

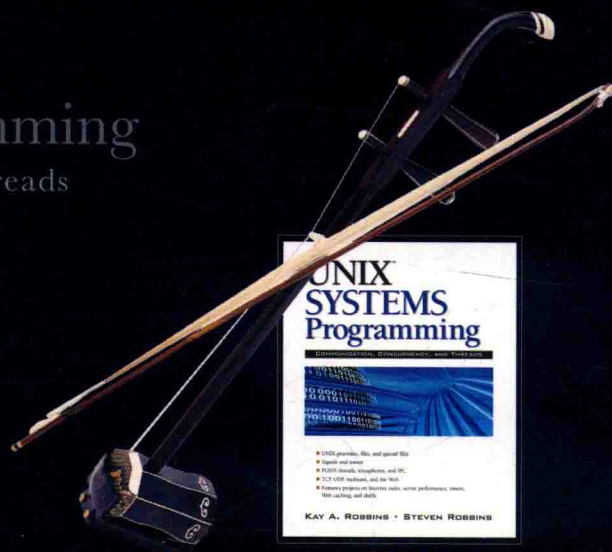
UNIX系统编程

(英文版)

通信、并发与线程

UNIX Systems Programming
Communication, Concurrency, and Threads

[美] Kay A. Robbins 著
Steven Robbins



· 原味精品书系 ·



UNIX系统编程

通信、并发与线程

(英文版)

UNIX Systems Programming
Communication, Concurrency, and Threads



【美】Kay A. Robbins Steven Robbins 著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书是一本基于最新 UNIX 标准的完备的参考书,对 UNIX 编程的要点进行了清晰易懂的介绍,从一些用于说明如何使用系统调用的短小代码段开始,逐渐过渡到能帮助读者扩展自己技能水平的实际项目中。书中对通信、并发和线程问题进行了深入探讨,对复杂的概念,例如信号和并发,进行了全面且清晰的解释。本书还覆盖了与文件、信号、信号量、POSIX 线程和客户机—服务器通信相关的内容。书中不仅提供了大量实例和练习,还专门设计了有针对性的项目并给出了参考答案。

本书适合作为高等院校计算机专业教材,也可供软件开发人员参考使用。

Original edition, entitled UNIX Systems Programming: Communication, Concurrency, and Threads 9780130424112 by Kay A. Robbins, Steven Robbins, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2003 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

China edition published by Pearson Education Asia Ltd. and Publishing House of Electronics Industry Copyright © 2017. The edition is manufactured in the People's Republic of China, and is authorized for sale and distribution only in the mainland of China exclusively (except Hong Kong SAR, Macau SAR, and Taiwan).

本书英文影印版专有出版权由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书仅限中国大陆境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

本书英文影印版贴有 Pearson Education 培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2016-2361

图书在版编目(CIP)数据

UNIX 系统编程:通信、并发与线程:英文/(美)凯罗·宾斯(Kay A. Robbins), (美)史蒂夫·罗宾斯(Steven Robbins)著. —北京:电子工业出版社,2017.4
(原味精品书系)

书名原文:UNIX Systems Programming: Communication, Concurrency, and Threads

ISBN 978-7-121-30853-6

I. ① U… II. ① 凯… ② 史… III. ① UNIX 操作系统—程序设计—英文 IV. ① TP316.81

中国版本图书馆 CIP 数据核字(2017)第 018598 号

策划编辑:刘佳禾

责任编辑:徐津平

印 刷:三河市双峰印刷装订有限公司

装 订:三河市双峰印刷装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×980 1/16 印张:57 字数:1175.2 千字

版 次:2017 年 4 月第 1 版

印 次:2017 年 4 月第 1 次印刷

定 价:198.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 51260888-819 faq@phei.com.cn。

献给 Nicole 和 Thomas。

用户注册

轻松注册成为博文视点社区用户（www.broadview.com.cn），您即可享受以下服务。

- **下载资源：**本书所提供的示例代码及资源文件均可在【下载资源】处下载。
- **提交勘误：**您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **与我们交流：**在页面下方【读者评论】处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/30853>

二维码：



关于网站

登录 <http://usp.cs.utsa.edu/usp> 可对本书资源进行免费下载。该网站内容还包括相关模拟器、测试工具、课程材料准备以及勘误列表。

前言

本书是 1995 年 Prentice Hall 出版社出版的 *Practical UNIX Programming: A Guide to Communication, Concurrency and Multithreading* 一书的第 2 版。为了更好地传达本书的内容，我们将书名修改为《UNIX 编程权威指南：通信、并发与线程（*UNIX Systems Programming: Communication, Concurrency and Threads*）》。与上一版相比，第 2 版不仅改变了书名，还对很多内容进行了修改。

互联网已经成为计算和社会领域的一个主导方面。私人信息已经联网，软件经常会受到不停的攻击。因此，编写正确的代码显得尤为重要。在这一版中，我们尝试尽量生成能够正确处理错误和特殊情况的代码。我们已经意识到，声称处理了所有错误但在给出的代码中却省略错误处理是没有效果的。不幸的是，错误处理会让代码变得更复杂，所以我们努力让代码变得更加清晰。

本书对第 1 版的另一个重要改进是采用了单一的 UNIX 规范，它被称为 POSIX。我们再也不需要决定使用哪个厂商的库函数——现在有了正式的版本。我们已经尽了最大努力来遵循这个标准。

练习和项目让本书变得与众不同。实际上，本书由美国国家科学基金会（National Science Foundation Grant）项目手册的一部分发展而来。在完成这个项目的初期开发后，我们逐渐认识到：完成这个项目所需要的资料分散在不同的地方——通常可以在提供了大量细节但几乎没有概念陈述的参考书中找到。因此，本书逐渐成为一本基于最新 UNIX 标准的完整参考书。

本书分为 4 个部分，每个部分都包含主题章节和项目章节。主题章节以循序渐进的方式涵盖了指定的内容，并以“试试看”和“看看会发生什么”的形式包含了很多示例和小练习，最后都以一个或者多个练习小节结束。本书还为进程管理、并发和通信的基本概念提供了很多编程练习。这些编程练习与传统科学课程中的实验作用相同，只有通过实践才能真正理解书中的概念。这些练习由易到难，很多练习只需要不到 100 行代码就能实现。

下表对本书的结构进行了总结——22 个章节被分为 4 个部分。其中的 15 个主题章节与 8 个项目章节相互独立。第一次通读本书时可以跳过项目章节。

部分	主题章节	项目章节
第 1 部分：基础知识	第 1 章 技术对程序的影响 第 2 章 程序 第 3 章 UNIX 中的进程 第 4 章 UNIX I/O 第 5 章 文件和目录 第 6 章 UNIX 特殊文件	第 7 章 令牌环
第 2 部分：异步事件	第 8 章 信号 第 9 章 时间和定时器	第 10 章 虚拟定时器 第 11 章 破解命令解释程序
第 3 部分：并发	第 12 章 POSIX 线程 第 13 章 进程同步 第 14 章 信号量 第 15 章 POSIX IPC	第 16 章 生产者 - 消费者 第 17 章 虚拟机
第 4 部分：通信	第 18 章 面向连接的通信 第 20 章 无连接通信	第 19 章 WWW 重定向 第 21 章 互联网广播 第 22 章 服务器性能

项目章节通过开发一个规模较大的应用程序来整合几个主题章节的资料。这些项目包含两个层面：除了说明编程思想，还引导读者理解与应用程序相关的高级主题。这些项目都是分阶段设计的，大多数完整的实现都只需要几百行代码。由于不需要编写大量代码，因此读者可以将注意力集中在概念的理解，而不是代码调试上。为了简化编程，我们提供了可用于网络通信和输出日志记录的库。对专业的程序员来说，主题章节结尾部分的练习提供了对概念的基本介绍。通常，使用本书的教师可以从中挑选几个练习和一个项目章节让学生在了一学期的课程中实现。每个项目都有多种变化，因此这些项目可以在多个学期反复使用。

读者可以用不同的方式阅读本书。第 1 部分的主题章节是阅读本书其他部分的基础。阅读完第 1 部分的主题章节后，读者可以以任何顺序阅读第 2 部分到第 4 部分的内容。但后继章节结尾部分关于交互的讨论除外（例如，线程如何与信号交互）。

本书读者应该是优秀的 C 程序员，但不一定是 UNIX C 程序员。读者应该熟悉 C 语言编程和基本的数据结构。对于刚刚接触 UNIX 的读者来说，附录 A 中给出了程序开发的必备知识。

本书包含标准函数的概要。概要右下角列出了指定函数的相关标准。

本书的内容是有限的。欢迎读者给我们提出意见和建议，读者可以给我们发送电子邮件到：authors@usp.cs.utsa.edu。虽然已经尽最大努力保证本书没有错误，但如果你是第一个向我们指出某个错误的人，我们会在本书的配套网站上对你表示诚挚的感谢。<http://usp.cs.utsa.edu/usp> 上提供了本书的相关信息，从这个 Web 站点可以下载本书中的所有代码。

致谢

非常感谢 Mike Speciner 和 Bob Lynch 通读了本书的全部手稿，并提出了很多有用的建议。我们尤其要对 Mary Lou Nohr 细心睿智的编辑工作表示感谢。还要感谢 Neal Wagner 和 Radia Perlman 给予我们鼓励和建议。

从 1988 年至今（2003 年），我们为本科生和研究生开设了操作系统课程，本书中的很多材料都曾经作为这些教学课程的一部分内容。学习这些课程的学生们经历了书稿发展的不同阶段，并对不断出现的项目进行了现场测试。他们在编程中的 bug、注释、抱怨和建议让本书变得更好，并让我们对如何将这些主题联系起来有了深刻的认识。发现了早期书稿中错误的学生有 Joseph Bell、Carlos Cardenas、Igor Grinshpan、Jason Jendrusch 和 James Manion。我们要感谢美国国家科学基金会通过 NSF-ILI 给我们拨款（USE-0950497），支持我们组建实验室，这样我们才有机会开展最初的课程，而本书正是根据这些课程编写的。NSF（DUE-975093、DUE-9752165 和 DUE-0088769）还为那些用于探讨和分析操作系统概念的工具提供了支持。

我们还要感谢 Prentice Hall 出版社的编辑 Greg Doench，感谢他在整个过程中为我们提供指导。还要感谢出版编辑 William Mara 让本书得以出版。感谢 L^AT_EX 2_ε 的制作者为我们提供了可以免费使用的排版软件。

我们还要感谢我们的家人，感谢他们给予我们无限的爱和支持，尤其要感谢我们的孩子 Nicole 和 Thomas 对这个艰巨的工程所展现出的热情和理解。

Contents

I	Fundamentals	1
1	Technology's Impact on Programs	3
1.1	Terminology of Change	4
1.2	Time and Speed	5
1.3	Multiprogramming and Time Sharing	7
1.4	Concurrency at the Applications Level	9
1.5	Security and Fault Tolerance	13
1.6	Buffer Overflows for Breaking and Entering	14
1.7	UNIX Standards	18
1.8	Additional Reading	20
2	Programs, Processes and Threads	21
2.1	How a Program Becomes a Process	22
2.2	Threads and Thread of Execution	23
2.3	Layout of a Program Image	24
2.4	Library Function Calls	26
2.5	Function Return Values and Errors	29
2.6	Argument Arrays	31
2.7	Thread-Safe Functions	38
2.8	Use of Static Variables	40
2.9	Structure of Static Objects	42
2.10	Process Environment	48
2.11	Process Termination	51
2.12	Exercise: An env Utility	54
2.13	Exercise: Message Logging	55
2.14	Additional Reading	56

3	Processes in UNIX	59
3.1	Process Identification	60
3.2	Process State	61
3.3	UNIX Process Creation and fork	64
3.4	The wait Function	71
3.5	The exec Function	78
3.6	Background Processes and Daemons	84
3.7	Critical Sections	86
3.8	Exercise: Process Chains	87
3.9	Exercise: Process Fans	88
3.10	Additional Reading	89
4	UNIX I/O	91
4.1	Device Terminology	92
4.2	Reading and Writing	92
4.3	Opening and Closing Files	102
4.4	The select Function	107
4.5	The poll Function	116
4.6	File Representation	119
4.7	Filters and Redirection	128
4.8	File Control	132
4.9	Exercise: Atomic Logging	135
4.10	Exercise: A cat Utility	141
4.11	Additional Reading	143
5	Files and Directories	145
5.1	UNIX File System Navigation	146
5.2	Directory Access	152
5.3	UNIX File System Implementation	158
5.4	Hard Links and Symbolic Links	162
5.5	Exercise: The which Command	173
5.6	Exercise: Biffing	174
5.7	Exercise: News biff	177
5.8	Exercise: Traversing Directories	179
5.9	Additional Reading	181

6	UNIX Special Files	183
6.1	Pipes	184
6.2	Pipelines	188
6.3	FIFOs	192
6.4	Pipes and the Client-Server Model	196
6.5	Terminal Control	203
6.6	Audio Device	214
6.7	Exercise: Audio	219
6.8	Exercise: Barriers	221
6.9	Exercise: The stty Command	223
6.10	Exercise: Client-Server Revisited	223
6.11	Additional Reading	223

7	Project: The Token Ring	225
7.1	Ring Topology	226
7.2	Ring Formation	227
7.3	Ring Exploration	234
7.4	Simple Communication	236
7.5	Mutual Exclusion with Tokens	237
7.6	Mutual Exclusion by Voting	238
7.7	Leader Election on an Anonymous Ring	239
7.8	Token Ring for Communication	241
7.9	Pipelined Preprocessor	243
7.10	Parallel Ring Algorithms	246
7.11	Flexible Ring	250
7.12	Additional Reading	251

II Asynchronous Events **253**

8	Signals	255
8.1	Basic Signal Concepts	256
8.2	Generating Signals	256
8.3	Manipulating Signal Masks and Signal Sets	261
8.4	Catching and Ignoring Signals— sigaction	267
8.5	Waiting for Signals— pause , sigsuspend and sigwait	273
8.6	Handling Signals: Errors and Async-signal Safety	283
8.7	Program Control with siglongjmp and sigsetjmp	286
8.8	Programming with Asynchronous I/O	288

8.9	Exercise: Dumping Statistics	299
8.10	Exercise: Spooling a Slow Device	299
8.11	Additional Reading	300
9	Times and Timers	301
9.1	POSIX Times	302
9.2	Sleep Functions	314
9.3	POSIX:XSI Interval Timers	315
9.4	Realtime Signals	320
9.5	POSIX:TMR Interval Timers	324
9.6	Timer Drift, Overruns and Absolute Time	329
9.7	Additional Reading	339
10	Project: Virtual Timers	341
10.1	Project Overview	342
10.2	Simple Timers	344
10.3	Setting One of Five Single Timers	347
10.4	Using Multiple Timers	357
10.5	A Robust Implementation of Multiple Timers	363
10.6	POSIX:TMR Timer Implementation	367
10.7	mycron , a Small Cron Facility	367
10.8	Additional Reading	368
11	Project: Cracking Shells	369
11.1	Building a Simple Shell	370
11.2	Redirection	374
11.3	Pipelines	376
11.4	Signal Handling in the Foreground	380
11.5	Process Groups, Sessions and Controlling Terminals	386
11.6	Background Processes in ush	391
11.7	Job Control	398
11.8	Job Control for ush	402
11.9	Additional Reading	405

III	Concurrency	407
12	POSIX Threads	409
12.1	A Motivating Problem: Monitoring File Descriptors	410
12.2	Use of Threads to Monitor Multiple File Descriptors	411
12.3	Thread Management	415
12.4	Thread Safety	431
12.5	User Threads versus Kernel Threads	433
12.6	Thread Attributes	436
12.7	Exercise: Parallel File Copy	443
12.8	Additional Reading	444
13	Thread Synchronization	447
13.1	POSIX Synchronization Functions	448
13.2	Mutex Locks	448
13.3	At-Most-Once and At-Least-Once-Execution	461
13.4	Condition Variables	465
13.5	Signal Handling and Threads	473
13.6	Readers and Writers	478
13.7	A strerror_r Implementation	483
13.8	Deadlocks and Other Pesky Problems	483
13.9	Exercise: Multiple Barriers	485
13.10	Additional Reading	486
14	Critical Sections and Semaphores	487
14.1	Dealing with Critical Sections	488
14.2	Semaphores	491
14.3	POSIX:SEM Unnamed Semaphores	494
14.4	POSIX:SEM Semaphore Operations	496
14.5	POSIX:SEM Named Semaphores	502
14.6	Exercise: License Manager	507
14.7	Additional Reading	509

15	POSIX IPC	511
15.1	POSIX:XSI Interprocess Communication	512
15.2	POSIX:XSI Semaphore Sets	514
15.3	POSIX:XSI Shared Memory	525
15.4	POSIX:XSI Message Queues	535
15.5	Exercise: POSIX Unnamed Semaphores	542
15.6	Exercise: POSIX Named Semaphores	543
15.7	Exercise: Implementing Pipes with Shared Memory	544
15.8	Exercise: Implementing Pipes with Message Queues	547
15.9	Additional Reading	548
16	Project: Producer Consumer Synchronization	549
16.1	The Producer-Consumer Problem	550
16.2	Bounded Buffer Protected by Mutex Locks	551
16.3	Buffer Implementation with Semaphores	555
16.4	Introduction to a Simple Producer-Consumer Problem	560
16.5	Bounded Buffer Implementation Using Condition Variables	564
16.6	Buffers with Done Conditions	565
16.7	Parallel File Copy	573
16.8	Threaded Print Server	575
16.9	Additional Reading	580
17	Project: The Not Too Parallel Virtual Machine	581
17.1	PVM History, Terminology, and Architecture	582
17.2	The Not Too Parallel Virtual Machine	584
17.3	NTPVM Project Overview	585
17.4	I/O and Testing of Dispatcher	591
17.5	Single Task with No Input	600
17.6	Sequential Tasks	601
17.7	Concurrent Tasks	604
17.8	Packet Communication, Broadcast and Barriers	605
17.9	Termination and Signals	605
17.10	Ordered Message Delivery	606
17.11	Additional Reading	606

IV	Communication	607
18	Connection-Oriented Communication	609
18.1	The Client-Server Model	610
18.2	Communication Channels	610
18.3	Connection-Oriented Server Strategies	614
18.4	Universal Internet Communication Interface (UICI)	618
18.5	UICI Implementations of Different Server Strategies	621
18.6	UICI Clients	624
18.7	Socket Implementation of UICI	629
18.8	Host Names and IP Addresses	641
18.9	Thread-Safe UICI	649
18.10	Exercise: Ping Server	652
18.11	Exercise: Transmission of Audio	653
18.12	Additional Reading	655
19	Project: WWW Redirection	657
19.1	The World Wide Web	658
19.2	Uniform Resource Locators (URLs)	658
19.3	HTTP Primer	660
19.4	Web Communication Patterns	665
19.5	Pass-through Monitoring of Single Connections	672
19.6	Tunnel Server Implementation	674
19.7	Server Driver for Testing	675
19.8	HTTP Header Parsing	676
19.9	Simple Proxy Server	679
19.10	Proxy Monitor	680
19.11	Proxy Cache	683
19.12	Gateways as Portals	684
19.13	Gateway for Load Balancing	685
19.14	Postmortem	686
19.15	Additional Reading	690