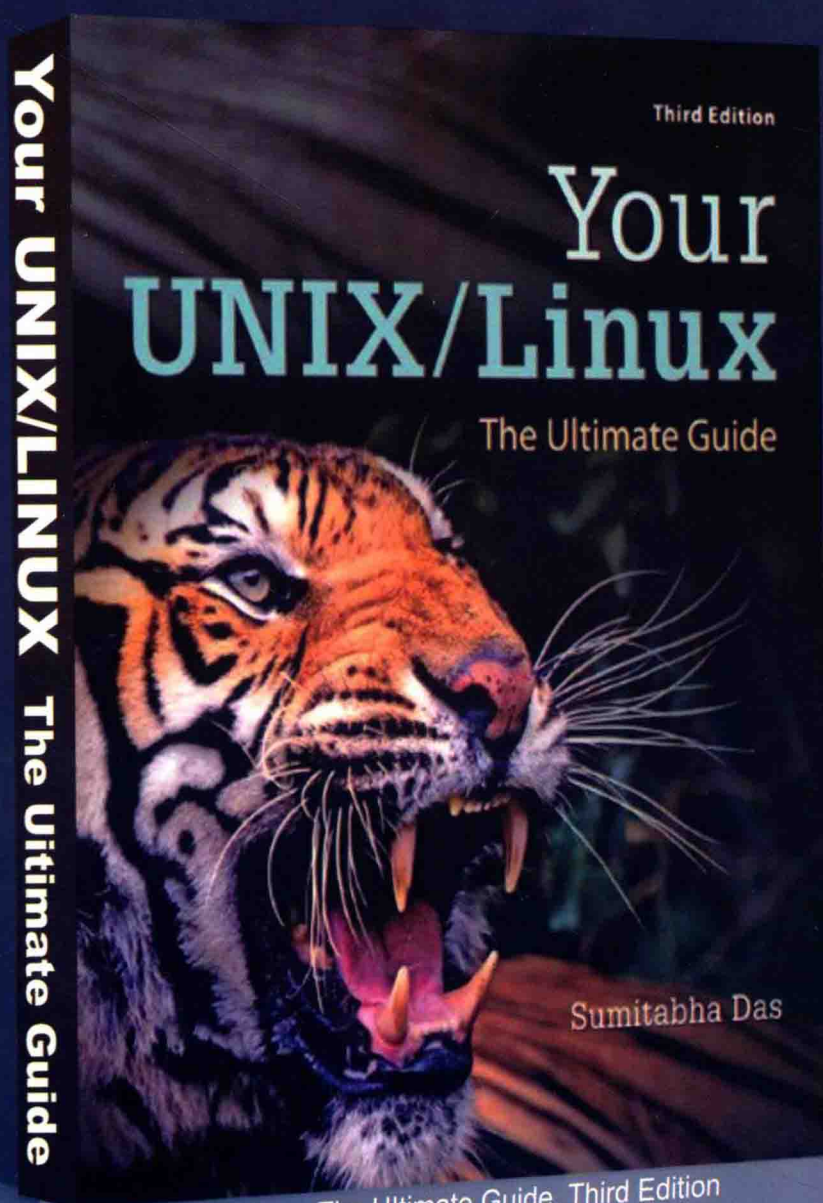


UNIX/Linux应用、 编程与系统管理 (第3版)

[美] Sumitabha Das 著
贾洪峰 李 莉 译



国外计算机科学经典教材

UNIX/Linux

应用、编程与系统管理

(第3版)

[美] Sumitabha Das 著

贾洪峰 李 莉 译

清华大学出版社

北 京

Sumitabha Das
Your UNIX/Linux: The Ultimate Guide, Third Edition
ISBN: 0-07-337620-5
Copyright © 2013 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education (Asia) and Tsinghua University Press Limited. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2014 by McGraw-Hill Education (Asia), a division of McGraw-Hill Education (Singapore) Pte. Ltd. and Tsinghua University Press Limited.

版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社有限公司合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港特别行政区、澳门特别行政区和台湾)销售。

版权©2014 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社有限公司所有。

北京市版权局著作权合同登记号 图字: 01-2013-7601

本书封面贴有 McGraw-Hill Education 公司防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

UNIX/Linux 应用、编程与系统管理(第3版)/(美)达斯(Das, S.) 著; 贾洪峰, 李莉 译.

—北京: 清华大学出版社, 2014

书名原文: Your UNIX/Linux: The Ultimate Guide, Third Edition

(国外计算机科学经典教材)

ISBN 978-7-302-35584-7

I. U… II. ①达… ②贾… ③李… III. UNIX 操作系统—程序设计—教材 ②Linux 操作系统—程序设计—教材 IV. ①TP316.8

中国版本图书馆 CIP 数据核字(2014)第 038574 号



责任编辑: 王 军 于 平

装帧设计: 牛艳敏

责任校对: 邱晓玉

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185mm × 260mm 印 张: 46.25 字 数: 1126 千字

版 次: 2014 年 4 月第 1 版 印 次: 2014 年 4 月第 1 次印刷

印 数: 1~3000

定 价: 99.00 元

产品编号: 054778-01

前言

一门不能改变思维方式的语言是不值得掌握的。

—Larry Wall(Perl 的发明者), Randal Schwartz

早在开始讲授本课程之前，我就决定写一本关于 UNIX 的书。我对当时市面上的相关书籍都不满意，也几乎没有任何专家可以寻求帮助。在费尽千辛万苦学完本课程之后，我决定写本书，向大家介绍自己的经验，但当时并没有想清楚它应当面向什么样的读者对象。多年之后，在我开始讲授这门课程时却发现，学生们提出的一些问题都已经在我的书中做了解答。这时，我才知道自己做对了，这实际上是一本教科书。

UNIX 经受住了时间的考验，而且在今后一段时间内应当仍然如此。但在说起 UNIX 时，还是有很多人会说它“不够友好”、“让人挠头”。初学者还是会因为它显而易见的复杂性而备感打击。即便是经验丰富的计算机专业人士，也很难再乐于接受它的工作方式，不愿意触碰它的命令行、无数的选项和复杂的语法。今天的所有 UNIX 和 Linux 系统都提供了图形用户界面，但命令行仍然是 UNIX 系统的主干。

UNIX 令人激动之处在于它的许多功能都是隐藏的。UNIX 并没有把一切都摆在明面上；它鼓励用户去创造、去革新。对于 UNIX 爱好者来说，如何找出能够完成一项复杂任务的命令组合或脚本，都是真正的挑战。这就是 UNIX，而且它最好继续保持这种风格。如果你欣赏这样一句话——“任何疯狂背后皆有章法可循”，那你也就上道了，这本书就是为你准备的。

与众不同

我最早接触的计算机就是 UNIX 机，在 UNIX 系统中摸爬滚打很长一段时间后，我认识到，妨碍理解 UNIX 的绊脚石并不是人们认为的那些。UNIX 之所以不容易理解，并不是因为它太难了，而是因为作者们缺乏足够的想象力，没有为它换一种表现方式。例如，书中经常会给出一条命令的 20 个选项，而其中可能只有一两项是真正有用的。我在饱受挫折之后，决定自行编写一本“真正”的 UNIX 书籍——一本知道用户真正需要什么 UNIX 书籍，一本用户愿意随身携带的 UNIX 书籍。

现实示例 UNIX 的概念很简单，但也很抽象，为什么要采用某一特定方式来处理某一

特定功能，其原因往往不是那么明显。要掌握这一操作系统，就需要全面理解这些概念。我确信已经足够清晰地解释了它的关键特性，既揭示了在设计时的考虑因素，也揭示了它们与现实世界的相关性。你会发现，本书中的许多例子都源于现实生活中的情景。

用户指南与程序员指南 UNIX 用户主要有两类，一类用户使用 UNIX 的原生工具；另一类用户则使用 shell 脚本和 UNIX 系统调用库来开发工具。本书足够详细地讨论了这两方面的内容，又使篇幅保持在便于阅读、携带的范围内(在同类书籍中，可能只有本书能做到这一点)。

前 11 章为“用户”服务，对于导论性的 UNIX 课程来说，这些内容就足够了。“开发人员”是 shell 或系统程序员，他们还得知道工作机理，例如在创建或链接一个文件时，目录会受到什么样的影响。为他们着想，前面几章包括了一些用于探究关键概念的特殊加框文字。这种安排应当不会对初学者造成影响，他们可以悄悄地忽略这些部分。第 16~18 章体现了 UNIX 的精华，要求系统程序员必须阅读这些章节。

醒目的学习提示 教学提示是本书的一个突出特征。它们有各种不同的名称，例如“注意”、“警告”和“提示”。我认为 Linux 是 UNIX 家庭中的一个重要成员，所以特别采用企鹅标记来突出显示 Linux 特性。

许多作者都是在单独一章中讨论每种 shell，我不太赞同这种做法。我主要使用 Bash shell 来讨论关键概念，然后再分别讨论 C shell、Korn shell、Bourne shell 的不同之处。

大量习题与练习 本书给出了大量问题，用来测验读者掌握的知识。其中有超过 1/3 是自测问题，其答案在附录 I 中给出。这些问题都是面向初学者的，在继续学习下一章之前，应当首先出色地回答这些问题。

难度更大、范围更广的问题留在“习题”部分。其中有一部分的难度很大，可能需要花费一些时间进行求解。这些习题会巩固(往往还会丰富)你的 UNIX 知识，所以不要忽视它们。这些问题的答案在本书网站上提供：<http://www.mhhe.com/das>。

新增内容

一些评论者似乎对本书第 2 版非常满意，他们警告说，再做改进可能不会有“太大收效”。我在牢记这一点的同时，在这一版中进行了以下修改：

- vi/vim 一章得以保留，而 emacs 一章则被压缩，转为附录。为使初学者更轻松过渡到 vi，第 3 章介绍了 pico 编辑器。
- bc 计算器实用程序在这一版中又回归了，但只是作为 shell 程序员的一个高效计算工具。
- 专门为讨论 C 语言程序设计的基础知识而增加了一章。尽管有些简略，但有了这一章，就足以理解后面两章的系统程序设计内容了。

- 现在的第 15 章包括了流行的并发版本系统(CVS),所有 Linux 系统中都可以找到它。SCCS 和 RCS 在这一版中继续保有一席之地,但在下一版,至少会删除它们中的一项。
- 由于 GNU 调试程序(gdb)优于 dbx,所以这一版中已经将 gdb 收纳在内,不过后者依然得以保留。

这些变化使本书的篇幅稍有增加。尽管增加了对 Linux 和 GNU 工具的介绍,但本书的通用特性仍然保留下来;它没有专门关注 UNIX 的任何具体分发版本,而是突出介绍了 Solaris 和 Linux 中的各种变型。

组织结构

这一版在逻辑上划分为用户部分和程序设计部分。UNIX 基础知识仅限于前11章,这部分内容在讨论联网工具时达到高潮。接下来的7章涵盖了程序设计的相关资料,包括 awk、shell 程序设计、系统程序设计和 perl。

UNIX 简介 第1章通过一个简单的实践练习揭示了 UNIX 的核心概念。接下来是 UNIX 的发展简史及功能展示。这一章还向读者介绍了 kernel 和 shell,它们共同处理系统的工作负载。在为开发可移植应用程序而构建框架的过程中,一些标准机构(例如 POSIX 和 The Open Group)扮演着重要角色,学习本章之后将会了解这些机构发挥的作用。

第 2 章介绍了 UNIX 命令行的结构。它还讨论了使用 man 命令查询联机文档的技巧。我们将学习使用电子邮件程序,改变密码,并了解系统中发生了什么。我们可能会出错,而且迟早会出错,所以还要知道如何使用键盘进行纠正。

文件 文件是 UNIX 的两大支柱之一,接下来的三章将讨论文件。第 3 章讨论了系统中各种类型的文件,以及处理这些文件的命令。我们将学习如何创建目录、在目录结构中导航定位、复制和删除目录中的文件。还将学习如何使用 pico 编辑器来编辑文件。UNIX 还提供了许多压缩实用工具,可以用来节省磁盘空间。

文件拥有属性(特性),第 4 章介绍了文件的主要属性,特别是用 ls -l 命令显示的属性。要知道,文件和目录面对攻击都是大门洞开的,所以要学会如何通过管理它们的权限来提供保护。利用链接,以多个名称来访问同一文件。我们可能还会忘记文件的存放位置,所以需要熟悉 find 命令。

一个人的工作效率还取决于他对编辑器功能的熟悉程度。第 5 章介绍了 vi,它在任何工作环境下都是功能最强大的文本编辑器之一。程序员使用这一编辑器的次数可能比其他任何人都多,所以本章中的大多数例子都采用程序代码段。附录 D 中对 vi 的功能进行了总结。

shell和进程 现在需要理解一个非常重要的程序,我们总是和它打交道的,那就是 shell。第 6 章介绍了shell的解释功能,包括它的许多元字符。我们将学习使用通配符,用单个模式来匹配一组类似的文件名。使用重定向和管道来操控命令的输入和输出。shell也

是一门编程语言，所以必须要等到第 13 章才能完全理解它。

第 7 章介绍了进程，它是 UNIX 系统的另一支柱。进程与文件类似，也拥有属性。这一章将帮助读者理解如何用 *fork-exec* 机制来创建进程，学习控制进程，在前台和后台之间移动它们，还将学习如何通过发送信号来终止进程。

UNIX shell 为定制环境提供了很好的机会(第 8 章)。要理解和使用 shell 变量、创建命令别名。选择一种提供所有这些功能的合适 shell，并学习使用初始化脚本来保存对环境的修改。

过滤器 接下来的两章讨论过滤器——这是 UNIX 工具包中的一些特殊命令，负责所有文本处理任务。第 9 章介绍了一些简单的过滤器，并说明它们在相互连接时，如何最为有效。专门有一个示例小节，其中给出了现实生活中的三个应用，它们都是在管道中使用这些过滤器加以解决的。

第 10 章讨论了两种功能强大的过滤器——*grep* 和 *sed*，它们共同处理所有的模式查找、编辑和替换操作。这里将介绍正则表达式，它是一种精心设计的模式匹配机制，通常会大幅简化查找与替换任务。过滤器内容之后有一章讨论联网工具(第 11 章)，本书第一部分到此结束。

程序设计 后面的 7 章讨论了 UNIX 的大多数程序设计功能。在第 12 章中，*awk* 命令表现得像是一个过滤器和一门程序设计语言。掌握了 *awk* 及其标准程序设计构造(例如 *if*、*for* 和 *while* 构造)，应当就为 shell 程序设计、*perl* 和 C 语言程序设计打好了基础。

最后，我们会将所有命令和管道放到 shell 脚本中。利用第 13 章讨论的 shell 程序设计功能，开发交互式与非交互式脚本。学习设计一种脚本，在以不同名字调用这种脚本时，其行为会发生相应变化。对 shell 程序员来说，本章给出的三个示例脚本是必读内容。

下面 4 章与 C 程序设计有着直接或间接的关系。第 14 章介绍了 C 语言程序设计的入门知识，这是这一版新增的唯一一章。接下来是程序开发工具(第 15 章)。利用 *make* 实用工具和一种功能强大的调试程序(*gdb*)来管理和调试程序。另外，还要学习使用 *SCCS*、*RCS* 和 *CVS* 来维护一个程序的多个版本。

本书有两章介绍了系统调用在 C 语言程序设计环境中的应用，第 16 章是其中的第一章，这一章讨论了与文件和 I/O 相关的系统调用。编写一些程序，执行面向目录的功能，例如列出文件等，还要学习如何提取和处理存储在 *inode* 中的文件属性。

第 17 章讨论了与进程有关的系统调用。学习使用系统调用中的 *fork* 和 *exec* 系列来创建进程。一旦理解了内核如何在内存中为已打开文件维护元数据，就能实现重定向和管道，以及如何在程序中处理信号。

我们在第 18 章遇到了 *perl*，它是 UNIX 世界中功能最强大的过滤器和脚本设计语言。大多数 UNIX 概念都内嵌在 *perl* 的设计中，许多 UNIX 用户离开它就什么也做不了，原因就在于此。尽管我们无法在单独一章中充分欣赏 *perl*，但第 18 章作为一个开端，应当是很有用的。

最后，每个用户都必须了解与系统管理有关的例程任务，第 19 章讨论了这一领域的基

本问题。理解系统提供的重要安全功能。熟悉与系统启动与关闭相关的活动，以及如何加载文件系统，并检查一致性。还要学习如何进行基本的备份操作。

致谢

本书的每一版都使我有机会与新的团队进行合作。本项目的主要推动者是 **Raghu Srinivasan**(全球出版商)，同他和 **Melissa Leick**(项目经理)的合作非常愉快。他们在 McGraw-Hill 指导团队的方式给我留下了深刻印象。我还必须感谢 **Brenda Rolwes**(设计协调者)和 **Curt Reynolds**(他负责营销安排)。还要特别感谢 **Melinda Bilecki**(策划编辑)，她一如既往地机敏。她是唯一一位从本书构思阶段就一直参与的人，这种连续性使我受益良多。

还必须感谢审稿者，他们对于本版形式、内容的调整发挥了重要作用：

Ivan Bajic，圣地亚哥州立大学

Bruce Elenbogen，密西根大学迪尔本分校

Ziad Kobti，加拿大温莎大学

Galina Piatnitskaia，密苏里大学圣路易斯分校

Paul Piwowarski，肯塔基大学

Darren Provine，罗文大学

Quazi M. Rahman，加拿大西安大略大学

我要向 **Deepti Narwat** 表示最诚挚的谢意，他是负责 **Cenveo Publisher Services** 的项目经理，本书的整个生产流程都由他负责。还有其他许多人不能一一提及，但他们都为本书做出了同样的贡献。

最后的“至理名言”

大多数例子都已经在 Solaris 和 Linux 上进行了测试，但不能保证它们在所有系统上都可以无错误运行。由于 UNIX 的分化，实际上不可能在所有 UNIX 中进行全面测试。如果有一些命令无法以本书说明的方式正常工作，请不要得出系统有 bug 的结论。不过，这些例子中仍然可能存在 bug，欢迎读者指出。

在结束前言之前，应当给出一点提醒建议。许多人都因为思维混乱、受到误导而错过了 UNIX 班车，现在深以为憾。不要让这件事再发生在你身上。一旦你决定探索 UNIX，应当学习利用已有的功能，不要再去重复实现别人已经实现过的重要功能。你很快就会发现，一个充满机会和激情的世界向你张开了怀抱。一定要充满热情和自信地探索这一领域，我与你同在。

Sumitabha Das

目 录

第 I 部分 UNIX 之用户篇

第 1 章 UNIX 简介	3
1.1 操作系统	4
1.2 UNIX 操作系统	5
1.3 了解你的计算机	5
1.4 系统管理员	7
1.5 登录和注销	7
1.5.1 登录	7
1.5.2 注销	8
1.6 实践练习	9
1.6.1 用 date 和 who 查看 系统信息	9
1.6.2 用 ps 查看进程	10
1.6.3 处理文件	11
1.6.4 处理目录	12
1.7 UNIX 的成功之路	14
1.7.1 伯克利: 第二学派	14
1.7.2 UNIX 的分化	15
1.7.3 互联网	15
1.7.4 Windows 的威胁	15
1.8 POSIX 和单一 UNIX 规范	16
1.9 Linux 和 GNU	16
1.10 UNIX 体系结构	17
1.10.1 分工: Kernel 和 shell	17
1.10.2 文件和进程	18
1.10.3 系统调用	19
1.11 UNIX 的功能	19
1.11.1 多用户系统	19
1.11.2 多任务系统	20
1.11.3 应用程序库	20

1.11.4 构建基块方法	20
1.11.5 模式匹配	21
1.11.6 程序设计实用工具	21
1.11.7 文档	21
1.12 小结	22
1.13 自测题	22
1.14 习题	23
第 2 章 熟悉 UNIX 命令	25
2.1 命令基础	26
2.1.1 PATH: 定位命令	26
2.1.2 命令的存放位置	27
2.2 命令结构	28
2.3 命令应用的灵活性	30
2.4 man: 联机帮助	31
2.4.1 导航与查找	33
2.4.2 用 man -k 和 man -f 获取更多帮助	33
2.5 man 文档	33
2.5.1 了解 man 页	34
2.5.2 用 man 来理解 man	35
2.6 echo: 显示消息	37
2.7 printf: echo 的替代命令	39
2.8 script: 记录会话	39
2.9 以 mailx 命令使用电子邮件	40
2.9.1 发送邮件	40
2.9.2 接收邮件	41
2.9.3 mailx 内部命令	42
2.10 passwd: 改变密码	43
2.11 uname: 显示计算机名和 操作系统	44

2.12	who: 了解用户	45	3.21	dos2unix、unix2dos 和 Tofrodos: 在 DOS 和 UNIX 之间转换	80
2.13	date: 显示系统日期	45	3.22	tar: 存档程序	81
2.14	stty: 在出错时	46	3.23	gzip: 压缩程序	82
2.15	X Window 系统	48	3.24	zip: 压缩和存档程序	84
2.15.1	终端模拟器	48	3.25	使用这些命令的其他方式	85
2.15.2	文件管理器	49	3.26	小结	85
2.16	小结	50	3.27	自测题	86
2.17	自测题	51	3.28	习题	86
2.18	习题	52			
第 3 章	文件系统	55	第 4 章	文件属性	89
3.1	文件	56	4.1	重温 ls(-l): 列出文件属性	89
3.1.1	普通(常规)文件	56	4.2	文件权限	91
3.1.2	目录文件	56	4.3	chmod: 改变文件权限	92
3.1.3	设备文件	57	4.3.1	相对权限	93
3.2	文件名中有什么	57	4.3.2	绝对赋予权限	94
3.3	文件系统的层次结构	58	4.3.3	递推操作(-R)	95
3.4	UNIX 文件系统	59	4.4	目录	96
3.5	随命令一起使用绝对路径名	60	4.4.1	读权限	96
3.6	HOME 变量和~: 主目录	61	4.4.2	写权限	96
3.7	pwd 和 cd: 在文件系统中 导航	61	4.4.3	执行权限	98
3.8	相对路径名(.和..)	63	4.5	umask: 默认的文件与 目录权限	98
3.9	mkdir: 创建目录	64	4.6	文件系统和 inode	99
3.10	rmdir: 删除目录	65	4.7	ln: 创建硬链接	101
3.11	ls: 列出文件	66	4.8	再论 ln: 创建符号链接	103
3.12	cp: 复制文件	69	4.9	文件所有权	105
3.13	mv: 重命名文件	70	4.9.1	chown: 改变文件所有权	106
3.14	rm: 删除文件	71	4.9.2	chgrp: 改变用户组 所有者	106
3.15	cat: 显示和串联文件	73	4.9.3	如何处理入侵者	107
3.16	more: UNIX 分页器	74	4.10	修改时间与访问时间	108
3.17	pico: 基本文本编辑	76	4.11	find: 定位文件	108
3.17.1	导航	77	4.11.1	选择准则	110
3.17.2	文本编辑	77	4.11.2	find 操作符(!, -o 和-a)	112
3.18	wc: 计算行数、字数和 字符数	77	4.11.3	操作部分的操作符	112
3.19	lp: 打印文件	78	4.12	小结	113
3.20	od: 查看不可打印字符	80	4.13	自测题	113

4.14 习题	114	5.9.3 修改文本(c)	139
第 5 章 vi/vim 编辑器	117	5.10 在文件之间复制和 移动文本	139
5.1 vi 基础	118	5.11 撤消之前的编辑 指令(u 和 U)	140
5.2 一些提示	120	5.12 查找模式(/和?)	141
5.3 输入模式——输入和 替换文本	121	5.13 重复上一条命令(.)	142
5.3.1 插入和追加文本(i 和 a)	122	5.14 替代-查找并替换(:s)	143
5.3.2 在行的两端插入和追加 文本(I 和 A)	122	5.15 set: 定制 vi	144
5.3.3 新建一行(o 和 O)	123	5.16 map: 映射键盘的按键	146
5.3.4 替换文本(r、s、R 和 S)	124	5.17 abbr: 文本输入的缩写	147
5.3.5 输入控制字符([Ctrl-v])	125	5.18 小结	147
5.4 保存文本并退出——ex 模式	126	5.19 自测题	148
5.4.1 保存工作(:w)	127	5.20 习题	149
5.4.2 保存并退出(:x 和:wq)	127	第 6 章 shell	151
5.4.3 中止编辑(:q)	127	6.1 作为命令处理器的 shell	152
5.4.4 写入选定行	128	6.2 shell 提供的功能	152
5.4.5 退出到 shell(:sh 和 [Ctrl-z])	129	6.3 模式匹配——通配符	153
5.5 从崩溃中恢复(:recover 和-r)	129	6.3.1 *和?	154
5.6 导航	130	6.3.2 字符类	155
5.6.1 4 个方向上的相对 移动(h、j、k 和 l)	131	6.3.3 匹配句点	156
5.6.2 滚动([Ctrl-f]、[Ctrl-b]、 [Ctrl-d]和[Ctrl-u])	131	6.3.4 综述	157
5.6.3 单词导航(b、e 和 w)	132	6.4 转义与引用	157
5.6.4 移到行的末端(0、 和\$)	132	6.4.1 转义	158
5.6.5 绝对移动(G)	133	6.4.2 引用	159
5.7 编辑文本，不使用操作符	133	6.4.3 在 echo 中进行转义	159
5.7.1 删除文本(x、X 和 dd)	133	6.5 重定向	160
5.7.2 移动文本(p)	134	6.5.1 标准输入	160
5.7.3 合并行(J)	134	6.5.2 标准输出	162
5.7.4 改变大小写(~)	135	6.5.3 文件描述符	163
5.8 纠正 C 程序	135	6.5.4 标准错误	164
5.9 用操作符编辑文本	136	6.5.5 过滤器——使用标准 输入与标准输出	165
5.9.1 删除和移动文本 (d、p 和 P)	137	6.6 集体处理	165
5.9.2 复制文本(y、p 和 P)	138	6.6.1 复制描述符	166
		6.6.2 命令分组	166
		6.7 /dev/null 和/dev/tty: 两个 独立的文件	167

6.8 管道 168

6.9 tee: 创建一个 T 型三通 170

6.10 命令替换 171

6.11 shell 变量 172

 6.11.1 引用和转义的影响 173

 6.11.2 使用 shell 变量的场合 174

6.12 shell 脚本 175

6.13 shell 对命令行的处理 175

6.14 有关通配符的更多信息 176

6.15 xargs: 构建动态命令行 177

6.16 小结 177

6.17 自测题 178

6.18 习题 179

第 7 章 进程 181

 7.1 进程基础 182

 7.2 shell 和 init 182

 7.3 ps: 显示进程属性 183

 7.4 系统进程和 init 184

 7.5 进程创建机制 187

 7.6 继承的进程属性 188

 7.7 不能使用单独进程的情景 190

 7.8 进程状态与僵尸
 进程(Zombie) 190

 7.9 信号处理 191

 7.10 在后台运行作业 193

 7.10.1 &: 不注销 194

 7.10.2 nohup: 安全注销 194

 7.11 作业控制 195

 7.12 at 和 batch: 推后执行 198

 7.12.1 at: 一次执行 198

 7.12.2 batch: 在批队列中
 执行作业 199

 7.12.3 限制 at 和 batch 的使用 199

 7.13 cron 和 crontab: 定期
 运行作业 199

 7.14 小结 202

 7.15 自测题 203

 7.16 习题 203

第 8 章 shell—定制环境 205

 8.1 shell 205

 8.2 环境变量 207

 8.3 常见环境变量 208

 8.4 别名 212

 8.5 命令历史记录 213

 8.5.1 访问之前的命令 214

 8.5.2 在先前命令中进行替换 215

 8.5.3 使用上一条命令中的
 参数(\$) 215

 8.5.4 历史记录变量 216

 8.6 行内命令编辑 217

 8.7 ~替换 218

 8.8 使用 set 选项 219

 8.9 初始化脚本 220

 8.9.1 登录脚本 221

 8.9.2 rc 文件 221

 8.10 C shell 222

 8.10.1 局部变量和环境变量 222

 8.10.2 别名 224

 8.10.3 历史记录 225

 8.10.4 其他 shell 变量 225

 8.10.5 初始化脚本 226

 8.11 目录栈操作 226

 8.12 小结 228

 8.13 自测题 229

 8.14 习题 229

第 9 章 简单过滤器 231

 9.1 pr: 对文件进行编页 232

 9.2 对比文件 233

 9.3 cmp: 逐字节对比 234

 9.4 comm: 哪些共同内容 235

 9.5 diff: 将一个文件转换为
 另一文件 236

 9.6 head: 显示一个文件的开头 236

 9.7 tail: 显示文件的末尾 237

 9.8 cut: 垂直划分文件 238

 9.9 paste: 粘贴文件 240

9.10	sort: 对文件进行排序	240	10.11	文本编辑	272
9.11	uniq: 定位重复行和 非重复行	244	10.11.1	插入和改变行 (i、a、c)	272
9.12	tr: 变换字符	245	10.11.2	删除行(d)	273
9.13	应用过滤器	247	10.12	替换(s)	274
9.13.1	列出当前目录中 5 个 最大的文件	248	10.12.1	在替换中使用正则 表达式	275
9.13.2	创建一份单词使用 情况列表	249	10.12.2	记忆模式(/)	275
9.13.3	找出两个密码文件 之间的差别	249	10.13	基本正则表达式回顾	276
9.14	小结	251	10.13.1	重复模式(&)	277
9.15	自测题	251	10.13.2	间隔正则表达式 (IRE)	277
9.16	习题	252	10.13.3	标记正则表达式 (TRE)	277
第 10 章	使用正则表达式的 过滤器——grep 和 sed	255	10.14	应用 IRE 和 TRE	278
10.1	示例数据库	256	10.14.1	处理电话簿	278
10.2	grep: 查找模式	256	10.14.2	用 POSIX 兼容的函数 替换过时的函数	279
10.2.1	grep 中的引用	257	10.14.3	变换 URL 中的 路径名	280
10.2.2	grep 失败的场合	258	10.15	小结	281
10.3	grep 选项	258	10.16	自测题	281
10.4	基本正则表达式(BRE) ——简介	260	10.17	习题	282
10.4.1	字符类	262	第 11 章	联网工具	285
10.4.2	*	262	11.1	TCP/IP 基础知识	286
10.4.3	句点	263	11.2	解析主机名和 IP 地址	287
10.4.4	指定模式位置(^和\$)	264	11.2.1	/etc/hosts: 主机文件	287
10.4.5	元字符失去其意义的 情景	265	11.2.2	域名系统(DNS)	288
10.5	扩展正则表达式(ERE)和 egrep	265	11.3	客户端-服务器: 网络 应用程序如何通信	290
10.5.1	+和?	266	11.4	ping: 检查网络	291
10.5.2	匹配多种模式(、(和))	266	11.5	telnet: 远程登录	292
10.6	sed: 流编辑器	267	11.6	ftp: 文件传输协议	293
10.7	行寻址	269	11.6.1	基本文件与目录处理	293
10.8	sed 选项	270	11.6.2	传送文件	294
10.9	上下文寻址	271	11.6.3	匿名 FTP	295
10.10	将选定行写入文件(w)	272	11.7	密码学基础	296
			11.7.1	对称密钥算法	296

11.7.2 非对称密钥算法.....	297	12.3 数字处理.....	321
11.8 SSH: 安全 shell.....	297	12.4 变量和表达式.....	322
11.8.1 主机验证.....	297	12.5 比较与逻辑运算符.....	324
11.8.2 rhosts/shosts 验证机制....	298	12.5.1 字符串和数值比较.....	324
11.8.3 采用对称算法的 用户验证.....	298	12.5.2 ~和!~: 正则表达式 运算符.....	325
11.8.4 使用 SSH 代理进行 非交互式登录.....	300	12.5.3 逻辑运算符.....	327
11.9 SSH 协议.....	301	12.6 -f 选项: 将 awk 程序 存储在文件中.....	327
11.9.1 远程登录与执行 命令(ssh 和 slogin).....	301	12.7 BEGIN 和 END 节.....	328
11.9.2 用 sftp 和 scp 进行 文件传输.....	302	12.8 位置参数.....	329
11.10 互联网邮件.....	303	12.9 数组.....	329
11.11 MIME: 处理邮件中的 二进制附件.....	305	12.9.1 关联(哈希)数组.....	330
11.12 在 TCP/IP 网络上 使用 X Window.....	306	12.9.2 ENVIRON[]; 环境 数组.....	331
11.12.1 显示器.....	306	12.10 内置变量.....	331
11.12.2 使用 DISPLAY 变量.....	307	12.11 函数.....	333
11.12.3 使用-display 选项.....	307	12.12 控制流——if 语句.....	335
11.13 HTTP 和万维网.....	307	12.13 for 循环.....	336
11.13.1 统一资源定位符 (URL).....	308	12.14 while 循环.....	338
11.13.2 HTTP: Web 的协议.....	309	12.15 结论.....	339
11.13.3 运行外部程序.....	309	12.16 小结.....	340
11.13.4 HTML: 网页的语言.....	310	12.17 自测题.....	340
11.13.5 Web 浏览器.....	311	12.18 习题.....	341
11.14 Web 上的多媒体: 重温 MIME.....	311	第 13 章 shell 程序设计.....	345
11.15 小结.....	312	13.1 shell 脚本.....	346
11.16 自测题.....	313	13.1.1 script.sh: 一个 简单脚本.....	346
11.17 习题.....	314	13.1.2 She-Bang 行.....	347
第 II 部分 UNIX 之程序员篇		13.2 read: 使脚本变为 交互式的.....	348
第 12 章 用 awk 进行过滤和 程序设计.....	317	13.3 使用命令行参数.....	348
12.1 awk 预备知识.....	318	13.4 exit 和\$?: 命令的 退出状态.....	350
12.2 使用 print 和 printf.....	320	13.5 逻辑运算符&&和 ——条件执行.....	351
		13.6 if 条件句.....	351
		13.7 使用test和[]对表达式求值.....	353

13.7.1	数值比较	353	13.18	shell 函数	379
13.7.2	[]: test 的快捷方式	354	13.19	trap: 脚本如何处理信号	380
13.7.3	字符串比较	355	13.20	eval: 两次求值	381
13.7.4	文件属性检验	355	13.21	exec 语句	382
13.7.5	使用复合条件	357	13.22	小结	382
13.8	两个重要的程序设计习惯	357	13.23	自测题	383
13.8.1	以交互式、非交互式 执行一项任务	357	13.24	习题	384
13.8.2	用不同名称调用 脚本(\$0)	359	第 14 章	perl——高级操纵器	387
13.9	case 条件句	359	14.1	perl 预备知识	388
13.10	expr、bc 和 basename: 计算与字符串处理	361	14.2	变量和常数	389
13.10.1	用 expr 进行计算	362	14.3	运算符	391
13.10.2	用 bc 进行浮点计算	362	14.3.1	比较与逻辑运算符	391
13.10.3	用 expr 处理字符串	363	14.3.2	串联运算符和重复 运算符(.和 x)	392
13.10.4	basename: 改变 文件名扩展	364	14.4	标准条件句与循环	392
13.11	示例程序 1: 自动选择 上一个 C 程序	365	14.4.1	if 条件句	393
13.12	for: 用列表进行循环	366	14.4.2	while、do-while 和 for 循环	393
13.12.1	列表的来源	366	14.5	从命令行参数读取文件	394
13.12.2	使用 for 的重要应用	367	14.5.1	用单行代码读取文件	395
13.13	while: 循环	367	14.5.2	在脚本中读取文件	395
13.13.1	在循环中处理重定向	368	14.6	当前行号(\$)和范围 运算符(..)	396
13.13.2	使用 while 等待文件	369	14.7	\$_: 默认变量	396
13.13.3	找出用户占用的空间	370	14.8	字符串处理函数	397
13.13.4	break 和 continue	371	14.9	列表和数组	399
13.14	示例脚本 2: 向备份文件 提供数字扩展名	371	14.10	foreach: 循环遍历 一个列表	401
13.15	用 set 和 shift 处理 位置参数	373	14.11	两个重要的列表函数	402
13.15.1	按名称杀死进程	374	14.11.1	split: 分割到列表或 数组中	402
13.15.2	shift: 左移位置参数	374	14.11.2	join: 联接列表	404
13.15.3	IFS 变量: set 的 默认分隔符	375	14.12	dec2bin.pl: 将十进制数 变换为二进制	404
13.16	示例脚本 3: 已下载 文件的目录	376	14.13	关联数组	405
13.17	Here 文档(<<)	378	14.14	使用正则表达式	407
			14.14.1	识别空白、数位和 单词	407

14.14.2	IRE 的 TRE 功能	408	15.11.3	隐式变换	437
14.15	用 s 和 tr 函数替换	409	15.12	type_conversion.c: 演示	
14.16	文件处理	411		类型变换	438
14.17	文件测试	412	15.13	求值顺序	440
14.18	子例程	413		15.13.1 运算符优先级	440
14.19	用 perl 进行 CGI 程序			15.13.2 运算符结合性	441
	设计——概述	414	15.14	控制流回顾	441
14.19.1	查询字符串	414		15.14.1 用 switch 进行决策	442
14.19.2	GET 和 POST:			15.14.2 用 do-while 重复	443
	请求方法	415		15.14.3 break 和 continue	443
14.20	小结	415	15.15	函数	444
14.21	自测题	416	15.16	first_func.c: 两个参数和	
14.22	习题	417		返回值	445
第 15 章	C 语言简介	421	15.17	函数实参、形参和局部	
15.1	C 语言	422		变量	446
15.2	first_prog.c: 理解我们的			15.17.1 按值传递	447
	第一个 C 程序	423		15.17.2 swap_failure.c: 形参与	
				局部变量的可见性	448
15.2.1	预处理器部分	424		15.17.3 以数组为函数实参	449
15.2.2	主体	424		15.17.4 以结构作为函数	
15.2.3	main: 独一无二的函数	425		实参	450
15.3	printf 和 scanf: 两个		15.18	指针	450
	重要函数	426	15.19	pointers.c: 指针的	
15.3.1	printf: 打印到终端	426		基本功能	452
15.3.2	scanf: 由键盘输入	427	15.20	指针属性	453
15.4	变量和常数	427		15.20.1 指针运算	453
15.5	基本数据类型	428		15.20.2 其他属性	454
	15.5.1 整型	428	15.21	指针和函数	455
	15.5.2 浮点类型	429		15.21.1 解决交换问题	455
	15.5.3 字符类型	430		15.21.2 用指针返回多个值	456
15.6	sizeof.c: 确定基本数据		15.22	指针和其他数据类型	456
	类型的大小	431		15.22.1 指针和数组	456
15.7	符号常量: 有名称的常量	432		15.22.2 指针和字符串	457
15.8	数组	432		15.22.3 字符串数组	458
15.9	arrays.c: 打印数组元素	433		15.22.4 指针和结构	458
15.10	结构	434	15.23	小结	459
15.11	运算符和表达式	435	15.24	自测题	460
	15.11.1 位运算符	436	15.25	习题	461
	15.11.2 自动变换	436			

第 16 章 程序开发工具	463	16.10.2 co 和 ci: 签出和 签入	488
16.1 处理拥有多个源文件的 C 应用程序	463	16.10.3 指定修订	489
16.1.1 包含多个源文件的 应用程序	464	16.10.4 rlog 和 rcs: 其他 RCS 功能	490
16.1.2 编译和链接应用程序	466	16.11 并发版本系统(CVS)	492
16.1.3 为什么需要.o 文件	467	16.11.1 建立存储库	493
16.2 make: 使程序保持最新	468	16.11.2 导入文件	493
16.2.1 删除冗余	470	16.11.3 签出	494
16.2.2 make 的其他功能: 清理与备份	470	16.11.4 编辑和签入	494
16.2.3 宏	471	16.11.5 合并和解决冲突	495
16.3 ar: 生成库(档案)	472	16.11.6 设置标记并导出 存储库	496
16.3.1 使用库	473	16.11.7 其他功能	496
16.3.2 用 make 维护档案	474	16.12 用 gdb 调试程序	497
16.4 静态库和共享库	474	16.12.1 启动 gdb	500
16.5 用 SCCS、RCS 和 CVS 实现版本控制	475	16.12.2 使用断点	501
16.6 SCCS 会话	476	16.12.3 单步执行程序	502
16.6.1 admin: 创建一个 SCCS 文件	477	16.12.4 再赋值	503
16.6.2 get: 签出	477	16.12.5 设定观察点	504
16.6.3 delta: 签入	478	16.12.6 gdb 的其他功能	505
16.6.4 unget: 如果改变主意	480	16.13 小结	506
16.7 活动与历史信息	481	16.14 自测题	506
16.7.1 sact: 显示活动状态	481	16.15 习题	507
16.7.2 prs: 显示 SCCS 历史	481	第 17 章 系统程序设计 I——文件	509
16.8 继续以 SCCS 编辑	482	17.1 系统调用的基础知识	510
16.8.1 get -r: 改变默认的 编号顺序	482	17.1.1 系统调用剖析	510
16.8.2 处理多个版本	483	17.1.2 系统调用与库 函数的对比	510
16.8.3 rmdel 和 comb: 删除和 合并版本	484	17.2 errno 和 perror: 处理错误	511
16.8.4 使用标识关键字	485	17.3 open: 打开和创建文件	513
16.9 控制对 SCCS 的访问权限	486	17.4 close: 关闭文件	515
16.9.1 控制用户访问	486	17.5 read: 读取文件	516
16.9.2 控制发布版本	486	17.6 write: 写文件	516
16.10 版本控制系统(RCS)	487	17.7 ccp.c: 复制文件	517
16.10.1 用 ci 创建 RCS 文件	488	17.8 lseek: 定位偏移指针	519
		17.9 truncate 和 ftruncate: 截短文件	521