

中国软件行业协会国际自由软件应用研究发展分会
与电子工业出版社联合向读者推荐 《自由软件丛书》

[英] Phil Cornes 著

童寿彬 等译 夏道藏 审校

Linux

从入门到精通

The **Linux A-Z**



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

Linux 从入门到精通

The Linux A - Z

[英] Phil Cornes 著

童寿彬 等译

夏道藏 审校

電子工業出版社
Publishing House of Electronics Industry

内 容 简 介

Linux 是符合 POSIX 标准的、具有全部 UNIX 特征的操作系统。它是个人计算机的 UNIX。在 Internet 和 Intranet 领域具有广泛的应用前景。本书划分为循序前进的五篇,使读者可以按自己的知识水平进行学习。基础篇引导没有 UNIX 经验的读者成为普通的 Linux 用户;管理篇指引读者如何获取、安装和配置 Linux 系统,并叙述如何成为名副其实的系统管理员;系统程序设计篇假定读者有一定的使用 C 语言进行程序设计的经验,本篇包含了 Linux 系统调用的主要方面,还包括了完整的实例研究;设备驱动程序篇提供了 Linux 内核和机器硬件之间的接口,包括设备驱动程序实例研究的源程序;内核篇选择了一些与内核有关的题目进行讨论,并以实例研究的形式提供了简单的用户线程库。

Translation copyright © 1998 by Publishing House of Electronics Industry.

The Linux A ~ Z.

Copyright © 1997 by Prentice Hall Europe.

All rights reserved.

Published by arrangement with the original publisher, Prentice Hall Europe, a Simon & Schuster company.

本书中文简体版版权由美国 Prentice Hall 欧洲公司授与电子工业出版社。未经出版者书面许可,不得以任何方式或手段复制或抄袭本书内容。版权所有,侵权必究。

书 名: Linux 从入门到精通

著 者: [英] Phil Cornes

译 者: 童寿彬 等

审 校 者: 夏道藏

责任编辑: 黄志瑜

特约编辑: 李洁生 陈 熠

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京市大中印刷厂

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787 × 1092 1/16 印张: 24 字数: 595 千字

版 次: 1998 年 7 月第 1 版 1998 年 7 月第 1 次印刷

书 号: ISBN 7-5053-4770-5
TP·2306

定 价: 40.00 元

著作权合同登记号 图字: 01-96-0877

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

前 言

Linux 可供实际使用已有若干年了。它是符合 POSIX 标准的操作系统。它不仅可以用在 Intel 系列个人计算机上,也可以用在 DEC Alpha 和 SUN SPARC 系列上。Linux 内核的创建是由芬兰赫尔辛基的 Linus Torvalds 个人完成的。以后,大量的系统软件设计专家共同对它进行改进和提高。到现在为止, Linux 已成为具有全部 UNIX 特征与 POSIX 兼容的操作系统。只要遵守自由软件基金会发布的通用公共许可证(GPL)条款,就可以自由使用 Linux 源程序。这样做的实际结果是任何人都可以随心所欲地取得、修改和传播 Linux 内核和所有的源程序。

这种自由利用 Linux 源程序的可能性激发了世界范围内热衷于计算机事业的人们的创造力。通过 Internet,这一软件的传播和使用就象野火一样迅速蔓延。现在,所有 UNIX 的主要功能都有相应的 Linux 工具和实用程序。和内核一样,只要你遵守 GPL 条款,就可以自由使用这些工具和实用程序。将这些综合在一起,实际就是一个完整的 UNIX 类操作系统。它可以供任何与 Internet 联网的用户自由使用。这是早期的情况。Linux 的传播很快超出了 Internet 的范围,深入到计算机应用的各个领域。从各种 Linux 发布广泛使用 Internet 以外的手段就足以证明这一点。这些手段包括电子公告牌,软盘组和光盘等。

Linux A-Z 适合于作为大学教材使用。适当增加一些源程序,也可以作为研究生课程使用。它是在作者的讲稿和实例研究的基础上写成的。这些原始材料已由作者在 Staffordshire 大学作为操作系统课程使用了一段时间。本书广泛的覆盖面使它成为学习 Linux 或操作系统课程的学生和那些愿意在专业知识基础上进一步深造的专业人员的主要参考书。

对读者的要求是掌握计算机科学课程的基本知识。特别是在个人计算机使用方面应有一定的经验,如掌握 PC 的硬件配置、磁盘分区以及系统软件,如 DOS 和 Windows 的安装等方面的知识。对本书后面三篇还要求读者有一定的 C 语言使用经验。这样便可以从这些材料的学习中取得更好的效果。

本书从各个不同侧面提供基础知识,为读者全面掌握 Linux 操作系统及有关的工具和实用程序的使用铺平道路。先从介绍基本概念入手,然后引导大家实践这些概念以巩固所学的知识。本书分五篇,使读者逐步深入掌握 Linux。从培训普通用户开始,通过系统的安装和管理,到用 Linux 系统调用(它的应用程序接口 API)进行程序设计,再进一步掌握各种硬件接口。在本书的最后,还探讨了一些深奥的内核秘密。

本人认为计算机应用是一门类似手工艺的实践技能。没有任何东西可以取代对这些材料的第一手实践经验。本书自始至终贯彻强调实践的观点。本书后面还提供了供进一步实践的完整实例研究。

本书五个篇章的组成是:

1. 基础——包括 1 到 6 章。在这部份中假定读者有使用个人计算机的经验,但没有使用 UNIX 或 Linux 的经验。逐步引导他们成为 Linux 的普通用户。从登录开始,经过学习典型的文件结构和使用一些简单的命令后,进入 vi 编辑程序,标准的 bash shell 以及有用的工具和实用程序的具体应用。最后,学习如何编写可供 Shell 执行的批处理程序 Script。

2. 管理——当初学者掌握基础知识以后或者有经验的用户已经具备这方面的知识,就

可以进而学习如何履行管理员的职责。7到14章讲述Linux系统管理方面的知识。从一台装有MS-DOS的个人计算机开始,对它进行重新配置,使得Linux可以和用户已有的应用同时放在一台机器上。一旦安装了Linux,就可以按管理员的特殊要求对系统进行配置。从如何配置和重建操作系统内核开始,然后建立用户帐号和配置各种可供使用的资源。资源的清单包括:打印机,调制解调器,鼠标,X-Windows客户程序和服务程序,串行、并行和网络通信,Internet以及DOS模拟程序。

3. 系统程序设计——这是进入Linux内核的艰难旅程的第一步,内容包括15到23章。从应用程序的角度看,这些章节包含了Linux系统调用的主要方面。包括的题目有:文件和终端的输入/输出;进程控制;不同的进程之间的通信机制(IPC),包括机器内部的进程之间和不同机器的进程之间的通信;称为信号(signal)的高级软件中断以及独立的守护(daemon)进程。第三篇还包括三个完整的实例研究:小型的Socket IPC库,小型的实例Shell以及提供简单的文件传输服务的守护进程。分别给出了源程序和注释。

4. 设备驱动程序——讨论了Linux应用程序接口以后,24到27章将讨论另一个主要的操作系统接口,即操作系统内核和机器的外部设备之间的接口。由设备驱动程序提供这一接口。它们分为字符设备驱动程序和数据块设备驱动程序两个大部分。本部分还讨论了供驱动程序内部使用的,Linux内核提供的支持功能。最后讨论一个简单驱动程序的实例研究以巩固所学的概念。

5. 内核——本书最后一篇包括28到32章,讨论有关Linux内核的一些有选择性的题目,以体会Linux各部分如何协调进行工作。这些题目包括:进程调度;Linux内存管理操作;允许Linux访问不同类型文件系统的虚文件系统转换(VFS);以及从内核的角度再次讨论系统调用。作为实例研究,还讨论了一个非常简单的用户线程(thread)库。

五个实例研究的源程序允许用户按自由软件基金会的通用公共许可证条款自由使用。它们可以用匿名ftp从Staffordshire大学计算机学院的ftp服务器

`ftp.soc.staffs.ac.uk`

在目录

`users/phil_cornes/linuxa2z`

下得到。

为了能从本书中受益,不管你当前的技能和专业水平如何,对你的唯一的要求是对这一课程有兴趣,并有更多地掌握这一专业知识的愿望。

译者的话

Linux 是在个人计算机和工作站上使用的 UNIX 操作系统。它不仅仅可以用在 Intel, AMD 及 Cyrix 系列个人计算机上,还可以用在许多工作站上。

Linux 是真正的多用户、多任务操作系统。在个人计算机和工作站上使用 Linux,能更有效地发挥硬件的功能,使个人计算机可以作为工作站和服务器使用,使工作站发挥出更高的效能。经过七年多时间的发展,大量自由软件已移植到 Linux 操作系统上,形成了完整的体系。现在,世界各地已有各种 Linux 发行版本。它们不仅仅包含 Linux 操作系统内核,还包括各种语言编译程序、数据库管理系统、图形用户界面、通信联网工具以及大量的其他应用软件。因此, Linux 是一个优秀的应用软件开发平台。

Linux 继承了 UNIX 的主要特征。它在 Internet 和 Intranet 的应用中占有明显优势,在教学和科研领域具有广阔的应用前景。自由软件的开发采用了开放和协作的开发模式。源程序是开放的,允许任何人取得、修改和重新发布自由软件的源程序。这种开发模式有利于充分地发挥程序开发人员的集体智慧,大量减少不必要的重复劳动,并使得自由软件的脆弱点能够及时发现和得到克服,使自由软件具有强大的生命力。随着为自由软件提供技术支持和信息服务的 Internet 网点的不断增加,随着在自由软件基础上的商品化软件的出现,以及在自由软件基础上的产业的兴起,标志着自由软件开始走向成熟。随着一些著名公司的产品加入自由软件的行列,势必增强自由软件的发展势头。Linux 正在竞争 UNIX 和 Windows NT 的市场份额。它也是目前唯一可以与 Microsoft Windows 抗衡的操作系统。

Linux 的基本系统对硬件要求很低。可以从网上下载 Linux 或用很低代价买到 Linux 的光盘。无疑这是学习 UNIX 类操作系统的理想工具。广大个人计算机用户只要有决心,都有条件使用它和掌握它。衷心希望广大的计算机爱好者投入推广自由软件的活动。

我国软件行业的许多领域一直处于空白状态。在自由软件走向成熟之际,充分发挥我国软件人员的积极性和创造能力,正是我国振兴软件行业的好时机。充分利用自由软件的宝贵资源,有利于我们从高起点起步,以便在比较短的时间内改变我国软件行业的落后状态。发挥自由软件资源的作用,也将为我国许多公司创造无限的商业机会。

为了推动自由软件在国内的应用,在胡贻志教授的大力倡导下,成立了国际自由软件应用研究发展分会。在梁祥丰社长和龚兰方总编的大力支持下,电子工业出版社与《分会》联合推出了自由软件丛书。《丛书》的选择得到了宫敏博士和黄志瑜主任的具体指导。《Linux 从入门到精通》是其中的第一本。

本书是一本很好的操作系统教材。它的内容充实,概念清晰,叙述连贯,有利于读者全面掌握 Linux 操作系统的总体概念,为开展各种应用打下坚实基础。参加本书译翻工作的有童寿彬,叶嵘,童悦,孟明宇,周庆华,刘捷,方鸣,陈磊等。本书由国家信息中心夏道藏总工审校。

译者水平有限,不当之处希望读者批评指正。

译者 1998.2.15.

目 录

第一篇 基础	(1)
第一章 Linux 基础	(2)
1.1 登录	(2)
1.2 文件系统的层次结构	(3)
1.3 考察文件系统的层次结构	(5)
1.4 口令文件	(6)
第二章 Linux 命令	(9)
2.1 目录的层次结构	(9)
2.2 文件系统	(11)
2.3 处理文件	(13)
2.4 联机帮助	(15)
2.5 安全	(16)
2.6 进程	(19)
2.7 空间	(21)
第三章 正文编辑	(27)
3.1 编辑方式	(27)
3.2 插入方式	(30)
3.3 命令方式	(31)
第四章 Bash	(37)
4.1 路径名扩展	(37)
4.2 输入/输出重新定向	(39)
4.3 管道	(41)
4.4 后台作业	(42)
4.5 作业控制	(43)
4.6 历史表	(44)
4.7 命令行编辑	(46)
4.8 命令补全	(46)
4.9 shell 函数	(47)
第五章 工具和实用程序	(50)
5.1 搜索和排序	(50)

5.2	修改文件	(58)
5.3	简单的数据处理	(64)
5.4	其他工具	(68)
5.5	DOS 文件	(70)
第六章	Shell Script	(73)
6.1	命令文件	(73)
6.2	变量	(73)
6.3	shell script 参数	(76)
6.4	登录 script	(77)
6.5	更多变量的赋值方法	(78)
6.6	自动化状态变量	(80)
6.7	流程控制	(81)
6.8	信号	(90)
第二篇	管理	(95)
第七章	准备和运行	(96)
7.1	硬件	(96)
7.2	各种 Linux 发布	(97)
7.3	安装 Linux	(98)
7.4	建立内核	(108)
第八章	用户	(114)
8.1	帐号的口令	(114)
8.2	增加用户帐号	(115)
8.3	用户组	(116)
8.4	安全	(117)
8.5	成批作业	(119)
8.6	档案	(122)
第九章	外部设备	(126)
9.1	控制台终端	(126)
9.2	调制解调器	(130)
9.3	鼠标	(133)
9.4	打印机	(135)
第十章	X-windows	(138)
10.1	服务程序和客户程序	(138)
10.2	X 配置	(138)
10.3	启动 X-windows	(143)

10.4	fvwm	(144)
10.5	xterm	(145)
第十一章 通信		(146)
11.1	独立的机器	(146)
11.2	TCP/IP	(146)
11.3	PPP	(151)
11.4	其他连接方法	(154)
第十二章 Internet		(158)
12.1	e-mail	(158)
12.2	telnet	(160)
12.3	ftp	(161)
12.4	archie	(163)
12.5	Usenet 新闻	(164)
12.6	gopher	(166)
12.7	world wide web	(167)
第十三章 运行 DOS		(169)
13.1	DOSEMU	(169)
13.2	安装	(169)
第十四章 安全		(172)
14.1	文件的权限	(172)
14.2	访问设备	(173)
14.3	口令	(174)
14.4	root 帐号	(174)
第三篇 系统程序设计		(177)
第十五章 文件		(178)
15.1	顺序文件	(178)
15.2	随机文件	(188)
15.3	终端输入/输出	(190)
15.4	索引节点	(195)
15.5	改变文件属性	(199)
15.6	文件系统的层次结构	(202)
第十六章 进程控制		(206)
16.1	进程识别号	(206)
16.2	用 fork 创建进程	(207)

16.3	exec 系统调用	(209)
16.4	wait 和 exit 系统调用	(212)
第十七章	进程间通信	(217)
17.1	管道	(217)
17.2	输入/输出重新定向	(219)
17.3	FIFOs	(223)
17.4	system V 的进程间通信机制	(224)
17.5	Sockets	(228)
第十八章	极小的 socket 库——实例研究	(236)
18.1	库函数	(236)
18.2	样本服务程序	(243)
18.3	样本客户程序	(246)
第十九章	信号	(253)
19.1	signal 系统调用	(253)
19.2	sigaction 系统调用	(255)
19.3	kill 系统调用	(257)
19.4	pause 系统调用	(258)
19.5	alarm 系统调用	(258)
第二十章	极小的 shell——实例研究	(261)
20.1	数据结构	(261)
20.2	代码	(262)
第二十一章	各种系统调用	(276)
21.1	umask 系统调用	(276)
21.2	mount 系统调用	(277)
21.3	umount 系统调用	(277)
21.4	sync 系统调用	(277)
21.5	显示时间	(278)
21.6	select 系统调用	(280)
第二十二章	守护进程	(283)
22.1	原理	(283)
22.2	实践	(283)
第二十三章	极小的守护进程——实例研究	(288)
23.1	守护进程清单	(288)

23.2	socket 服务程序增加的内容	(293)
23.3	socket 服务程序的改变	(293)
第四篇 设备驱动程序		(295)
第二十四章 设备驱动程序基础		(296)
24.1	概述	(296)
24.2	设备驱动程序子程序	(297)
24.3	设备驱动程序原理	(298)
第二十五章 字符设备驱动程序		(300)
25.1	登记字符设备	(300)
25.2	工作内存	(301)
25.3	基本入口点	(303)
25.4	特殊的控制函数	(306)
25.5	中断	(307)
25.6	设备驱动程序安装	(308)
第二十六章 微小设备驱动程序——实例研究		(309)
26.1	Header 信息	(309)
26.2	init 函数	(312)
26.3	open 函数	(313)
26.4	release 函数	(314)
26.5	write 函数	(314)
26.6	read 函数	(316)
26.7	ioctl 函数	(317)
第二十七章 块设备驱动程序		(319)
27.1	块设备驱动程序的概念	(319)
27.2	登记块设备驱动程序	(322)
27.3	中断	(324)
27.4	计时程序	(327)
第五篇 内核		(329)
第二十八章 进程调度		(330)
28.1	背景	(330)
28.2	细节	(331)
第二十九章 小型线程——实例研究		(334)
29.1	库调用	(335)
29.2	调用 C 函数	(336)

29.3	线程调度	(338)
29.4	环境切换	(339)
29.5	启动新线程	(340)
29.6	约会机制	(343)
29.7	建立通道	(345)
29.8	send 和 receive 函数	(346)
29.9	无界缓冲区	(348)
第三十章 内存管理		(350)
30.1	受保护的编址方式	(350)
30.2	分页技术	(351)
30.3	Linux 特色	(353)
第三十一章 文件存储		(354)
31.1	VFS	(354)
31.2	System V	(355)
31.3	minix	(359)
31.4	ext2	(360)
第三十二章 系统调用		(362)
32.1	中断和异常	(362)
32.2	系统调用机制	(362)
32.3	增加新的系统调用	(363)
GNU 通用公共许可证 (GPL)		(368)

第一篇

基 础

第一章 Linux 基础

在本书第一篇中,假定你对 Linux 或任何 UNIX 类操作系统完全没有经验或只有很少经验。从一个新用户开始,也就是说,从一个刚从 Linux 系统管理员那里取得帐号,登录名和口令的用户开始,逐步掌握 Linux 基础知识。

这是一种理想的学习情况。但是,我知道许多读者在一开始就不仅仅是一个新用户,他还必须同时担负起系统管理员的责任。处于这种情况下,最好交替阅读第一篇和第二篇前几章的内容。这样你就能在学习基础知识的同时,完成最少必需的安装和系统管理任务。事实上,这些任务并不像听起来那么困难。因为现在可供使用的许多 Linux 发布(distribution)中几乎所有的配置参数都有默认值。使它们用起来非常方便。但是,你仍然不能掉以轻心。对于一个完整的 Linux 操作系统的某些部分的安装和配置仍然需要十分小心地对待。

1.1 登录

作为开始,你的 Linux 将发出下列登录提示符,邀请你登录到系统中来:

```
login:
```

在这里敲入用户登录名,并按 enter 或 return 键。如果你的帐号设置了口令,系统将发出下列提示符请你输入口令:

```
password:
```

在输入口令后,再按 enter 或 return 键。在屏幕上不显示作为口令输入的字符。如果正确地送入了登录名和口令,系统将送回提示符请你输入命令。如果在登录过程中出现了错误,系统将发出登录错误信息:

```
Login incorrect
```

并发出新的登录提示符让你重试。注意,如果你是新用户又是系统管理员,在一般情况下,应该尽量避免使用 root 登录帐号。除非要完成系统管理任务,在非用不可时你才动用它。理由是:root 是一个特权登录帐号,它拥有很大的权力。它能越过 Linux 正常的安全和完整性检查。即使一个有经验的管理人员,偶尔犯下耗费时间和代价高昂的错误也是常有的事。

要注意的另一点是 Linux(大多数 UNIX 类系统也一样)区分大、小写。小写字母和对应的大写字母被看作是不同的字符。在多数情况下, Linux 要求用小写字母输入字和命令,大写字母不起作用。

作为普通用户帐号,当你成功地完成登录后,系统将执行一个称为 shell 的程序。任何程

序,当它正在系统内执行时,被称为进程(process)。正是你的 shell 进程给你提供了命令行提示符。作为默认值,对非特权用户用美元符号 \$ 作提示符,对系统特权用户(root)用 # 号作提示符。shell 还承担从键盘接受输入的命令,并安排它们执行的任务。

由于历史原因,实际上有许多不同的 shell 程序可供使用。它们大致提供相同的功能。在本书中将集中使用自由软件基金会(GNU)提供的称为 bash 的 shell 程序。它也是大多数 Linux 发布默认的 shell 程序。

一旦出现了 shell 提示符,就可以送入命令名称和命令所需要的任何参数。shell 将执行这些命令。如果一条命令花费了很长的时间来运行,或者在屏幕上产生了大量的输出,你可能希望中断它。在它正常结束之前,停止它的执行。对多数命令讲,你可以从键盘上按 ctrl-c(即在按下 ctrl 键后,同时按 c 键),用它来送出中断信号。

当你准备结束你的登录对话过程(session)时,可以在提示符后面送入 logout 命令,退出登录。还有其它退出登录的方法,最常用的是在提示符后面送入文件结束符(EOF)。这可以在键盘上按 ctrl-d 来实现。这里的问题是:shell 可以被设置成忽略 EOF。一些 Linux 发布就是这样默认设定的。我们在后面讨论 shell 的细节时,你会知道如何定制 shell 的响应方式。

1.2 文件系统的层次结构

当你完成登录后,下一任务是熟悉基本文件系统的结构。Linux 文件系统包含 3 类文件:

普通文件:这些文件只是字节的集合。系统没有在文件中加入特定的结构。它们用作文本文件(包括源程序文件),程序使用的数据文件,以及程序本身的可执行二进制文件。

目录文件:目录是一种结构。它容许将一组文件放在一起。从概念上讲,目录好比是一个容器,可以用来存放其他文件和目录。事实上,目录本身只包含其他文件的名称和一些类似如何从磁盘上找到这些文件的简单信息。由于一个目录可以包含子目录名称,文件系统形成一个层次结构。

特殊文件:许多不同的文件类型属于这一范围。当你深入学习时将看到这一切。简单讲,特殊文件与进程之间的通信以及进程和连接到机器的各种各样的外部设备之间的通信有关。

所有这些类型的文件均放在一个大的树形层次结构中(见图 1.1)。树的顶部是一个单独的目录,称为根(root)目录(请勿与 root 登录名相混淆)。并且用斜杠符号 / 表示根目录。在根目录下,有一些用于不同目的的标准子目录和文件。这些高层的目录和文件结构只是从传统的意义讲是标准的,但并非一定要那样去做。

图 1.1 表示一个根目录(/)包含了 dev, bin 和 home 子目录。接下去 home 目录又包含了 mike 和 lynne 目录等等。在目录的最底层包含普通文件和特殊文件(如 a 和 b)。有了这些文件和目录之间的关系后,你可以说 home 是 mike 的双亲目录,而 a 是 proj1 目录的一个子文件。

两个或多个文件或目录具有相同名称的情况常常会出现。只要它们包含在不同的双亲目录中,就不会发生混淆。很明显,需要有无二义性地引用特定文件的办法。这可以用指定路径名的办法来实现。路径名是一串用斜杠字符 / 分隔的(目录或文件的)名称。如果这串字符用斜杠字符开头,则称为绝对路径名。它表示从根目录开始。例如:proj1 目录的绝对路径名是:

```
/home/lynne/proj1
```

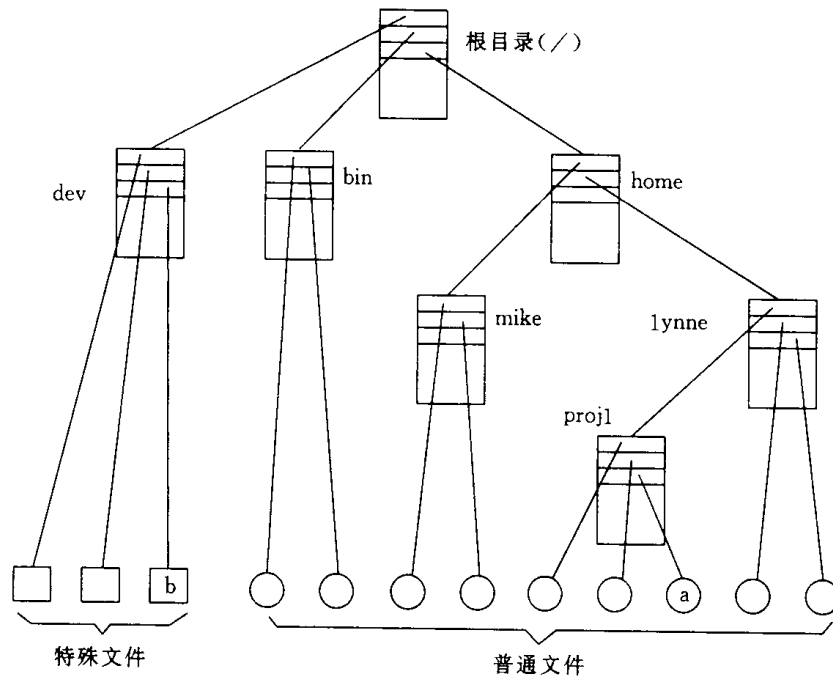


图 1.1 主要的文件类型形成树形的层次结构

而特殊文件 b 的绝对路径名则是：

```
/dev/b
```

正像你看到的,绝对路径名从根目录开始,并由此向下延伸。也可以用其他目录名作为开始来指定文件的路径名。用这种方法给出的路径名称为相对路径名。例如,从 lynne 目录(或严格地称为 /home/lynne 目录)到普通文件 a 的相对路径名是：

```
proj1/a
```

它表示在当前目录 /home/lynne 下,通过包含 a 文件的 proj1 目录取得 a 文件的路径。

当你改变目录时,新目录成为当前的工作目录。当前工作目录的概念相当重要,每当在 Linux 系统上执行一个程序时,作为结果的进程将当前工作目录设为它的内部状态的一部分。正是这个目录,进程将它用作访问任何文件的相对路径名的起点。除非进程为自己另设特殊的当前工作目录(多数情况下不这么做),一般情况下取用户启动程序时所处的目录作为默认的当前工作目录。

如何指定从 /home/mike 目录到 a 文件的相对路径名? 问题是 /home/mike 目录和文件 a 处在目录树的不同分支上。为解决这一问题,需要知道另外两个名称,这两个名称都自动地包含在每个 Linux 目录中。它们分别称为 . (发音为 dot) 和 .. (发音为 dot-dot)。
 . 是当前目录的别名。而 .. 则是当前目录的双亲目录的别名。在指定相对路径名时,.. 允许你回到目录树的上一层。有了 .. 的标记法,上面问题的答案就成为从 /home/mike 目录回到上一层目录 /home,然后从右面的分支向下到达 a 文件。得到的相对路径名如下：

```
../lynne/proj1/a
```


顺便说一下,虽然根目录没有上层目录,但仍然保留名称 .. ,它被用来指向根目录本身。根目录作特殊处理,它就是它自身的双亲目录。这也表示在指定相对路径名时,使用太多的 .. 不会造成问题。

1.3 考察文件系统的层次结构

在登录时,有一个特殊的目录和你的登录名联系在一起。它被称为你的起始(home)目录。起始目录实际是最初的当前工作目录,这是属于你的整个目录树的起点。你在系统工作时所建立的所有文件和目录,一般都放在你的起始目录下面。

在整个系统的目录层次结构中,找出起始目录的实际位置的最简单办法是在登录后直接使用 pwd 命令。pwd 命令告诉你:当前工作目录,也就是现在所处的目录是什么。在刚刚登录时,它将是起始目录。在起始目录下使用 pwd 命令将得到类似下面的输出:

```
$ pwd
/home/you
```

其中 /home/you 应该用你自己的起始目录的绝对路径名代替。

为了将当前工作目录从起始目录转换到目录层次结构中的其他位置,可以用 cd(改变目录)命令。所以,要将目录改到根目录就应该用下面的命令:

```
$ cd /
```

在 cd 命令中用所要的目标目录的路径名作为参数。

为了得到包含在当前目录下的文件和目录名称的清单,可以使用 ls 命令。对一个典型的 Linux 系统讲,在根目录下使用 ls 命令将得到下面的输出:

```
$ ls
bin      dev      home     mnt      sbin     var
boot    dos      lib      proc     tmp      vmlinuz
cdrom   etc      lost+found root     usr
```

Linux 目录层次结构中最重要分支表示在图 1.2 中。应该用 pwd, cd 和 ls 命令去考察你自己机器上的目录层次结构。

如果想从目录树中任何一点回到起始目录,可以用 cd 命令。并用相应的路径名作为参数。但是,也可以只送入 cd 命令而不给参数。这是使直接回到起始目录的一条捷径。在图 1.2 所示的 Linux 目录树中,各主要分支包含的文件的一般功能见表 1.1。

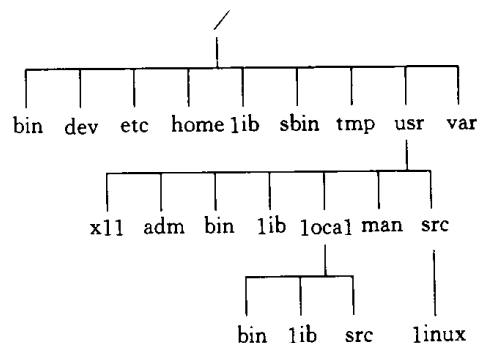


图 1.2 主要的 Linux 目录树结构