令制作一些标签:

```
foreach i {raised sunken flat groove ridge} {
    label .$i -relief $i -text $i;
    pack .$i ;
}
```

该示例将遍历 relief 的所有类型,对于每种类型创建一个标签,并把每个标签的 text 选项设置为相应 relief 类型的名字。整个布局将如图 21.1 所示。

有两点需要注意:第一是标签的大小不一;第二是标签的排放方式是一个放在另一个 之上。这种情况是 pack 命令缺省表现形式的一个示例, pack 命令将自动确定每个构件的大 小,然后把每个构件放置在前一个构件的下面。



图 21.1 各种显示效果的标签

现在,把所有的标签设置为同样的大小,并把它们并排放置而不是一个标签压在另一 个标签之上。为达到这一目的,可以有两种方法。第一种方法是,把循环改写为: foreach i {raised sunken flat groove ridge} {

label .\$i -relief \$i -text \$i -height 10 -width 10;

pack .\$i -side left

}

第二种方法是,利用 configure 选项重新配置标签。configure 选项的语法如下:

widget configure option

在这里可以利用下面的循环(在创建标签之后):

```
foreach i {raised sunken flat groove ridge} {
```

.\$i configure -height 10 -width 10;

pack .\$i -side left

}

```
为什么要利用 configure 呢?
```

这是因为当交互运行 wish 时,如果已经提供了循环的一个版本,那么当修改并再次运行它时,将会产生下面的错误信息:

window name "raised" already exists in parent

通过这句话, wish 告诉程序员, 他的程序试图重新创建现存的构件(在这里是标签 raised)。

由于这一原因,所以需要使用 configure。事实上,任何需要修改现有构件的情况都要 求使用 configure。

在本例中,想要使用循环新版本的唯一方法是:利用 destroy 命令破坏现有的标签。 如下所示:

foreach i {raised sunken flat groove ridge} {destroy .\$i}

这个示例有两个问题应该解决。一个问题是标签之间相距太近,因此难以区分,另一 个问题是窗口的大部分是空的。

通过在排放标签时填充空白并通过增加它们的边框宽度,可以使它们很容易地被区分 开来。为了使标签占用全部可用的空间,可以把 pack 的 fill 选项设置为 both,从而使标签 在 x 和 y 方向上都进行延伸,并且把 expand 选项设置为 true,如下所示:

foreach i {raised sunken flat groove ridge} {

label .\$i -relief \$i -text \$i;

.\$i configure -height 5 -width 5 -borderwidth 5;

pack .\$i -side left -padx 5 -fill both -expand 1;

}

此时产生的结果将如图 21.2 所示。

注意:为了查看 fill 和 expand 选项的效果,需要重设窗口的大小,并使标签适应 窗口。通过把 label 替换为不同类型的构件,可以很容易地对这个示例进行修改, 从而使它显示出各种效果的标签。



图 21.2 各种显示效果的标签

21.5 习 题

- 1. 比较 shell、perl 和 tcl 的语法,着重比较条件和循环语句的异同。
- 2. 累加 10 以内的自然数, 然后将结果输出到屏幕。
- 3. 从终端读入字符串, 然后写入一个文件。