

UNIX/Linux/OS X 中的 Shell 编程 (第4版)

Shell
Programming
in Unix, Linux and OS X
Fourth Edition

[美] Stephen G. Kochan Patrick Wood 著
门佳 译

UNIX/Linux/OS X 中的 Shell 编程 (第 4 版)

Shell
Programming
in Unix, Linux and OS X
Fourth Edition

[美] Stephen G. Kochan Patrick Wood 著
门佳 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

UNIX/Linux/OS X中的Shell编程 : 第4版 / (美) 斯蒂芬·G. 寇肯 (Stephen G. Kochan), (美) 帕特里克·伍德 (Patrick Wood) 著; 门佳译. — 北京: 人民邮电出版社, 2017. 12

ISBN 978-7-115-47041-6

I. ①U… II. ①斯… ②帕… ③门… III. ①Linux操作系统—程序设计 IV. ①TP316.89

中国版本图书馆CIP数据核字(2017)第258662号

版权 声 明

Stephen G. Kochan and Patrick Wood: Shell Programming in Unix, Linux and OS X (Fourth Edition)

Copyright © 2017 Pearson Education, Inc.

ISBN: 9780134496009

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior consent of Addison Wesley.

版权所有。未经出版者书面许可, 对本书任何部分不得以任何方式或任何手段复制和传播。

本书中文简体字版由人民邮电出版社经 Pearson Education, Inc. 授权出版。版权所有, 侵权必究。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签。无标签者不得销售。

-
- ◆ 著 [美] Stephen G. Kochan Patrick Wood
 - 译 门 佳
 - 责任编辑 傅道坤
 - 责任印制 焦志炜
 - ◆ 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>
 - 北京市艺辉印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 21.75
 - 字数: 456 千字 2017 年 12 月第 1 版
 - 印数: 1—2 400 册 2017 年 12 月北京第 1 次印刷

著作权合同登记号 图字: 01-2016-7592 号

定价: 69.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

内容提要

本书是经典图书 *Unix Shell Programming* 时隔 15 年之后的全新升级版本，全面讲解了如何在 POSIX 标准 Shell 环境中开发程序，以充分发挥 UNIX 和类 UNIX 操作系统的潜在功能。

本书共分为 14 章，其内容涵盖了 Linux/UNIX 的基础知识，Shell 的概念、工作原理和运行机制，编写 Shell 程序时使用的一些工具，Shell 中的脚本与变量，在 Shell 中如何解释引用，传递参数，条件语句，循环，数据的读取及打印，Shell 环境，交互式以及非标准 Shell 的特性等。本书后面的两个附录还提供了 POSIX 标准 Shell 的特性汇总信息，以及有助于进一步学习掌握 Shell 编程的资源。

本书坚持以“实例教学”为理念，旨在鼓励读者动手实践，从而彻底掌握 Shell 编程。本书实例丰富，内容易懂，特别适合有志于掌握 Shell 编程的 Linux/UNIX 初级用户阅读。

关于作者

Stephen Kochan 是多本 UNIX 和 C 语言畅销书的作者与合著者，其中包括 *Programming in C*、*Programming in Objective-C*、*Topics in C Programming* 和 *Exploring the Unix System*。他之前是 AT&T 贝尔实验室的软件顾问，负责开发和讲授 UNIX 和 C 语言编程相关的课程。

Patrick Wood 是 Electronics for Imaging 公司（坐落于新泽西）的 CTO（首席技术官）。他之前曾经是贝尔实验室的一名技术人员，并在 1985 年遇到了 Kochan 先生。随后他们俩共同创建了 Pipeline Associates, Inc. 公司，提供 UNIX 咨询服务，当时他是公司的副总裁。他们共同写作了 *Exploring the Unix System*、*Unix System Security*、*Topics in C Programming* 和 *Unix Shell Programming* 等图书。

前言

在过去几十年中所出现的 UNIX 和类 UNIX 操作系统家族已经成为如今最为流行、使用最广泛的操作系统之一，这都算不上什么秘密了。对于使用了多年 UNIX 的程序员而言，一切都顺理成章：UNIX 系统为程序开发提供了既优雅又高效的环境。这正是 Dennis Ritchie 和 Ken Thompson 在 20 世纪 60 年代晚期在贝尔实验室开发 UNIX 时的初衷。

注意

在本书中，我们使用的术语 UNIX 泛指基于 UNIX 的操作系统大家族，其中包括像 Solaris 这样真正的 UNIX 操作系统以及像 Linux 和 Mac OS X 这样的类 UNIX 操作系统。

UNIX 系统最重要的特性之一就是各式各样的程序。超过 200 个基本命令会随着标准操作系统发行，Linux 还对标准命令数量做了扩充，通常能达到 700~1000 个！这些命令（也称为工具）从统计文件行数、发送电子邮件到显示特定年份的日历，可谓无所不能。

不过 UNIX 真正的威力并非来自数量庞大的命令，而在于你可以非常轻松、优雅地将这些命令组合在一起完成非常复杂的任务。

UNIX 的标准用户界面是命令行，其实就是 Shell，它的角色是作为用户和系统最底层之间（内核）的缓冲带。Shell 就是一个程序，读入用户输入的命令，将其转换成系统更易于理解的形式。它还包括了一些核心编程构件，可以做出判断、执行循环以及为变量储值。

从 AT&T 发行版（源自 Stephen Bourne 在贝尔实验室编写的初版）开始，标准 Shell 就是同 UNIX 系统捆绑在一起的。自那时起，IEEE 根据 Bourne Shell 以及后续的一些其他 Shell 制订了标准。该标准目前的（本书写作之时）版本是 Shell and Utilities volume of IEEE Std 1003.1-2001，也称为 POSIX 标准。本书余下的内容都离不开 Shell。

书中的例子在运行着 Mac OS X 10.11 的 Mac 计算机、Ubuntu Linux 14.0 以及运行着 SunOS 5.7 旧版的 Sparcstation Ultra-30 下均通过了测试。除了第 14 章中的一些 Bash 示例，其他所有的例子都是用 Korn Shell 运行的，当然，在 Bash 下也没有问题。

因为 Shell 提供了一种解释型编程语言，所以能够快速方便地编写、修改和调试程序。因此，我们使用 Shell 作为首选编程语言，等你熟练掌握了 Shell 编程之后，相信你会做出同样的选择。

本书的组织形式

本书假设你熟悉系统和命令行相关的基础知识，也就是说，知道怎么登录，知道如何创建、编辑、删除文件，也知道如何使用目录。如果你对于 Linux 或 UNIX 已经手生

了，我们在第 1 章“基础概述”中会复习一些基础知识。除此之外，这一章中也会讲到文件名替换、I/O 重定向及管道。

在第 2 章“什么是 Shell”中，解答了 Shell 究竟是什么、它的工作原理，以及如何利用 Shell 作为与操作系统交互的主要方式。你将了解每次登录时所发生的事情、Shell 程序是如何启动的、如何解析命令行以及如何为你执行其他程序。第 2 章的关键点在于要明白 Shell 不过是另一个程序罢了，没什么特别的地方。

第 3 章“常备工具”讲解了一些有助于编写 Shell 程序的工具，其中包括 cut、paste、sed、grep、sort、tr 和 uniq。在这些工具的选择上的确比较主观，但它们为书中接下来要开发的程序打下了基础。另外，本章还详细讨论了正则表达式，在很多 UNIX 命令中都对其有所涉及，如 sed、grep 和 ed。

第 4 章～第 9 章继续为编写 Shell 程序做铺垫。你将学习到如何编写自己的命令、变量的用法、编写可接受参数的程序、条件判断、循环命令（for、while 和 until）以及使用 read 命令从终端或文件中读取数据。第 5 章“引用”专门讨论了 Shell 中最有意思（通常也会令人困惑）的话题之一：如何解释引用。

到这里，所有基本的 Shell 编程构件已经全部讲完，你已经有能力编写 Shell 程序，并解决特定的问题了。

第 10 章“环境”所讲述的主题（环境）对于真正理解 Shell 的运作方式非常重要。你会在本章中学到局部变量和导出变量、子 Shell、特殊的 Shell 变量（如 HOME、PATH 和 CDPATH）以及如何设置.profile 文件。

第 11 章“再谈参数”和第 12 章“拓展内容”讲述了一些之前没有提及的知识点，在第 13 章“再谈 rolo”中给出了名为 rolo 的电话簿程序的最终版，该程序的开发过程贯穿全书。

第 14 章“交互式与非标准 Shell 特性”讨论了多种 Shell 特性，这些特性要么并非 IEEE POSIX 标准 Shell 的正式组成部分（不过在大部分 UNIX 和 Linux Shell 中都可以使用），要么主要是以交互方式使用，而非用于程序中。

附录 A “Shell 总结”中总结出了 IEEE POSIX 标准 Shell 的各种特性。

附录 B “更多的相关信息”中列出了参考资料和资源，包括不同 Shell 的下载站点。

本书所秉持的哲学是实例教学法。我们坚信：在演示某种特性的具体用法时，恰当选择的实例所带来的效果要远胜于干巴巴的陈述。“一图胜……”的那句老话看起来也适用于编码。

我们鼓励你在自己的系统中敲入并测试每个例子，只有这样你才能掌握 Shell 编程。不要畏惧尝试。试着修改例子中的命令来观察效果，或是加入不同的选项和特性，使程序变得更加实用或强健。

目录

第 1 章 基础概述	1
1.1 基础命令	1
1.1.1 显示日期和时间: date 命令	1
1.1.2 找出已登录人员: who 命令	1
1.1.3 回显字符: echo 命令	2
1.2 使用文件	2
1.2.1 列举文件: ls 命令	3
1.2.2 显示文件内容: cat 命令	3
1.2.3 统计文件中单词数量: wc 命令	3
1.2.4 命令选项	3
1.2.5 复制文件: cp 命令	4
1.2.6 文件重命名: mv 命令	4
1.2.7 删除文件: rm 命令	5
1.3 使用目录	5
1.3.1 主目录和路径名	6
1.3.2 显示工作目录: pwd 命令	7
1.3.3 更改目录: cd 命令	7
1.3.4 ls 命令的更多用法	10
1.3.5 创建目录: mkdir 命令	12
1.3.6 在目录之间复制文件	12
1.3.7 在目录间移动文件	13
1.3.8 文件链接: ln 命令	14
1.3.9 删除目录: rmdir 命令	17
1.4 文件名替换	18
1.4.1 星号	18
1.4.2 匹配单个字符	20
1.5 文件名中不易察觉的部分	21
1.5.1 文件名中的空格	21
1.5.2 其他怪异的字符	22

1.6 标准输入/输出和 I/O 重定向	22
1.6.1 标准输入和标准输出	22
1.6.2 输出重定向	24
1.6.3 输入重定向	26
1.7 管道	27
1.8 标准错误	29
1.9 命令后话	29
1.9.1 在一行中输入多个命令	29
1.9.2 向后台发送命令	30
1.9.3 ps 命令	30
1.10 命令总结	31
第 2 章 什么是 Shell	33
2.1 内核和实用工具	33
2.2 登录 Shell	34
2.3 在 Shell 中输入命令	37
2.4 Shell 的职责	38
2.4.1 程序执行	39
2.4.2 变量及文件名替换	40
2.4.3 I/O 重定向	41
2.4.4 管道	42
2.4.5 环境控制	42
2.4.6 解释型编程语言	43
第 3 章 常备工具	44
3.1 正则表达式	44
3.1.1 匹配任意字符: 点号 (.)	44
3.1.2 匹配行首: 脱字符 (^)	46
3.1.3 匹配行尾: 美元符号 (\$)	46
3.1.4 匹配字符组: [...]	47
3.1.5 匹配零个或多个字符: 星号 (*)	50
3.1.6 匹配固定次数的子模式: \{...\}	52
3.1.7 保存已匹配的字符: \(...\)	54
3.2 cut	57
3.3 paste	61

3.3.1	-d 选项	62
3.3.2	-s 选项	62
3.4	sed	63
3.4.1	-n 选项	64
3.4.2	删除行	65
3.5	tr	66
3.5.1	-s 选项	69
3.5.2	-d 选项	69
3.6	grep	70
3.6.1	正则表达式与 grep	73
3.6.2	-v 选项	74
3.6.3	-l 选项	74
3.6.4	-n 选项	75
3.7	sort	76
3.7.1	-u 选项	76
3.7.2	-r 选项	77
3.7.3	-o 选项	77
3.7.4	-n 选项	78
3.7.5	跳过某些字段	78
3.7.6	-t 选项	79
3.7.7	其他选项	80
3.8	uniq	80
3.8.1	-d 选项	81
3.8.2	其他选项	82
第 4 章	脚本与变量	83
4.1	命令文件	83
4.2	变量	86
4.2.1	显示变量值	87
4.2.2	未定义变量的值为空	89
4.2.3	文件名替换与变量	91
4.2.4	<code>\${variable}</code> 结构	92
4.3	内建的整数算术操作	92
第 5 章	引用	95
5.1	单引号	95

5.2 双引号	98
5.3 反斜线	101
5.3.1 使用反斜线续行	102
5.3.2 双引号中的反斜线	102
5.4 命令替换	103
5.4.1 反引号	104
5.4.2 <code>\$(...)</code> 结构	104
5.5 <code>expr</code> 命令	108
第6章 传递参数	111
6.1 变量 <code>\$#</code>	112
6.2 变量 <code>\$*</code>	113
6.3 在电话簿中查找联系人	113
6.4 向电话簿中添加联系人	115
6.5 从电话簿中删除联系人	116
6.6 <code>shift</code> 命令	118
第7章 条件语句	120
7.1 退出状态	120
7.2 变量 <code>\$?</code>	121
7.3 <code>test</code> 命令	124
7.3.1 字符串操作符	124
7.3.2 <code>test</code> 的另一种格式	128
7.3.3 整数操作符	129
7.3.4 文件操作符	130
7.3.5 逻辑否定操作符 <code>!</code>	131
7.3.6 逻辑“与”操作符 <code>-a</code>	131
7.3.7 括号	132
7.3.8 逻辑“或”操作符 <code>-o</code>	132
7.4 <code>else</code>	133
7.5 <code>exit</code> 命令	135
7.6 <code>elif</code>	137
7.7 <code>case</code> 命令	141
7.7.1 特殊的模式匹配字符	142
7.7.2 调试选项 <code>-x</code>	144
7.7.3 再谈 <code>case</code>	147

7.8	空命令:	148
7.9	&&和 	149
第 8 章	循环	151
8.1	for 命令	151
8.1.1	\$@变量	154
8.1.2	不使用列表的 for 命令	155
8.2	while 命令	156
8.3	until 命令	158
8.4	再谈循环	162
8.4.1	跳出循环	162
8.4.2	跳过循环中余下的命令	163
8.4.3	在后台执行循环	164
8.4.4	循环上的 I/O 重定向	165
8.4.5	将数据导入及导出循环	166
8.4.6	单行循环	166
8.5	getopts 命令	167
第 9 章	数据的读取及打印	171
9.1	read 命令	171
9.1.1	文件复制程序	171
9.1.2	特殊的 echo 转义字符	173
9.1.3	mycp 的改进版本	174
9.1.4	mycp 的最终版本	175
9.1.5	菜单驱动的电话簿程序	179
9.1.6	变量\$\$与临时文件	183
9.1.7	read 的退出状态	184
9.2	printf 命令	187
第 10 章	环境	192
10.1	局部变量	192
10.2	导出变量	194
10.3	PS1 和 PS2	198
10.4	HOME	199
10.5	PATH	199
10.6	当前目录	206

10.7 再谈子 Shell	208
10.7.1 命令	209
10.7.2 exec 命令	212
10.7.3 (...)和{ ...; }	212
10.7.4 另一种将变量传给予 Shell 的方法	215
10.8 .profile 文件	216
10.9 TERM 变量	217
10.10 TZ 变量	218
第 11 章 再谈参数	219
11.1 参数替换	219
11.1.1 \${parameter}	219
11.1.2 \${parameter:-value}	220
11.1.3 \${parameter:=value}	220
11.1.4 \${parameter:?value}	221
11.1.5 \${parameter:+value}	222
11.1.6 模式匹配	222
11.1.7 \${#variable}	224
11.2 \$0 变量	224
11.3 set 命令	225
11.3.1 -x 选项	225
11.3.2 无参数的 set	226
11.3.3 使用 set 为位置参数重新赋值	227
11.3.4 --选项	228
11.3.5 set 的其他选项	230
11.4 IFS 变量	230
11.5 readonly 命令	233
11.6 unset 命令	234
第 12 章 拓展内容	235
12.1 eval 命令	235
12.2 wait 命令	237
12.3 \$!变量	237
12.4 trap 命令	238
12.4.1 不使用参数的 trap	239

12.4.2 忽略信号	240
12.4.3 重置信号	240
12.5 再谈 I/O	241
12.5.1 <&-与>&-	242
12.5.2 行内输入重定向	242
12.5.3 Shell 归档文件	244
12.6 函数	247
12.6.1 删除函数	250
12.6.2 return 命令	250
12.7 type 命令	250
第 13 章 再谈 rolo	251
13.1 数据格式化考量	251
13.2 rolo	252
13.3 add	254
13.4 lu	255
13.5 display	256
13.6 rem	257
13.7 change	259
13.8 listall	260
13.9 样例输出	261
第 14 章 交互式与非标准 Shell 特性	265
14.1 使用正确的 Shell	265
14.2 ENV 文件	266
14.3 命令行编辑	267
14.4 命令历史	267
14.5 vi 行编辑模式	268
14.6 emacs 行编辑模式	272
14.7 访问历史记录的其他方法	275
14.7.1 history 命令	275
14.7.2 fc 命令	276
14.7.3 r 命令	277
14.8 函数	278
14.8.1 局部变量	278

14.8.2 自动载入函数.....	278
14.9 整数算术.....	278
14.9.1 整数类型.....	279
14.9.2 不同基数的数字.....	280
14.10 alias 命令.....	282
14.11 数组.....	284
14.12 作业控制.....	289
14.13 其他特性.....	292
14.13.1 cd 命令的其他特性.....	292
14.13.2 波浪符替换.....	292
14.13.3 搜索次序.....	293
14.14 兼容性总结.....	294
附录 A Shell 总结.....	295
附录 B 更多的相关信息.....	329

第 1 章

基础概述

本章将会对 UNIX 系统进行简要讲述，其中包括文件系统、基本命令、文件名替换、I/O 重定向及管道。

1.1 基础命令

1.1.1 显示日期和时间：date 命令

date 命令可以显示出日期和时间：

```
$ date
Thu Dec  3 11:04:09 MST 2015
$
```

date 会打印出星期、月份、日期、时间（24 小时制，依据系统时区设置）及年份。在本书的所有例子中，我们使用**加粗字体**，来表示用户输入的内容，使用正常字体表示 UNIX 系统显示的内容，使用楷体表示交互过程中的注释。

按 Enter（回车）键就可以将 UNIX 命令提交给系统。Enter 键表示你已经完成了输入，剩下的事情就交给 UNIX 系统了。

1.1.2 找出已登录人员：who 命令

who 命令可以用来获取当前已登录到系统中的所有用户的信息：

```
$ who
pat      tty29    Jul 19 14:40
ruth     tty37    Jul 19 10:54
steve    tty25    Jul 19 15:52
$
```

目前，已登录的用户有 3 名：pat、ruth 和 steve。除了每个用户的 ID 之外，还列出了用户所在的 tty 编号以及用户登录的日期和时间。当用户登录系统时，UNIX 系统会为用户所在的终端或网络设备分配一个唯一的标识数字，这个数字就是 tty 编号。

也可以使用 who 命令来获取本人的信息：

```
$ who am i
pat          tty29    Jul 19 14:40
$
```

who 和 who am i 其实都是同一个命令：who。在后一种用法中，am 和 i 是 who 命令的参数（这并不是一个展示命令行参数用法的好例子；只是出于对 who 命令的好奇心而已）。

1.1.3 回显字符：echo 命令

echo 命令会在终端打印出（或者说回显）你在行中输入的所有内容（这里有一些例外情况，随后你就会知道）：

```
$ echo this is a test
this is a test
$ echo why not print out a longer line with echo?
why not print out a longer line with echo?
$ echo
显示空行
$ echo one          two    three          four    five
one two three four five
$
```

在上面的例子中，你会注意到 echo 将单词间多余的空白字符（blank）压缩了。这是因为在 UNIX 系统中，单词（word）非常重要，而空白字符就是用来分隔单词的。UNIX 系统通常会忽略多余的空白字符（下一章中会详细讲述相关内容）。

1.2 使用文件

UNIX 系统只识别 3 种基本类型的文件：普通文件、目录文件和特殊文件。普通文件就是系统中包含数据、文本、程序指令或其他内容的那些文件。目录，或者说是文件夹，会在本章后续部分讲述。最后，和名字一样，特殊文件是对 UNIX 系统有特殊意义的文件，通常和某种形式的 I/O 相关联。

文件名可以由键盘上能够直接输入的任意字符（有些字符甚至可以是无法直接输入的）组成，总数量不能超过 255 个。如果文件名中的字符多于 255 个，UNIX 系统会忽略多余的字符。