TURING 图灵程序设计丛书

第2版

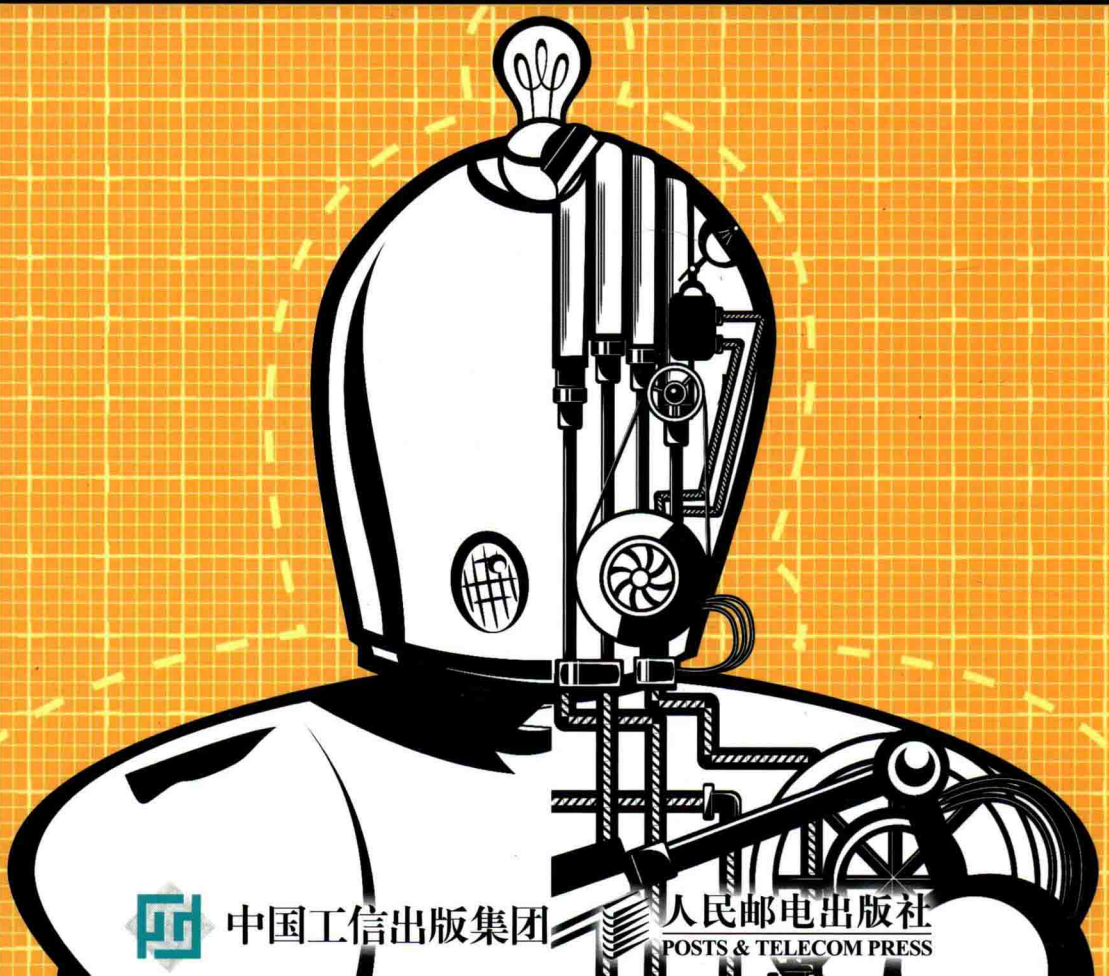
How Linux Works

2nd Edition

What Every Superuser Should Know

精通Linux

【美】Brian Ward 著　姜南 袁志鹏 译



中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

How Linux Works

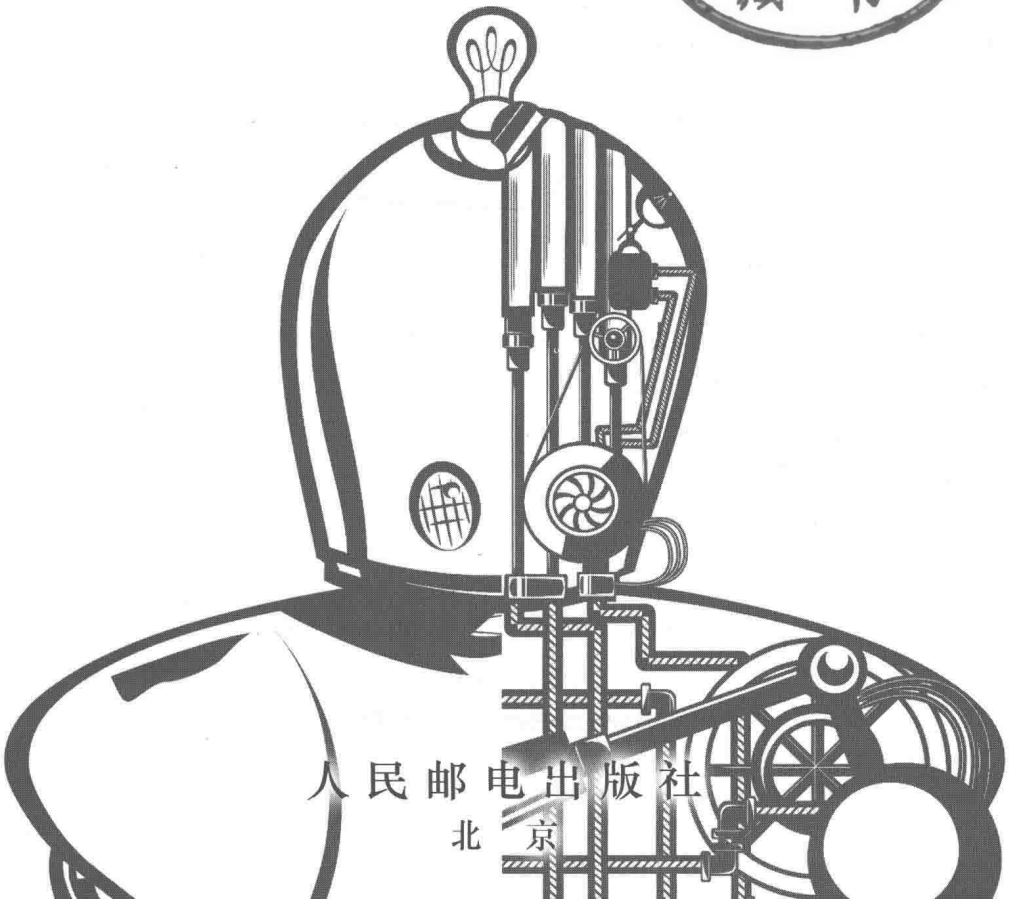
2nd Edition

What Every Superuser Should Know

精通Linux

第2版

【美】Brian Ward 著 姜南 袁志鹏 译



人民邮电出版社
北京

图书在版编目（CIP）数据

精通Linux : 第2版 / (美) 沃德 (Ward,B.) 著 ; 姜南, 袁志鹏译. -- 北京 : 人民邮电出版社, 2015.7
(图灵程序设计丛书)
ISBN 978-7-115-39492-7

Ⅰ. ①精… Ⅱ. ①沃… ②姜… ③袁… Ⅲ. ①Linux操作系统 Ⅳ. ①TP316.89

中国版本图书馆CIP数据核字(2015)第122015号

内 容 提 要

本书讲解了 Linux 操作系统的工作机制以及运行 Linux 系统所需的常用工具和命令。根据系统启动的大体顺序，本书更深入地介绍从设备管理到网络配置的各个部分，最后演示了系统各部分的运行方式，并介绍了一些基本技巧和开发人员常用的工具。

本书适合对 Linux 感兴趣的初学者阅读。

◆ 著 [美] Brian Ward
译 姜 南 袁志鹏
责任编辑 岳新欣
执行编辑 张 曼
责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：800×1000 1/16
印张：19
字数：449千字 2015年7月第1版
印数：1-4 000册 2015年7月河北第1次印刷
著作权合同登记号 图字：01-2014-8584号

定价：59.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第 0021 号

前　言

之所以写这本书，是因为我觉得你应该对你使用的计算机有所了解。你应该可以让软件去做你想让它去做的事（当然要在它的能力范围之内）。要做到这一点，关键是必须理解软件能做什么，以及是怎么做的。这些正是本书要介绍的内容。这样你就不必对着计算机抓狂了。

如果你要学习这方面的知识，Linux是一个很好的平台，因为它是一个透明的系统。特别是大多数系统配置都存放在文本文件中，让人一目了然。难点在于了解每个组件分别负责什么，以及它们如何协同工作。

读者对象

我们学习Linux的原因可能各不相同。对于IT从业者（如系统运维人员）来说，他们需要了解本书中的几乎所有内容。对于Linux软件架构师和开发人员来说，他们同样需要了解这些内容，以便发挥操作系统的最大功效。对于只需考虑个人所用Linux系统的研究人员和学生来说，本书能够让他们理解为什么系统是那个样子的。

还有一些堪称多面手的读者，出于兴趣、谋利或其他原因而摆弄计算机，喜欢探究事情的根源，喜欢尝试不同的可能性。或许你就是其中一位。

阅读要求

虽然开发人员热爱Linux，不过并非开发人员才能阅读本书，只要你有一些基础的计算机知识即可。也就是说，你需要知道如何操作GUI（特别是能看懂各种Linux发行版的安装和配置界面），需要知道什么是文件什么是目录（文件夹）。此外，还要有心理准备随时查看系统文档或者上网搜索一些相关文章。当然，我前面提过，最重要的是你对电脑的热情。

阅读方法

要对任何技术建立起系统的认识都不是件容易的事。而说到软件系统的工作原理，就更复杂了。有时候面对大量的技术细节，读者会难以抓住重点（因为人类大脑无法同时处理太多新概念），但是如果解释得不够透彻，又会让读者一知半解，不利于后面的学习。

本书每一章都会先介绍最重要最基本的知识，以便让读者能够继续深入。为了突出重点，有

些地方简化了很多内容。随着一章内容的展开，更多细节才会在最后几节出现。这些内容需要你马上就掌握吗？通常不用，我经常也会这么提醒你。如果你觉得正在阅读的内容有些枯燥难懂，可以随时跳到下一章或者稍事休息。

动手操作

你最好准备一台可以用来实际操作的Linux计算机。你可以使用虚拟机，比如我就使用VirtualBox来测试本书中的很多实例。你需要拥有超级用户（root）权限，不过多数情况下你需要以普通用户身份登录系统。我们将主要通过终端窗口或者远程会话来运行命令行。如果你之前毫无经验也无大碍，书中第2章会让你尽快上手。

书中的命令通常是像下面这样：

```
$ ls /
[输出结果]
```

你只需输入第一行粗体的文本，非粗体的文本是系统的输出结果。$是普通用户提示符。如果你是超级用户的话则是#。（详见第2章。）

本书结构

本书分为三个部分。第一部分整体介绍Linux系统以及运行Linux系统所需的常用工具和命令。随后我们会根据系统启动的大体顺序，更深入地介绍从设备管理到网络配置的各个部分。最后我们会演示系统各部分的运行方式，并介绍一些基本技巧和开发人员常用的工具。

除第2章以外，开始的几章均主要讲解Linux内核，然后逐步涉及用户空间。（如果你现在对我所说的一头雾水也没关系，我们将在第1章中介绍这些概念。）

本书的内容尽量保证对各个版本的Linux系统均适用。但要涵盖各个系统之间的差异也实在是项繁琐的工作，所以我尽量考虑两个主要的Linux版本：Debian（包括Ubuntu）和RHEL/Fedora/CentOS。本书主要针对的是桌面和服务器系统。嵌入式系统（如Android和OpenWRT）也多有涉及，但各系统之间的差异还需要你自己去探索。

第 2 版的新内容

本书的第1版侧重于从用户的角度来介绍Linux系统，旨在帮助读者了解系统各部分的工作原理。彼时Linux上的软件安装和配置还不是那么容易。

有幸的是，随着各种新版本的出现，这些问题已然不复存在，所以我剔除了一些较为陈旧和不太相关的内容（比如打印），以便能够更加深入地介绍Linux内核。你可能没有意识到你将会多么频繁地和内核打交道。

当然，上一版中的很多内容随着时间推移也发生了较大变化，我花了大量的精力梳理和更新

了它们，特别是在Linux的启动和设备管理方面。我对很多内容也进行了重新组织，以满足当下读者的阅读兴趣与需要。

本书没有发生变化的是它的厚度。我希望读者能够尽快上手，因此会解释一些不太容易理解的细节，但我又不想让这本书变得你拿都拿不动。只要掌握了本书介绍的知识，自己再去深入探索就不是一件难事了。

我还删掉了第1版中一些关于历史背景的介绍，目的是突出重点。如果你对Linux和Unix的历史感兴趣，可以参考Peter H. Salus所著*The Daemon, the Gnu, and the Penguin*（Reed Media Services，2008），这本书详细介绍了我们使用的各种软件的历史沿革。

关于术语

关于操作系统中某些组件应该叫什么，一直都存在争论。甚至“Linux”是否应该叫作“GNU/Linux”也存在争论，因为其中使用了GNU项目的成果。本书中我们尽量使用通用术语，不使用拗口、生硬的词汇。

致　　谢

感谢以下对本书的第 1 版提供过帮助的人：James Duncan、Douglas N. Arnold、Bill Fenner、Ken Hornstein、Scott Dickson、Dan Ehrlich、Felix Lee、Scott Schwartz、Gregory P. Smith、Dan Sully、Karol Jurado 以及 Gina Steele。在第 2 版的写作中，我要特别感谢 Jordi Gutiérrez Hermoso 卓越的技术审阅工作，他为本书提供了极有价值的建议和勘误。还要感谢 Dominique Poulain 和 Donald Karon，他们在本书写作期间就给出了非常好的反馈意见，还要感谢 Hsinju Hsieh 在我写作这本书期间对我的宽容。

最后，我还要感谢本书策划编辑 Bill Pollock、项目编辑 Laurel Chun，以及 No Starch Press 出版社的 Serena Yang、Alison Law 和其他人为本书面世所做的一如既往的卓越工作。

第 1 版书评

“非常棒的书。在近 350 页的内容中涵盖了 Linux 的所有基础知识。”

——EWEEK

“对于那些想要学习 Linux，同时对操作系统内部工作原理又不太熟悉的读者，本书绝对值得推荐。”

——O’REILLYNET

“介绍 Linux 基础知识最好的书之一，同时也适合 Linux 高级用户阅读，五星。”

——OPENSOURCE-BOOK-REVIEWS.COM

“本书的成功源于它对内容的良好组织和对技术细节的深入探讨。”

——KICKSTART NEWS

“本书对 Linux 的介绍可谓标新立异。它朴实无华，注重对命令行的介绍，并且深入到系统内部，而非仅仅停留在图形用户界面。”

——TECHBOOKREPORT.COM

“本书很好地介绍了 Linux 系统的工作原理。”

——HOSTING RESOLVE

目　录

第 1 章　概述 ……1
1.1　Linux 操作系统中的抽象级别和层次 ……2
1.2　硬件系统：理解主内存 ……3
1.3　内核 ……3
1.3.1　进程管理 ……4
1.3.2　内存管理 ……5
1.3.3　设备驱动程序和设备管理 ……5
1.3.4　系统调用和系统支持 ……5
1.4　用户空间 ……6
1.5　用户 ……7
1.6　前瞻 ……8
第 2 章　基础命令和目录结构 ……9
2.1　Bourne shell: /bin/sh ……9
2.2　shell 的使用 ……10
2.2.1　shell 窗口 ……10
2.2.2　cat 命令 ……11
2.2.3　标准输入输出 ……11
2.3　基础命令 ……11
2.3.1　ls 命令 ……12
2.3.2　cp 命令 ……12
2.3.3　mv 命令 ……12
2.3.4　touch 命令 ……13
2.3.5　rm 命令 ……13
2.3.6　echo 命令 ……13
2.4　浏览目录 ……13
2.4.1　cd 命令 ……14
2.4.2　mkdir 命令 ……14
2.4.3　rmdir 命令 ……14
2.4.4　shell 通配符 ……14
2.5　中间命令 ……15
2.5.1　grep 命令 ……15
2.5.2　less 命令 ……16
2.5.3　pwd 命令 ……16
2.5.4　diff 命令 ……16
2.5.5　file 命令 ……17
2.5.6　find 和 locate 命令 ……17
2.5.7　head 和 tail 命令 ……17
2.5.8　sort 命令 ……17
2.6　更改密码和 shell ……18
2.7　dot 文件 ……18
2.8　环境变量和 shell 变量 ……18
2.9　命令路径 ……19
2.10　特殊字符 ……19
2.11　命令行编辑 ……20
2.12　文本编辑器 ……21
2.13　获取在线帮助 ……21
2.14　shell 输入输出 ……23
2.14.1　标准错误输出 ……23
2.14.2　标准输入重定向 ……24
2.15　理解错误信息 ……24
2.15.1　解析 Unix 的错误信息 ……24
2.15.2　常见错误 ……25
2.16　查看和操纵进程 ……26
2.16.1　命令选项 ……26
2.16.2　终止进程 ……27
2.16.3　任务控制 ……27
2.16.4　后台进程 ……28
2.17　文件模式和权限 ……28
2.17.1　更改文件权限 ……29
2.17.2　符号链接 ……30
2.17.3　创建符号链接 ……30
2.18　归档和压缩文件 ……31
2.18.1　gzip 命令 ……31

2.18.2 tar 命令 ····· 31
2.18.3 压缩归档文件（.tar.gz） ····· 32
2.18.4 zcat 命令 ····· 32
2.18.5 其他的压缩命令 ····· 33
2.19 Linux 目录结构基础 ····· 33
2.19.1 root 目录下的其他目录 ····· 34
2.19.2 /usr 目录 ····· 35
2.19.3 内核位置 ····· 35
2.20 以超级用户的身份运行命令 ····· 35
2.20.1 sudo 命令 ····· 35
2.20.2 /etc/sudoers ····· 35
2.21 前瞻 ····· 36

第 3 章 设备管理 ····· 37

3.1 设备文件 ····· 37
3.2 sysfs 设备路径 ····· 38
3.3 dd 命令和设备 ····· 39
3.4 设备名总结 ····· 40
3.4.1 硬盘：/dev/sd* ····· 40
3.4.2 CD 和 DVD：/dev/sr* ····· 41
3.4.3 PATA 硬盘：/dev/hd* ····· 41
3.4.4 终端设备/dev/tty/*、/dev/pts/* 和/dev/tty ····· 41
3.4.5 串行端口：/dev/ttyS* ····· 42
3.4.6 并行端口：/dev/lp0 和/dev/lp1 ····· 42
3.4.7 音频设备：/dev/snd/*、/dev/dsp、/dev/audio 和其他 ····· 43
3.4.8 创建设备文件 ····· 43
3.5 udev ····· 44
3.5.1 devtmpfs ····· 44
3.5.2 udevd 的操作和配置 ····· 44
3.5.3 udevadm ····· 46
3.5.4 设备监控 ····· 47
3.6 详解 SCSI 和 Linux 内核 ····· 47
3.6.1 USB 存储设备和 SCSI ····· 50
3.6.2 SCSI 和 ATA ····· 50
3.6.3 通用 SCSI 设备 ····· 51
3.6.4 访问设备的多种方法 ····· 51

第 4 章 硬盘和文件系统 ····· 53

4.1 为磁盘设备分区 ····· 55
4.1.1 查看分区表 ····· 55
4.1.2 更改分区表 ····· 56
4.1.3 磁盘和分区的构造 ····· 57
4.1.4 固态硬盘 ····· 58
4.2 文件系统 ····· 59
4.2.1 文件系统类型 ····· 59
4.2.2 创建文件系统 ····· 60
4.2.3 挂载文件系统 ····· 60
4.2.4 文件系统 UUID ····· 62
4.2.5 磁盘缓冲、缓存和文件系统 ····· 62
4.2.6 文件系统挂载选项 ····· 63
4.2.7 重新挂载文件系统 ····· 64
4.2.8 /etc/fstab 文件系统表 ····· 64
4.2.9 /etc/fstab 的替代者 ····· 65
4.2.10 文件系统容量 ····· 65
4.2.11 检查和修复文件系统 ····· 66
4.2.12 特殊用途的文件系统 ····· 68
4.3 交换空间 ····· 68
4.3.1 使用磁盘分区作为交换空间 ····· 69
4.3.2 使用文件作为交换空间 ····· 69
4.3.3 你需要多大的交换空间 ····· 69
4.4 前瞻：磁盘和用户空间 ····· 70
4.5 深入传统文件系统 ····· 70
4.5.1 查看 inode 细节 ····· 72
4.5.2 在用户空间中使用文件系统 ····· 73
4.5.3 文件系统的演进 ····· 73

第 5 章 Linux 内核的启动 ····· 75

5.1 启动消息 ····· 75
5.2 内核初始化和启动选项 ····· 76
5.3 内核参数 ····· 77
5.4 引导装载程序 ····· 78
5.4.1 引导装载程序任务 ····· 78
5.4.2 引导装载程序概述 ····· 79
5.5 GRUB 简介 ····· 79
5.5.1 使用 GRUB 命令行浏览设备和分区 ····· 81
5.5.2 GRUB 配置信息 ····· 83
5.5.3 安装 GRUB ····· 84
5.6 UEFI 安全启动的问题 ····· 86
5.7 链式加载其他操作系统 ····· 86
5.8 引导装载程序细节 ····· 86
5.8.1 MBR 启动 ····· 87

5.8.2 UEFI 启动 ……87
5.8.3 GRUB 工作原理 ……87

第 6 章 用户空间的启动 ……89

6.1 init 介绍 ……89
6.2 System V 运行级别 ……90
6.3 识别你的 init ……91
6.4 systemd ……91
6.4.1 单元和单元类型 ……91
6.4.2 systemd 中的依赖关系 ……92
6.4.3 systemd 配置 ……94
6.4.4 systemd 操作 ……96
6.4.5 在 systemd 中添加单元 ……98
6.4.6 systemd 进程跟踪和同步 ……99
6.4.7 systemd 的按需和资源并行启动 ……99
6.4.8 systemd 的 System V 兼容性 ……103
6.4.9 systemd 辅助程序 ……103
6.5 Upstart ……104
6.5.1 Upstart 初始化过程 ……104
6.5.2 Upstart 任务 ……105
6.5.3 Upstart 配置 ……107
6.5.4 Upstart 操作 ……110
6.5.5 Upstart 日志 ……111
6.5.6 Upstart 运行级别和 System V 兼容性 ……111
6.6 System V init ……112
6.6.1 System V init 启动命令顺序 ……113
6.6.2 System V init 链接池 ……114
6.6.3 run-parts ……115
6.6.4 System V init 控制 ……115
6.7 关闭系统 ……116
6.8 initramfs ……117
6.9 紧急启动和单用户模式 ……118

第 7 章 系统配置：日志、系统时间、批处理任务和用户 ……119

7.1 /etc 目录结构 ……119
7.2 系统日志 ……120
7.2.1 系统日志 ……120
7.2.2 配置文件 ……120
7.3 用户管理文件 ……122
7.3.1 /etc/passwd 文件 ……122
7.3.2 特殊用户 ……123
7.3.3 /etc/shadow 文件 ……124
7.3.4 用户和密码管理 ……124
7.3.5 用户组 ……124
7.4 getty 和 login ……125
7.5 设置时间 ……125
7.5.1 内核时间和时区 ……126
7.5.2 网络时间 ……127
7.6 使用 cron 来调度日常任务 ……127
7.6.1 安装 crontab 文件 ……128
7.6.2 系统 crontab 文件 ……128
7.6.3 cron 的未来 ……129
7.7 使用 at 进行一次性任务调度 ……129
7.8 了解用户 ID 和用户切换 ……129
7.9 用户标识和认证 ……131
7.10 PAM ……132
7.10.1 PAM 配置 ……133
7.10.2 关于 PAM 的一些注解 ……135
7.10.3 PAM 和密码 ……136
7.11 前瞻 ……136

第 8 章 进程与资源利用详解 ……137

8.1 进程跟踪 ……137
8.2 使用 lsof 查看打开的文件 ……138
8.2.1 lsof 输出 ……138
8.2.2 lsof 的使用 ……139
8.3 跟踪程序执行和系统调用 ……139
8.3.1 strace 命令 ……139
8.3.2 ltrace 命令 ……141
8.4 线程 ……141
8.4.1 单线程进程和多线程进程 ……141
8.4.2 查看线程 ……142
8.5 资源监控简介 ……143
8.6 测量 CPU 时间 ……143
8.7 调整进程优先级 ……144
8.8 平均负载 ……145
8.8.1 uptime 的使用 ……145
8.8.2 高负载 ……145
8.9 内存 ……146
8.9.1 内存工作原理 ……146
8.9.2 内存页面错误 ……146

8.10 使用 vmstat 监控 CPU 和内存性能……147
8.11 I/O 监控……149
8.11.1 使用 iostat……149
8.11.2 使用 iotop 查看进程的 I/O 使用和监控……150
8.12 使用 pidstat 监控进程……151
8.13 更深入的主题……151
第 9 章 网络与配置……153
9.1 网络基础……153
9.2 网络层次……154
9.3 网际层……155
9.3.1 查看自己计算机的 IP 地址……156
9.3.2 子网……157
9.3.3 共用子网掩码与无类域内路由选择……157
9.4 路由和内核路由表……158
9.5 基本 ICMP 和 DNS 工具……159
9.5.1 ping……159
9.5.2 traceroute……160
9.5.3 DNS 与 host……160
9.6 物理层与以太网……161
9.7 理解内核网络接口……161
9.8 配置网络接口……162
9.9 开机启动的网络配置……163
9.10 手动和开机启动的网络配置带来的问题……163
9.11 一些网络配置管理器……164
9.11.1 NetworkManager 的操作……164
9.11.2 与 NetworkManager 交互……164
9.11.3 NetworkManager 的配置……165
9.12 解析主机名……166
9.12.1 /etc/hosts……167
9.12.2 resolv.conf 文件……167
9.12.3 缓存和零配置 DNS……167
9.12.4 /etc/nsswitch.conf 文件……168
9.13 Localhost……168
9.14 传输层：TCP、UDP 和 Service……169
9.14.1 TCP 端口与连接……169
9.14.2 建立 TCP 连接……169
9.14.3 端口的数字和/etc/services……170
9.14.4 TCP 的特点……171
9.14.5 UDP……171
9.15 普通本地网络……172
9.16 理解 DHCP……173
9.16.1 Linux 的 DHCP 客户端……173
9.16.2 Linux 的 DHCP 服务器……173
9.17 将 Linux 配置成路由器……174
9.18 私有网络……175
9.19 网络地址转换（IP 伪装）……176
9.20 路由器与 Linux……177
9.21 防火墙……177
9.21.1 Linux 防火墙基础……178
9.21.2 设置防火墙规则……179
9.21.3 防火墙策略……181
9.22 以太网、IP 和 ARP……182
9.23 无线以太网……183
9.23.1 iw……184
9.23.2 无线网络安全……184
9.24 小结……185
第 10 章 网络应用与服务……186
10.1 服务的基本概念……186
10.2 网络服务器……188
10.3 SSH……189
10.3.1 SSHD 服务器……190
10.3.2 SSH 客户端……191
10.4 守护进程 inetd 和 xinetd……193
10.5 诊断工具……193
10.5.1 lsof……194
10.5.2 tcpdump……195
10.5.3 netcat……196
10.5.4 扫描端口……197
10.6 远程程序调用……198
10.7 网络安全……198
10.7.1 典型漏洞……199
10.7.2 安全资源……199
10.8 前瞻……200
10.9 套接字：进程与网络的通信方式……200
10.10 Unix 域套接字……201
10.10.1 对开发者的好处……201
10.10.2 列出 Unix 域套接字……202

第 11 章　shell 脚本 ……203
11.1　shell 脚本基础 ……203
11.2　引号与字面量 ……204
11.2.1　字面量 ……205
11.2.2　单引号 ……205
11.2.3　双引号 ……205
11.2.4　单引号的字面义 ……206
11.3　特殊变量 ……206
11.3.1　单个参数：$1，$2，…… ……207
11.3.2　参数的数量：$# ……207
11.3.3　所有参数：$@ ……207
11.3.4　脚本名：$0 ……208
11.3.5　进程号：$$ ……208
11.3.6　退出码：$? ……208
11.4　退出码 ……208
11.5　条件判断 ……209
11.5.1　防范空参数 ……209
11.5.2　使用其他命令来测试 ……210
11.5.3　elif ……210
11.5.4　逻辑结构&&和|| ……210
11.5.5　测试条件 ……211
11.5.6　用 case 进行字符串匹配 ……213
11.6　循环 ……214
11.6.1　for 循环 ……214
11.6.2　while 循环 ……214
11.7　命令替换 ……215
11.8　管理临时文件 ……216
11.9　here 文档 ……216
11.10　重要的 shell 脚本工具 ……217
11.10.1　basename ……217
11.10.2　awk ……218
11.10.3　sed ……218
11.10.4　xargs ……219
11.10.5　expr ……219
11.10.6　exec ……219
11.11　子 shell ……220
11.12　在脚本中包含其他文件 ……220
11.13　读取用户输入 ……221
11.14　什么时候（不）应该使用 shell 脚本 ……221

第 12 章　在网络上传输文件 ……222
12.1　快速复制 ……222
12.2　rsync ……222
12.2.1　rsync 基础 ……223
12.2.2　准确复制目录结构 ……224
12.2.3　以斜杠结尾 ……224
12.2.4　排除文件与目录 ……226
12.2.5　合并、检查及冗长模式 ……226
12.2.6　压缩 ……227
12.2.7　限制带宽 ……227
12.2.8　传文件到你的计算机 ……227
12.2.9　更多有关 rsync 的话题 ……227
12.3　文件共享 ……228
12.4　用 Samba 分享文件 ……228
12.4.1　配置服务器 ……228
12.4.2　服务器访问控制 ……229
12.4.3　密码 ……229
12.4.4　启动服务器 ……231
12.4.5　诊断和日志文件 ……231
12.4.6　配置文件共享 ……231
12.4.7　home 目录 ……232
12.4.8　共享打印机 ……232
12.4.9　使用 Samba 客户端 ……232
12.4.10　作为客户去访问文件 ……233
12.5　NFS 客户端 ……234
12.6　有关网络文件服务的选择与局限的更多内容 ……234

第 13 章　用户环境 ……235
13.1　创建启动文件的规则 ……235
13.2　何时需要修改启动文件 ……236
13.3　shell 启动文件的元素 ……236
13.3.1　命令路径 ……236
13.3.2　帮助手册的路径 ……237
13.3.3　提示符 ……237
13.3.4　别名 ……238
13.3.5　权限掩码 ……238
13.4　启动文件的顺序及例子 ……238
13.4.1　bash shell ……239
13.4.2　tcsh shell ……241
13.5　用户默认设置 ……241

13.5.1 shell默认设置……242
13.5.2 编辑器……242
13.5.3 翻页器……242
13.6 启动文件的一些陷阱……242
13.7 前瞻……243

第14章 Linux桌面概览……244

14.1 桌面组件……244
14.1.1 窗口管理器……245
14.1.2 工具包……245
14.1.3 桌面环境……245
14.1.4 应用……245
14.2 近观X Window系统……245
14.2.1 显示管理器……246
14.2.2 网络透明性……246
14.3 探索X客户端……247
14.3.1 X事件……247
14.3.2 理解X输入以及偏好设定……248
14.4 X的未来……250
14.5 D-Bus……250
14.5.1 系统和会话实例……251
14.5.2 监视D-Bus消息……251
14.6 打印……251
14.6.1 CUPS……252
14.6.2 格式转换与打印过滤器……252
14.7 其他有关桌面的话题……253

第15章 开发工具……254

15.1 C编译器……254
15.1.1 多个源码文件……255
15.1.2 头（include）文件和目录……256
15.1.3 连接库……257
15.1.4 共享库……258
15.2 make……261
15.2.1 一个Makefile实例……261
15.2.2 内置规则……262
15.2.3 最终的程序构建……262
15.2.4 保持更新……263
15.2.5 命令行参数与选项……263
15.2.6 标准宏和变量……264
15.2.7 常规的目标……264
15.2.8 组织一个Makefile……265
15.3 调试器……266
15.4 Lex和Yacc……267
15.5 脚本语言……267
15.5.1 Python……268
15.5.2 Perl……268
15.5.3 其他脚本语言……268
15.6 Java……269
15.7 展望：编译包……270

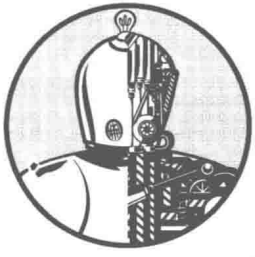
第16章 从C代码编译出软件……271

16.1 软件的构建系统……271
16.2 解开C源码包……272
16.3 GNU autoconf……273
16.3.1 一个autoconf的例子……274
16.3.2 使用打包工具来安装……275
16.3.3 configure脚本的选项……275
16.3.4 环境变量……276
16.3.5 autoconf的目标……277
16.3.6 autoconf的日志文件……277
16.3.7 pkg-config……277
16.4 实践安装……278
16.5 打补丁……279
16.6 编译和安装的问题排查……280
16.7 前瞻……282

第17章 在基础上搭建……284

17.1 Web服务器与应用……284
17.2 数据库……285
17.3 虚拟化……285
17.4 分布式计算与实时计算……286
17.5 嵌入式系统……286
17.6 结束语……287

第 1 章 概　述



乍看起来，Linux这样的现代操作系统非常复杂，内部有多得令人眼花缭乱的各种组件在同步运行和相互通信。比如：Web服务器可以连接到数据库服务器，还有可能用到很多其他程序也在使用的公共组件。那么，整个系统究竟是怎样运作的呢？

理解操作系统工作原理最好的方法是抽象思维，换句话说，你可以暂时忽略大部分细节。就像坐车一样，通常你不会去在意车内固定发动机的装配螺栓，也不会关心你走的路是谁修筑的。如果你是一个乘客的话，你可能只关心车要做的事情（比如车要把你带到哪）以及车的一些基本操作（比如如何打开车门、怎样系好安全带）。

但如果你在开车的话，就需要了解更多的细节，比如如何控制油门、怎样换挡，还有如何处理意外情况。

如果我们觉得开车这个事情太复杂，就可以运用“抽象思维”来帮助理解。首先你可以将“一辆汽车在路上行驶”抽象为三个部分：汽车、道路和驾驶操作。这样有助于将复杂的问题分解开来。如果道路颠簸，你不会去埋怨车辆本身和你的驾驶技术。相反，你可能会问为什么这条路这么烂，或者如果这是条新修的路的话，那么筑路工人的活干得可真够差劲的。

软件开发人员运用抽象思维来开发操作系统和应用程序。在计算机软件领域有许多术语来描述抽象的子系统，如子系统、模块和包等。本书中我们使用组件这个相对简单的词。在软件开发过程中，开发人员通常不用太关心他们需要使用的组件的内部结构，他们只关心能使用哪些组件，以及怎么个用法。

本章概述了Linux操作系统涉及的主要组件。虽然每一个组件都包含纷繁复杂的技术细节，但我们将暂时忽略这些细节，而专注于这些组件在系统中发挥的功能。

1.1　Linux 操作系统中的抽象级别和层次

在组织得当的前提下，通过抽象将系统分解为组件有助于我们了解其工作机制。我们将组件划分为*层次*或*级别*。组件的层次（或级别）代表它在用户和硬件系统之间所处的位置。Web浏览器、游戏等应用处于最高层，底层则是计算机硬件系统，如内存。操作系统处于这两层之间。

Linux操作系统主要分为三层。如图1-1所示，最底层是硬件系统，包括内存和中央处理器（用于计算和从内存中读写数据），此外硬盘和网络接口也是硬件系统的一部分。

硬件系统之上是*内核*，它是操作系统的核心。内核是运行在内存中的软件，它向中央处理器发送指令。内核管理硬件系统，是硬件系统和应用程序之间进行通信的接口。

*进程*是指计算机中运行的所有程序，由内核统一管理，它们组成了最顶层，称为*用户空间*。（另一个更确切的术语是用户进程，无论它们是否直接和用户交互。例如，所有的Web服务器都是以用户进程的形式运行的。）

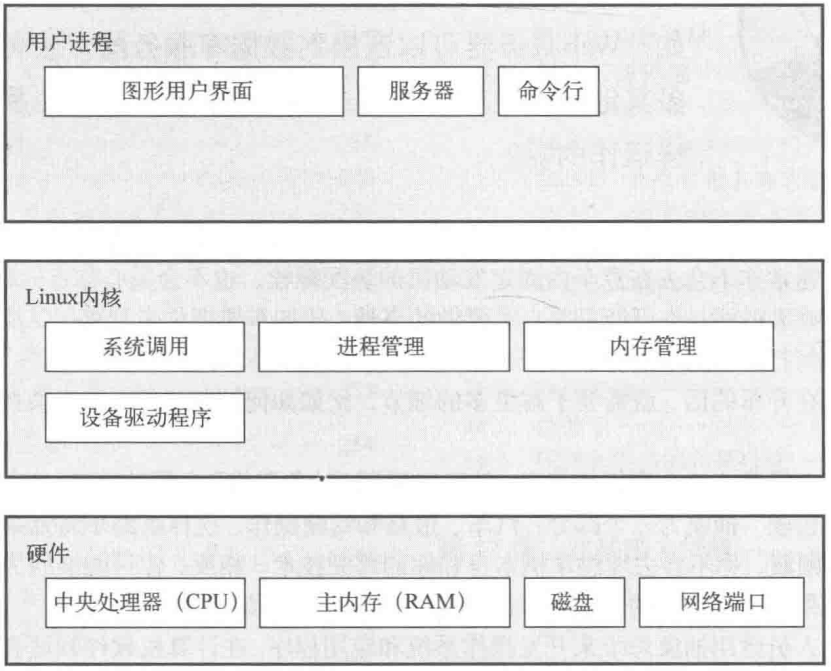


图1-1　Linux系统的基本组成

内核和用户进程之间最主要的区别是：内核在*内核模式*（kernel mode）中运行，而用户进程则在*用户模式*（user mode）中运行。在内核模式中运行的代码可以不受限地访问中央处理器和内存，这种模式功能强大，但也非常危险，因为内核进程可以轻而易举地使整个系统崩溃。那些只有内核可以访问的空间我们称为*内核空间*（kernel space）。

相对于内核模式，用户模式对内存和中央处理器的访问有一定程度的限制，可访问的内存空

间通常很小，对CPU的操作也很安全。*用户空间*指的是那些用户进程能够访问的内存空间。如果一个用户进程出错并崩溃的话，其导致的后果也相对有限，并且能够被内核清理掉。例如，如果你的Web浏览器崩溃了，不会影响到你正在运行的其他程序。

理论上来说，一个用户进程出问题并不会对整个系统造成严重的影响。当然这取决于我们如何定义“严重的影响”，并且还取决于该进程拥有的权限。因为不同的进程拥有的权限可能不同，一些进程能够执行一些别的进程无权执行的操作。举个例子，如果拥有足够的权限，用户进程可以将硬盘上的数据全部清除。也许你会觉得这样太危险，但好在操作系统提供了一些相关的安全措施，而且大多数用户进程并没有这个权限。

1.2 硬件系统：理解主内存

主内存（main memory）或许是所有硬件系统中最为重要的部分。基本上来讲，主内存存储0和1这样的数据。我们将每个0和1称为一个比特（或位，bit）。内核和进程就在主内存中运行，它们就是一系列比特的大合集。所有外围设备的数据输入和输出都通过主内存完成，同样是以一系列0和1的形式。中央处理器像一个操作员一样处理内存中的数据，它从内存读取指令和数据，然后将运算结果写回内存。

在我们谈论内存、进程、内核和其他内容时，你会经常看到*状态*（state）这个词。严格说来，一个状态就是一组特定排列的比特。例如，内存中0110、0001和1011这三组比特值即表示三个不同的状态。

一个进程动辄由几百万个比特值组成，因而使用抽象词汇来描述状态可能比使用比特值更简单一些。我们可以使用进程已经完成的任务或者当前正在执行的任务来描述其状态，如“进程正在等待用户输入”或者“进程正在执行启动任务的第二个阶段”。

注解 *我们通常使用抽象词汇而非比特值来描述状态，映像*（image）*这个词用来表示比特值在内存中的特定物理排列。*

1.3 内核

我们之所以介绍主内存和状态，是因为内核的几乎所有操作都和主内存相关。其中之一是将内存划分为很多区块，并且一直维护着这些区块的状态信息。每一个进程拥有自己的内存区块，且内核必须确保每个进程只使用它自己的内存区块。

内核负责管理以下四个方面。

- *进程*：内核决定哪个进程可以使用CPU。
- *内存*：内核管理所有的内存，为进程分配内存，管理进程间的共享内存以及空闲内存。
- *设备驱动程序*：作为硬件系统（如磁盘）和进程之间的接口，内核负责操控硬件设备。