

UNIX网络编程

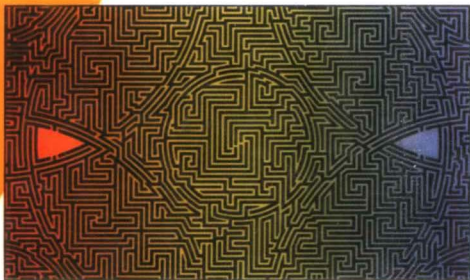
卷I 套接字联网API

(英文版·第3版)

UNIX[®] Network Programming

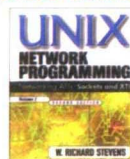
The Sockets Networking API

VOLUME 1
THIRD EDITION



W. RICHARD STEVENS
BILL FENNER
ANDREW M. RUDOFF

► Updated Classic!



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

(美) W. Richard Stevens
Bill Fenner
Andrew M. Rudoff

著

经典原版书库

UNIX网络编程

卷I 套接字联网API

(英文版·第3版)

UNIX Network Programming, Volume 1
The Sockets Networking API
(Third Edition)

江苏工业学院图书馆
藏书章

W Richard Stevens
(美) Bill Fenner 著
Andrew M. Rudoff

English reprint edition copyright © 2004 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *UNIX Network Programming, Volume 1: The Sockets Networking API, Third Edition* (ISBN: 0-13-141155-1) by W. Richard Stevens, Bill Fenner and Andrew M. Rudoff, Copyright © 2004.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版由Pearson Education Asia Ltd.授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

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)销售发行。

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。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2004-2307

图书在版编目(CIP)数据

UNIX网络编程 卷I: 套接字联网API(英文版·第3版)/(美)史蒂文斯(Stevens, W. R.)等著. -北京: 机械工业出版社, 2004. 7

(经典原版书库)

书名原文: UNIX Network Programming, Volume 1: The Sockets Networking API, Third Edition

ISBN 7-111-14685-9

I. U … II. 史 … 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(2004)第057883号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 迟振春

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2004年7月第1版第1次印刷

787mm × 1092mm 1/16 · 63.75 印张

印数: 0 001 - 4 000 册

定价: 99.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭开了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周立柱
范明
袁崇义
谢希仁

王珊
吕建
李伟琴
陆丽娜
周克定
郑国梁
高传善
裘宗燕

冯博琴
孙玉芳
李师贤
陆鑫达
周傲英
施伯乐
梅宏
戴葵

史忠植
吴世忠
李建中
陈向群
孟小峰
钟玉琢
程旭

史美林
吴时霖
杨冬青
周伯生
岳丽华
唐世渭
程时端

秘 书 组

武卫东

温莉芳

刘江

杨海玲

Foreword

When the original text of this book arrived in 1990, it was quickly recognized as the definitive reference for programmers to learn network programming techniques. Since then, the art of computer networking has changed dramatically. All it takes is a look at the return address for comments from the original text ("uunet!hsi!netbook") to make this clear. (How many readers will even recognize this as an address in the UUCP dialup network that was commonplace in the 1980s?)

Today, UUCP networks are a rarity and new technologies such as wireless networks are becoming ubiquitous! With these changes, new network protocols and programming paradigms have been developed. But, programmers have lacked a good reference from which to learn the intricacies of these new techniques.

This book fills that void. Readers who have a dog-eared copy of the original book will want a new copy for the updated programming techniques and the substantial new material describing next-generation protocols such as IPv6. Everyone will want this book because it provides a great mix of practical experience, historical perspective, and a depth of understanding that only comes from being intimately involved in the field.

I've already enjoyed and learned from reading this book, and surely you will, too.

Sam Leffler

Preface

Introduction

This book is for people who want to write programs that communicate with each other using an application program interface (API) known as sockets. Some readers may be very familiar with sockets already, as that model has become synonymous with network programming. Others may need an introduction to sockets from the ground up. The goal of this book is to offer guidance on network programming for beginners as well as professionals, for those developing new network-aware applications as well as those maintaining existing code, and for people who simply want to understand how the networking components of their system function.

All the examples in this text are actual, runnable code tested on Unix systems. However, many non-Unix systems support the sockets API and the examples are largely operating system-independent, as are the general concepts we present. Virtually every operating system (OS) provides numerous network-aware applications such as Web browsers, email clients, and file-sharing servers. We discuss the usual partitioning of these applications into *client* and *server* and write our own small examples of these many times throughout the text.

Presenting this material in a Unix-oriented fashion has the natural side effect of providing background on Unix itself, and on TCP/IP as well. Where more extensive background may be interesting, we refer the reader to other texts. Four texts are so commonly mentioned in this book that we've assigned them the following abbreviations:

- APUE: *Advanced Programming in the UNIX Environment* [Stevens 1992]
- TCPv1: *TCP/IP Illustrated, Volume 1* [Stevens 1994]
- TCPv2: *TCP/IP Illustrated, Volume 2* [Wright and Stevens 1995]
- TCPv3: *TCP/IP Illustrated, Volume 3* [Stevens 1996]

TCPv2 contains a high level of detail very closely related to the material in this book, as it describes and presents the actual 4.4BSD implementation of the network programming functions for the sockets API (`socket`, `bind`, `connect`, and so on). If one understands the implementation of a feature, the use of that feature in an application makes more sense.

Changes from the Second Edition

Sockets have been around, more or less in their current form, since the 1980s, and it is a tribute to their initial design that they have continued to be the network API of choice. Therefore, it may come as a surprise to learn that quite a bit has changed since the second edition of this book was published in 1998. The changes we've made to the text are summarized as follows:

- This new edition contains updated information on IPv6, which was only in draft form at the time of publication of the second edition and has evolved somewhat.
- The descriptions of functions and the examples have all been updated to reflect the most recent POSIX specification (POSIX 1003.1-2001), also known as the *Single Unix Specification Version 3*.
- The coverage of the X/Open Transport Interface (XTI) has been dropped. That API has fallen out of common use and even the most recent POSIX specification does not bother to cover it.
- The coverage of TCP for transactions (T/TCP) has been dropped.
- Three chapters have been added to describe a relatively new transport protocol, SCTP. This reliable, message-oriented protocol provides multiple streams between endpoints and transport-level support for multihoming. It was originally designed for transport of telephony signaling across the Internet, but provides some features that many applications could take advantage of.

- A chapter has been added on *key management sockets*, which may be used with Internet Protocol Security (IPsec) and other network security services.
- The machines used, as well as the versions of their variants of Unix, have all been updated, and the examples have been updated to reflect how these machines behave. In many cases, examples were updated because OS vendors fixed bugs or added features, but as one might expect, we've discovered the occasional new bug here and there. The machines used for testing the examples in this book were:
 - Apple Power PC running MacOS/X 10.2.6
 - HP PA-RISC running HP-UX 11i
 - IBM Power PC running AIX 5.1
 - Intel x86 running FreeBSD 4.8
 - Intel x86 running Linux 2.4.7
 - Sun SPARC running FreeBSD 5.1
 - Sun SPARC running Solaris 9

See Figure 1.16 for details on how these machines were used.

Volume 2 of this *UNIX Network Programming* series, subtitled *Interprocess Communications*, builds on the material presented here to cover message passing, synchronization, shared memory, and remote procedure calls.

Using This Book

This text can be used as either a tutorial on network programming or as a reference for experienced programmers. When used as a tutorial or for an introductory class on network programming, the emphasis should be on Part 2, "Elementary Sockets" (Chapters 3 through 11), followed by whatever additional topics are of interest. Part 2 covers the basic socket functions for both TCP and UDP, along with SCTP, I/O multiplexing, socket options, and basic name and address conversions. Chapter 1 should be read by all readers, especially Section 1.4, which describes some wrapper functions used throughout the text. Chapter 2 and perhaps Appendix A should be referred to as necessary, depending on the reader's background. Most of the chapters in Part 3, "Advanced Sockets," can be read independently of the others in that part of the book.

To aid in the use of this book as a reference, a thorough index is provided, along with summaries on the end papers of where to find detailed descriptions of all the functions and structures. To help those reading topics in a random order, numerous references to related topics are provided throughout the text.

Source Code and Errata Availability

The source code for all the examples that appear in the book is available on the Web at www.unpbook.com. The best way to learn network programming is to take these programs, modify them, and enhance them. Actually writing code of this form is the *only* way to reinforce the concepts and techniques. Numerous exercises are also provided at the end of each chapter, and most answers are provided in Appendix E.

A current errata for the book is also available from the same Web site.

Acknowledgments

The first and second editions of this book were written solely by W. Richard Stevens, who passed away on September 1, 1999. His books have set a high standard and are largely regarded as concise, laboriously detailed, and extremely readable works of art. In providing this revision, the authors struggled to maintain the quality and thorough coverage of Rich's earlier editions and any shortcomings in this area are entirely the fault of the new authors.

The work of an author is only as good as the support from family members and friends. Bill Fenner would like to thank his dear wife, Peggy (beach ¼ mile champion), and their housemate, Christopher Boyd for letting him off all his household chores while working in the treehouse on this project. Thanks are also due to his friend, Jerry Winner, whose prodding and encouragement were invaluable. Likewise, Andy Rudoff wants to specifically thank his wife, Ellen, and girls, Jo and Katie, for their understanding and encouragement throughout this project. We simply could not have done this without all of you.

Randall Stewart with Cisco Systems, Inc. provided much of the SCTP material and deserves a special acknowledgment for this much-valued contribution. The coverage of this new and interesting topic simply would not exist without Randall's work.

The feedback from our reviewers was invaluable for catching errors, pointing out areas that required more explanation, and suggesting improvements to our text and code examples. The authors would like to thank: James Carlson, Wu-Chang Feng, Rick Jones, Brian Kernighan, Sam Leffler, John McCann, Craig Metz, Ian Lance Taylor, David Schwartz, and Gary Wright.

Numerous individuals and their organizations went beyond the normal call of duty

to provide either a loaner system, software, or access to a system, all of which were used to test some of the examples in the text.

- Jessie Haug of IBM Austin provided an AIX system and compilers.
- Rick Jones and William Gilliam of Hewlett-Packard provided access to multiple systems running HP-UX.

The staff at Addison Wesley has been a true pleasure to work with: Noreen Regina, Kathleen Caren, Dan DePasquale, Anthony Gemellaro, and a very special thanks to our editor, Mary Franz.

In a trend that Rich Stevens instituted (but contrary to popular fads), we produced camera-ready copy of the book using the wonderful Groff package written by James Clark, created the illustrations using the `gpic` program (using many of Gary Wright's macros), produced the tables using the `gtbl` program, performed all the indexing, and did the final page layout. Dave Hanson's `loom` program and some scripts by Gary Wright were used to include the source code in the book. A set of `awk` scripts written by Jon Bentley and Brian Kernighan helped in producing the final index.

The authors welcome electronic mail from any readers with comments, suggestions, or bug fixes.

Bill Fenner
Woodside, California

Andrew M. Rudoff
Boulder, Colorado

October 2003
authors@unpbook.com
<http://www.unpbook.com>

Contents

Foreword		ix
Preface		x
Part 1. Introduction and TCP/IP		1
<hr/>		
Chapter 1. Introduction		3
1.1	Introduction	3
1.2	A Simple Daytime Client	6
1.3	Protocol Independence	10
1.4	Error Handling: Wrapper Functions	11
1.5	A Simple Daytime Server	13
1.6	Roadmap to Client/Server Examples in the Text	16
1.7	OSI Model	18
1.8	BSD Networking History	20
1.9	Test Networks and Hosts	22
1.10	Unix Standards	25
1.11	64-Bit Architectures	28
1.12	Summary	29
Chapter 2. The Transport Layer: TCP, UDP, and SCTP		31
2.1	Introduction	31
2.2	The Big Picture	32
2.3	User Datagram Protocol (UDP)	34

2.4	Transmission Control Protocol (TCP)	35
2.5	Stream Control Transmission Protocol (SCTP)	36
2.6	TCP Connection Establishment and Termination	37
2.7	TIME_WAIT State	43
2.8	SCTP Association Establishment and Termination	44
2.9	Port Numbers	50
2.10	TCP Port Numbers and Concurrent Servers	52
2.11	Buffer Sizes and Limitations	55
2.12	Standard Internet Services	61
2.13	Protocol Usage by Common Internet Applications	62
2.14	Summary	63

Part 2. Elementary Sockets 65

Chapter 3. Sockets Introduction 67

3.1	Introduction	67
3.2	Socket Address Structures	67
3.3	Value-Result Arguments	74
3.4	Byte Ordering Functions	77
3.5	Byte Manipulation Functions	80
3.6	inet_aton, inet_addr, and inet_ntoa Functions	82
3.7	inet_pton and inet_ntop Functions	83
3.8	sock_ntop and Related Functions	86
3.9	readn, writen, and readline Functions	88
3.10	Summary	92

Chapter 4. Elementary TCP Sockets 95

4.1	Introduction	95
4.2	socket Function	95
4.3	connect Function	99
4.4	bind Function	101
4.5	listen Function	104
4.6	accept Function	109
4.7	fork and exec Functions	111
4.8	Concurrent Servers	114
4.9	close Function	117
4.10	getsockname and getpeername Functions	117
4.11	Summary	120

Chapter 5. TCP Client/Server Example 121

5.1	Introduction	121
5.2	TCP Echo Server: main Function	122
5.3	TCP Echo Server: str_echo Function	123
5.4	TCP Echo Client: main Function	124
5.5	TCP Echo Client: str_cli Function	125
5.6	Normal Startup	126
5.7	Normal Termination	128

5.8	POSIX Signal Handling	129	
5.9	Handling SIGCHLD Signals	132	
5.10	wait and waitpid Functions	135	
5.11	Connection Abort before accept Returns	139	
5.12	Termination of Server Process	141	
5.13	SIGPIPE Signal	142	
5.14	Crashing of Server Host	144	
5.15	Crashing and Rebooting of Server Host	144	
5.16	Shutdown of Server Host	145	
5.17	Summary of TCP Example	146	
5.18	Data Format	147	
5.19	Summary	151	
Chapter 6.	I/O Multiplexing: The select and poll Functions		153
6.1	Introduction	153	
6.2	I/O Models	154	
6.3	select Function	160	
6.4	str_cli Function (Revisited)	167	
6.5	Batch Input and Buffering	169	
6.6	shutdown Function	172	
6.7	str_cli Function (Revisited Again)	173	
6.8	TCP Echo Server (Revisited)	175	
6.9	pselect Function	181	
6.10	poll Function	182	
6.11	TCP Echo Server (Revisited Again)	185	
6.12	Summary	188	
Chapter 7.	Socket Options		191
7.1	Introduction	191	
7.2	getsockopt and setsockopt Functions	192	
7.3	Checking if an Option Is Supported and Obtaining the Default	194	
7.4	Socket States	198	
7.5	Generic Socket Options	198	
7.6	IPv4 Socket Options	214	
7.7	ICMPv6 Socket Option	216	
7.8	IPv6 Socket Options	216	
7.9	TCP Socket Options	219	
7.10	SCTP Socket Options	222	
7.11	fcntl Function	233	
7.12	Summary	236	
Chapter 8.	Elementary UDP Sockets		239
8.1	Introduction	239	
8.2	recvfrom and sendto Functions	240	
8.3	UDP Echo Server: main Function	241	
8.4	UDP Echo Server: dg_echo Function	242	
8.5	UDP Echo Client: main Function	244	
8.6	UDP Echo Client: dg_cli Function	245	

8.7	Lost Datagrams	245	
8.8	Verifying Received Response	246	
8.9	Server Not Running	248	
8.10	Summary of UDP Example	250	
8.11	connect Function with UDP	252	
8.12	dg_cli Function (Revisited)	256	
8.13	Lack of Flow Control with UDP	257	
8.14	Determining Outgoing Interface with UDP	261	
8.15	TCP and UDP Echo Server Using select	262	
8.16	Summary	264	
Chapter 9. Elementary SCTP Sockets			267
9.1	Introduction	267	
9.2	Interface Models	268	
9.3	sctp_bindx Function	272	
9.4	sctp_connectx Function	274	
9.5	sctp_getpaddrx Function	275	
9.6	sctp_freepaddrx Function	275	
9.7	sctp_getladdrx Function	275	
9.8	sctp_freeladdrx Function	276	
9.9	sctp_sendmsg Function	276	
9.10	sctp_rcvmsg Function	277	
9.11	sctp_opt_info Function	278	
9.12	sctp_peeloff Function	278	
9.13	shutdown Function	278	
9.14	Notifications	280	
9.15	Summary	286	
Chapter 10. SCTP Client/Server Example			287
10.1	Introduction	287	
10.2	SCTP One-to-Many-Style Streaming Echo Server: main Function	288	
10.3	SCTP One-to-Many-Style Streaming Echo Client: main Function	290	
10.4	SCTP Streaming Echo Client: str_cli Function	292	
10.5	Exploring Head-of-Line Blocking	293	
10.6	Controlling the Number of Streams	299	
10.7	Controlling Termination	300	
10.8	Summary	301	
Chapter 11. Name and Address Conversions			303
11.1	Introduction	303	
11.2	Domain Name System (DNS)	303	
11.3	gethostbyname Function	307	
11.4	gethostbyaddr Function	310	
11.5	getservbyname and getservbyport Functions	311	
11.6	getaddrinfo Function	315	
11.7	gai_strerror Function	320	
11.8	freeaddrinfo Function	321	
11.9	getaddrinfo Function: IPv6	322	

11.10	getaddrinfo Function: Examples	324
11.11	host_serv Function	325
11.12	tcp_connect Function	326
11.13	tcp_listen Function	330
11.14	udp_client Function	334
11.15	udp_connect Function	337
11.16	udp_server Function	338
11.17	getnameinfo Function	340
11.18	Re-entrant Functions	341
11.19	gethostbyname_r and gethostbyaddr_r Functions	344
11.20	Obsolete IPv6 Address Lookup Functions	346
11.21	Other Networking Information	348
11.22	Summary	349

Part 3. Advanced Sockets **351**

Chapter 12. IPv4 and IPv6 Interoperability **353**

12.1	Introduction	353
12.2	IPv4 Client, IPv6 Server	354
12.3	IPv6 Client, IPv4 Server	357
12.4	IPv6 Address-Testing Macros	360
12.5	Source Code Portability	361
12.6	Summary	362

Chapter 13. Daemon Processes and the inetd Superserver **363**

13.1	Introduction	363
13.2	syslogd Daemon	364
13.3	syslog Function	365
13.4	daemon_init Function	367
13.5	inetd Daemon	371
13.6	daemon_inetd Function	377
13.7	Summary	379

Chapter 14. Advanced I/O Functions **381**

14.1	Introduction	381
14.2	Socket Timeouts	381
14.3	recv and send Functions	387
14.4	readv and writev Functions	389
14.5	recvmsg and sendmsg Functions	390
14.6	Ancillary Data	395
14.7	How Much Data Is Queued?	398
14.8	Sockets and Standard I/O	399
14.9	Advanced Polling	402
14.10	Summary	408