



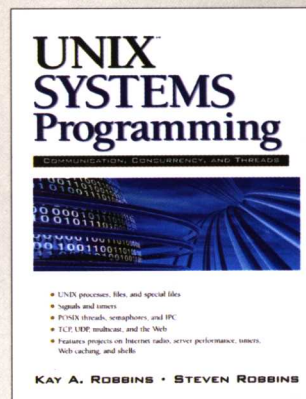
UNIX Systems Programming

Communication,
Concurrency, and Threads

[美] Kay A. Robbins Steven Robbins 著

UNIX 系统编程

(英文版)



典藏原版书苑

UNIX 系统编程

(英文版)

UNIX Systems Programming:

Communication, Concurrency, and Threads

江苏工业学院图书馆
藏书章
Steven Robbins

人民邮电出版社

图书在版编目 (CIP) 数据

UNIX 系统编程/ (美) 罗宾斯 (Robbins, K. A.), (美) 罗宾斯 (Robbins, S.) 著.

—北京: 人民邮电出版社, 2006.8

(典藏原版书苑)

ISBN 7-115-14984-4

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

中国版本图书馆 CIP 数据核字 (2006) 第 076825 号

版 权 声 明

Original edition, entitled UNIX Systems Programming: Communication, Concurrency and Threads, 0130424110 by Kay A. Robbins, Steven Robbins, published by Pearson Education, Inc, publishing as Prentice Hall PTR, Copyright © 2003 by 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 POSTS & TELECOMMUNICATIONS PRESS Copyright © 2006.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in People's Republic of China excluding Hong Kong, Macau and Taiwan.

仅限于中华人民共和国境内 (不包括中国香港、澳门特别行政区和中国台湾地区) 销售。

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。

典藏原版书苑

UNIX 系统编程 (英文版)

◆ 著 [美] Kay A. Robbins Steven Robbins

责任编辑 刘映欣

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京顺义振华印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 800×1000 1/16

印张: 57

字数: 1 226 千字

2006 年 8 月第 1 版

印数: 1—3 000 册

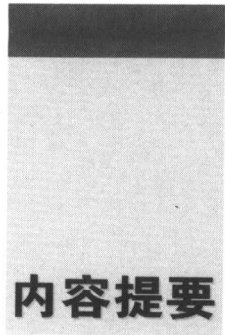
2006 年 8 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-3200 号

ISBN 7-115-14984-4/TP · 5544

定价: 88.00 元

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223



内容提要

本书是 UNIX 系统编程的经典教材，是基于最新 UNIX 标准的参考书，对 UNIX 编程本质进行了清晰透彻的介绍。本书完全覆盖文件、信号、信号量、POSIX 线程以及客户端—服务器通信等内容，对通信、并发和多线程问题进行了深入研究，并对信号和并发等复杂的概念进行了全面深入的解释。本书还包含了关于 Web、UDP 以及服务器性能等方面的更新内容，这些内容已经在实际教学中得到了广泛验证。书中还提供了大量的实例、练习、可重用的代码以及用于网络通信程序的简化库。

本书从一些代码片段开始介绍如何使用系统调用，阐述了如何设计出完善的 UNIX 系统软件，从而帮助读者提高技术水平。不管是使用 Linux、Solaris、Mac OS X 还是基于 POSIX 的系统的读者，都可以从本书中学习如何设计并实现可靠的 UNIX 软件。

To Nicole and Thomas

About the Web Site

The *<http://usp.cs.utsa.edu/usp>* web site offers additional resources for the book, including all of the programs in downloadable form. These programs are freely available with no restrictions other than acknowledgment of their source. The site also has links to simulators, testing tools, course material prepared by the authors, and an errata list.

前言

本书是 1995 年 Prentice Hall 出版的 *Practical UNIX Programming: A Guide to Communication, Concurrency and Multithreading* 的第 2 版。为了更好地体现这一版本的内容，我们对书名（英文原版）以及书中的其他许多地方做了修改。

因特网在计算机行业乃至整个社会中扮演着重要的角色。我们的个人信息一经联网，我们的软件就会经常受到攻击。因此，编写正确的代码就显得尤为重要。在这本新版中，我们尽量生成能够正确处理错误和特殊情况的代码。只是声称“处理所有的错误”却在给出的代码示例中省略错误处理是没有说服力的，但是，错误处理会使代码变得更加复杂，因此我们将尽量使代码清晰简洁。

本书对前一版本的另一个重大改进就是采用了单一 UNIX 规范，本书中称为 POSIX。这样就不用再为选用哪个厂商的库函数而发愁了——现在已有正式的版本了。在本书的编写过程中，我们尽了最大的努力来遵循这一标准。

本书的一大特点是包含练习和项目。实际上，本书就是源于一个项目手册，这个项目手册是美国国家科学基金课题（National Science Foundation Grant）的一部分。在完成这个项目的初期开发之后，我们逐渐认识到：完成这个项目所需的资料分散在很多地方，这些资料通常分散在那些提供了大量细节但很少进行概念描述的参考书中。后来，这些资料就逐步被我们总结成一本基于最新 UNIX 标准的完备的参考书了。

本书分为 4 个部分，每部分都包含一些关于主题和项目的章节。主题部分以循序渐进的方式涵盖了特定的资料，其中包括很多例子和小练习。本书还针对进程管理、并发和通信过程的基本概念提供了编程练习（即项目部分）。这些编程练习与传统学科中的实验课程有相同的作用，只有通过实践，才能真正理解书中的概念。这些练习的难度由浅入深，很多练习都只需不超过 100 行的代码即可实现。

下表中对本书的结构进行了总结。一共有 14 个主题章，另外还有 8 个独立的项目章。初次阅读本书时可以略过项目部分。

Part (部分)	Topic Chapter (主题部分)	章序号	Project Chapter (项目部分)	章序号
I Fundamentals (基础知识)	Technology's Impact (技术对程序的冲击)	1		
	Programs (程序、进程和线程)	2		
	Processes in UNIX (UNIX系统中的进程)	3		
	UNIX I/O	4		
	Files and Directories (文件和目录)	5		
	UNIX Special Files (UNIX特殊文件)	6		
			The Token Ring (令牌环)	7
II Asynchronous Events (异步事件)	Signals (信号)	8		
	Times and Timers (时间和定时器)	9		
			Virtual Timers (虚拟定时器)	10
			Cracking Shells (破解命令解释程序)	11
III Concurrency (并发)	POSIX Threads (POSIX线程)	12		
	Thread Synchronization (线程同步)	13		
	Semaphores (临界区和信号量)	14		
	POSIX IPC	15		
			Producer Consumer (生产者消费者同步)	16
			Virtual Machine (非完全并行虚拟机)	17
IV Communication (通信)	Connection-Oriented Commun (面向连接的通信)	18		
			WWW Redirection (WWW重定向)	19
	Connectionless Commun (无连接通信和多播)	20		
			Internet Radio (因特网音频广播)	21
			Server Performance (服务器性能)	22

在本书的项目部分，通过开发一个扩展的应用将几个主题中的资料整合起来。这些项目包含两层意思，除了说明编程思想之外，还引导读者深入理解与应用相关的高级话题。这些项目都是分阶段设计的，大多数完整的实现都只需几百行代码。由于不需要编写大量代码，读者可以将注意力集中在理解概念而不是调试代码上。为了简化编程，我们在网站上提供了网络通信和输出记录的库，读者可以免费下载。对专业的程序员来说，主题部分结尾处的练习提供了精简的资料。通常，使用本书的教师可以挑选几个练习和一个主要的项目在一个学期的课程中让学生们完成。每个项目都有很多变化，因此这些项目可以在多个学期中反复使用。

读者可以选择不同的方式阅读本书。Part I 的主题部分是学习本书其他内容的基础。读完 Part I 的主题部分后，读者就可以按任意顺序学习 Part II 到 Part IV 了。但后续各章结尾处有关交互（例如，线程是怎样与信号交互的）的讨论除外。

我们假定读者是一名优秀的 C 程序员，不一定是 UNIX C 的程序员。读者应该熟悉 C 语言编程和基本的数据结构。如果读者刚接触 UNIX，那么附录 A 中提供了程序开发的一些基础知识。

本书中有大量标准函数框（synopsis box），在框中的右下角列出了用于说明函数的相关标准。

本书的编写工作似乎已经结束了，但我们对知识的追求却永无止境。欢迎读者提出宝贵意见和建议。可以通过电子邮件 authors@usp.cs.utsa.edu 与我们联系。我们已经尽了最大的努力使书中的差错尽量少，但是，如果你是第一个向我们指出错误的人，我们会在本书的网站上向你表示诚挚的谢意。Web 站点 <http://usp.cs.utsa.edu/usp> 上提供了与本书相关的信息，本书包含的所有代码也都可以从该站点下载。

致谢

我们非常感谢 Mike Speciner 和 Bob Lynch，他们通读了整本书稿，并提出了很多宝贵建议。尤其要感谢 Mary Lou Nohr 细致入微的编辑工作。还要对 Neal Wagner 和 Radia Perlman 所给予的鼓励和建议表示感谢。

从 1988 年至今（2003），我们教授了大学本科生和研究生的操作系统课程，本书中很多资料都曾在这些教学课程中实践过。学习此课程的学生们都经历了书稿创作的不同阶段，并对不断出现的项目进行了实际的测试。他们提供的程序 bug、注释甚至抱怨和建议都对本书的编写工作有很大作用，并且使我们对书中所描述的这些主题之间的相互联系有了更深刻的认识。在早期书稿中发现错误的学生有 Joseph Bell、Carlos Cadenas、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 授权的 USE-0950497 对我们组建实验室提供的支持，这样我们才有机会去开展最初的课程，而这些课程正是本书的基础。NSF（DUE-975093，DUE-9752165 和 DUE-0088769）还为开发用于探索和分析操作系统概念的工具提供了支持。

我们要感谢 Prentice Hall 的编辑 Greg Doench 在本书的编写过程中提供的指导，还要感谢产品编辑 William Mara 使此书得以出版。本书使用 L^AT_EX 2_ε 排版，我们还要对使这一软件可以免费使用的生产商表达谢意。

特别要感谢我们的家人，他们给了我们无限的关爱和支持，尤其要感谢我们的孩子 Nicole 和 Thomas，他们对这项艰巨的任务给予了极大的热情和理解。

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

20 Connectionless Communication and Multicast	691
20.1 Introduction to Connectionless Communication	692
20.2 Simplified Interface for Connectionless Communication	693
20.3 Simple-Request Protocols	697
20.4 Request-Reply Protocols	702
20.5 Request-Reply with Timeouts and Retries	708
20.6 Request-Reply-Acknowledge Protocols	714
20.7 Implementation of UICI UDP	715
20.8 Comparison of UDP and TCP	724
20.9 Multicast	725
20.10 Exercise: UDP Port Server	729
20.11 Exercise: Stateless File Server	730
20.12 Additional Reading	732
21 Project: Internet Radio	733
21.1 Project Overview	734
21.2 Audio Device Simulation	735
21.3 UDP Implementation with One Program and One Receiver . .	735
21.4 UDP Implementation with Multiple Programs and Receivers .	746
21.5 UDP Implementation of Radio Broadcasts	747
21.6 Multicast Implementation of Radio Broadcasts	750
21.7 TCP Implementation Differences	750
21.8 Receiving Streaming Audio Through a Browser	755
21.9 Additional Reading	759
22 Project: Server Performance	761
22.1 Server Performance Costs	762
22.2 Server Architectures	762
22.3 Project Overview	767
22.4 Single-Client Driver	767
22.5 Multiple-Client Driver	771
22.6 Thread-per-request and Process-per-request Implementations .	774
22.7 Thread-worker-pool Strategy	774
22.8 Thread-worker Pool with Bounded Buffer	775
22.9 Process-worker Pool	775
22.10 Influence of Disk I/O	776
22.11 Performance Studies	780
22.12 Report Writing	790
22.13 Additional Reading	792