

PEARSON

UNIX

网络编程

卷 1：套接字联网 API

(第 3 版)

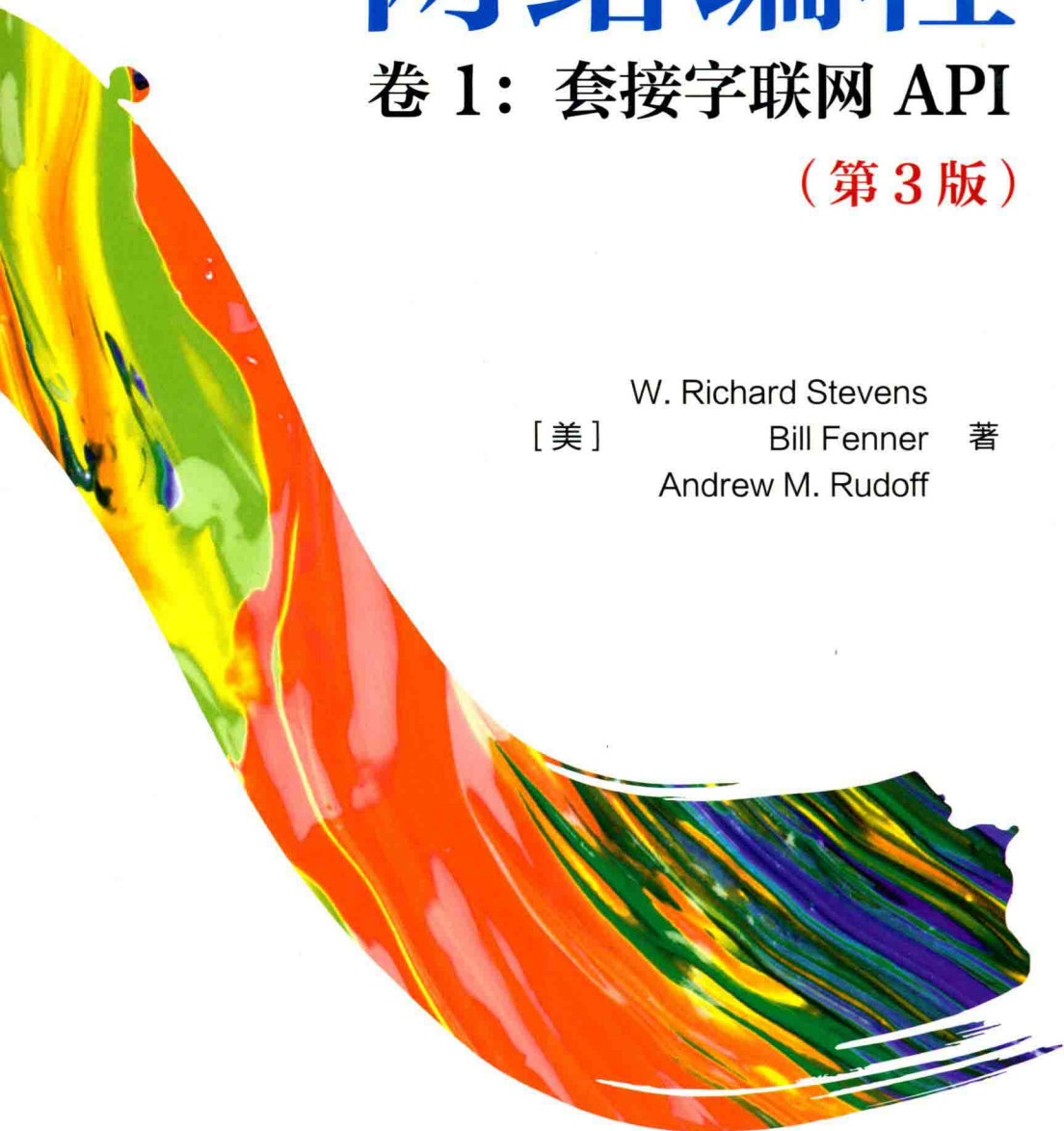
W. Richard Stevens

[美]

Bill Fenner

著

Andrew M. Rudoff



 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

Unix Network Programming, Volume 1: The Sockets Networking API Third Edition

PEARSON

UNIX

网络编程

卷 1: 套接字联网 API

(第 3 版)



[美] W. Richard Stevens
Bill Fenner 著
Andrew M. Rudoff

人民邮电出版社

北京

图书在版编目 (CIP) 数据

UNIX网络编程：第3版. 第1卷, 套接字联网API /
(美) 史蒂文斯 (Stevens, W. R.), (美) 芬纳
(Fenner, B.), (美) 鲁道夫 (Rudoff, A. M.) 著; 匿名
译. — 2版. — 北京: 人民邮电出版社, 2015. 8

书名原文: Unix Network Programming, Volume 1:
The Sockets Networking API, Third Edition
ISBN 978-7-115-36719-8

I. ①U… II. ①史… ②芬… ③鲁… ④匿… III. ①
UNIX操作系统—程序设计 IV. ①TP316.81

中国版本图书馆CIP数据核字(2015)第143512号

内 容 提 要

本书是一部 UNIX 网络编程的经典之作! 书中全面深入地介绍了如何使用套接字 API 进行网络编程。全书不但介绍了基本编程内容, 还涵盖了与套接字编程相关的高级主题, 对于客户/服务器程序的各种设计方法也作了完整的探讨, 最后还深入分析了流这种设备驱动机制。

本书内容详尽且具权威性, 几乎每章都提供精选的习题, 并提供了部分习题的答案, 是网络研究和开发人员理想的参考书。

◆ 著 [美] W. Richard Stevens Bill Fenner Andrew M. Rudoff

责任编辑 杨海玲

责任印制 张佳莹 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>

北京艺辉印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 51.5

字数: 1 363 千字

2015 年 8 月第 2 版

印数: 1-4 500 册

2015 年 8 月北京第 1 次印刷

著作权合同登记号 图字: 01-2009-5715 号

定价: 129.00 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: 京崇工商广字第 0021 号

版 权 声 明

Authorized translation from the English language edition, entitled *UNIX Network Programming, Volume 1: The Sockets Networking API, Third Edition*, 9780131411555 by W. Richard Stevens, Bill Fenner, and Andrew M. Rudoff, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2004 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.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2015.

本书中文简体字版由 Pearson Education Asia Ltd. 授权人民邮电出版社独家出版。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。

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。

序

本书的第1版于1990年问世，并迅速成为程序员学习网络编程的权威参考书。时至今日，计算机网络技术已发生了翻天覆地的变化。只要看看第1版给出的用于征集反馈意见的地址（“uunet!hsi!netbook”）就一目了然了。（有多少读者能看出这是20世纪80年代很流行的UUCP拨号网络的地址？）

现在UUCP网络已经很罕见了，而无线网络等新技术则变得无处不在！在这种背景下，新的网络协议和编程范型业已开发出来，但程序员却苦于找不到一本好的参考书来学习这些复杂的新技术。

这本书填补了这一空白。拥有本书旧版的读者一定想要一个新的版本来学习新的编程方法，了解IPv6等下一代协议方面的新内容。所有人都非常期待本书，因为它完美地结合了实践经验、历史视角以及在本领域浸淫多年才能获得的透彻理解。

阅读本书是一种享受，我收获颇丰。相信大家定会有同感。

Sam Leffler

前 言

概述

本书面向的读者是那些希望自己编写的程序能使用称为套接字 (socket) 的API进行彼此通信的人。有些读者可能已经非常熟悉套接字了, 因为这个模型几乎已经成了网络编程的同义词, 但有些读者可能仍需要从头开始学习。本书想达到的目标是向大家提供网络编程指导。这些内容不仅适用于专业人士, 也适用于初学者; 不仅适用于维护已有代码, 也适用于开发新的网络应用程序; 此外, 还适用于那些只是想了解一下自己系统中网络组件的工作原理的人。

书中的所有示例都是在Unix系统上测试通过的真实的、可运行的代码。但是, 考虑到许多非Unix的操作系统也支持套接字API, 因而我们选取的示例与所讲述的一般性概念, 在很大程度上是与操作系统无关的。几乎每种操作系统都提供了大量的网络应用程序, 如网页浏览器、电子邮件客户端、文件共享服务器等。我们按常规的划分方法把这些应用程序分为客户程序和服务器程序, 并在书中多次编写了相应的小型示例。

面向Unix介绍网络编程自然免不了要介绍Unix本身和TCP/IP的相关背景知识。需要更详尽的背景知识时, 我们会指引读者查阅其他书籍。本书中经常提到以下4本书, 我们将其简记如下。

- 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]。

其中TCPv2包含了与本书内容密切相关的细节, 它描述并给出了套接字API中网络编程函数 (socket、bind、connect等) 的真实4.4BSD实现。如果已经理解某个特性的实现, 那么在应用程序中使用该特性就更有意义了。

与第2版的区别

从20世纪80年代开始, 套接字就差不多是现在这个样子了。时至今日, 套接字仍然是网络API的首选, 其最初的设计的确值得称道。因此, 当读者发现我们对出版于1998年的第2版又做了不少改动时, 可能会觉得惊讶。本书中所做的改动归纳如下。

- 新版本包含了IPv6的最新信息。在第2版出版时, IPv6尚处于草案阶段, 这些年来已经有所发展。
- 更新了全部函数和示例的描述, 以反映最新的POSIX规范 (POSIX 1003.1-2001), 即 *Single Unix Specification Version 3*。
- 删去了X/Open传输接口 (XTI) 的内容。这个API已经不常用了, 连最新的POSIX 规范也不再提到。

- 删去了事务TCP协议（T/TCP）的内容。
- 新增了三章用于描述一种相对较新的传输协议——SCTP。这个可靠的面向消息的协议能够在两个端点之间提供多个流，并为多归属技术提供传输层支持。该协议最初是为了在因特网上传输电话信号而设计的，但它的一些特性可以用于许多应用。
- 新增一章描述密钥管理套接字，该套接字可用于网际协议安全（IPsec）和其他网络安全服务。
- 第2版中使用的机器及Unix变体都按最新版本更新，示例也根据机器的特性做了修改。许多情况下，修改示例是因为操作系统厂商修正了程序缺陷或者新增了特性。但读者可以想见，新的缺陷总能不时地被发现。本书中用于测试示例的机器如下：
 - 运行MacOS/X 10.2.6的Apple Power PC；
 - 运行HP-UX 11i 的HP PA-RISC；
 - 运行AIX 5.1的IBM Power PC；
 - 运行FreeBSD 4.8的Intel x86；
 - 运行Linux 2.4.7的Intel x86；
 - 运行FreeBSD 5.1的Sun SPARC；
 - 运行Solaris 9的Sun SPARC。

这些机器的具体用法见图1-16。

本系列的第2卷（《UNIX网络编程 卷2：进程间通信》）基于本卷的内容进一步讨论了消息传递、同步、共享内存及远程过程调用。

如何使用本书

本书既可以作为网络编程的教程，也可以作为有经验的程序员的参考书。用作网络编程的教程或入门级教材时，重点应放在第二部分（第3章至第11章），然后可以看看其他感兴趣的课题。第二部分包含了TCP和UDP的基本套接字函数，以及SCTP、I/O多路复用、套接字选项和基本名字与地址的转换。所有读者都应该阅读第1章，尤其是1.4节，介绍了一些贯穿全书的包裹函数。读者可以根据自身的知识背景，选读第2章，或许还有附录A。第三部分的大多数章节可以彼此独立地进行阅读。

为了方便读者把本书作为参考书，本书提供了完整的全文索引，并在最后几页总结了每个函数和结构的详细描述在正文中的哪里可以找到。为了给不按顺序阅读本书的读者提供方便，我们在全书中为相关主题提供了大量的交叉引用。

源代码与勘误

书中所有示例的源代码可以从www.unpbook.com获得^①。学习网络编程的最好方法就是下载这些程序，对其进行修改和改进。只有这样实际编写代码才能深入理解有关概念和方法。每章末尾提供了大量的习题，大部分在附录E中给出答案。

本书的最新勘误表也可以在上述网站获取。

^① 书中所有示例的源代码也可以从图灵网站（www.turingbook.com）本书网页免费注册下载。——编者注

致谢

本书第1版和第2版由W. Richard Stevens独立撰写，他不幸于1999年9月1日去世。Richard的著作体现了非常高的水准，被公认为是精练、翔实且极具可读性的艺术作品。在撰写这一修订版的过程中，我们力图保持Richard之前版本的高质量 and 全面性，这方面的任何不足都完全是新作者的过错。

任何作者的著作离不开家人与朋友的支持。Bill Fenner在此感谢爱妻Peggy（沙滩1/4英里赛冠军）与好友Christopher Boyd在本书撰写过程中承担了全部的家务，还要感谢朋友Jerry Winner，他的激励是无价的。同样地，Andy Rudoff要特别感谢他的妻子Ellen和两个女儿Jo、Katie自始至终的理解与鼓励。没有你们的支持，我们不可能完成本书。

思科公司的Randall Stewart提供了许多SCTP的材料，非常感谢他的巨大贡献。如果缺少了他的工作，本书就不能涵盖这一新颖而有趣的主题。

本书的审稿人给出了宝贵的反馈意见。他们发现了一些错误，指出了一些需要更多解释的地方，并对文字和代码示例提出了一些改进建议。作者在这里对如下审稿人表示感谢：James Carlson、Wu-Chang Feng、Rick Jones、Brian Kernighan、Sam Leffler、John McCann、Craig Metz、Ian Lance Taylor、David Schwartz和Gary Wright。

许多个人及其单位为本书中一些示例的测试提供了帮助，他们义务向我们出借系统、软件或为我们提供系统访问权限。

- IBM奥斯汀实验室的Jessie Haug提供了AIX系统和编译器。
- 惠普公司的Rick Jones和William Gilliam为我们提供了运行HP-UX的多个系统的访问权限。

与Addison Wesley出版社的员工合作非常愉快，他们是Noreen Regina、Kathleen Caren、Dan DePasquale和Anthony Gemellaro。要特别感谢本书的编辑Mary Franz。

为了延续Rich Stevens的风格（不过该风格与流行的风格相反），我们用James Clark编写的优秀的Groff包为本书排版，用gpic程序绘制插图（其中用到了许多由Gary Wright编写的宏），用gtbl程序生成了表格，我们为全书添加了索引，并设计了最终的版式。录入源代码时用到了Dave Hanson的lcom程序和Gary Wright写的一些脚本。在生成最终索引的过程中，还用到了Jon Bentley与Brian Kernighan编写的一组awk脚本。

欢迎读者以电子邮件的方式反馈意见、提出建议或订正错误。

Bill Fenner
加利福尼亚州伍德赛德市
Andrew M. Rudoff
科罗拉多州博尔德市
2003年10月
authors@unpbook.com
<http://www.unpbook.com>

目 录

第一部分 简介和TCP/IP

第1章 简介	2
1.1 概述	2
1.2 一个简单的时间获取客户程序	5
1.3 协议无关性	9
1.4 错误处理：包裹函数	10
1.5 一个简单的时间获取服务器程序	12
1.6 本书中客户/服务器程序示例索引表	14
1.7 OSI模型	16
1.8 BSD网络支持历史	17
1.9 测试用网络及主机	19
1.10 Unix标准	22
1.11 64位体系结构	24
1.12 小结	25
习题	25
第2章 传输层：TCP、UDP和SCTP	27
2.1 概述	27
2.2 总图	27
2.3 用户数据报协议（UDP）	29
2.4 传输控制协议（TCP）	30
2.5 流控制传输协议（SCTP）	31
2.6 TCP连接的建立和终止	31
2.7 TIME_WAIT状态	37
2.8 SCTP关联的建立和终止	38
2.9 端口号	42
2.10 TCP端口号与并发服务器	43
2.11 缓冲区大小及限制	45
2.12 标准因特网服务	50
2.13 常见因特网应用的协议使用	51
2.14 小结	52
习题	53

第二部分 基本套接字编程

第3章 套接字编程简介	56
3.1 概述	56
3.2 套接字地址结构	56
3.3 值-结果参数	61
3.4 字节排序函数	63
3.5 字节操纵函数	66
3.6 inet_aton、inet_addr和inet_ntoa函数	67
3.7 inet_pton和inet_ntop函数	68
3.8 sock_ntop和相关函数	70
3.9 readn、writen和readline函数	72
3.10 小结	76
习题	76
第4章 基本TCP套接字编程	77
4.1 概述	77
4.2 socket函数	77
4.3 connect函数	80
4.4 bind函数	81
4.5 listen函数	84
4.6 accept函数	88
4.7 fork和exec函数	90
4.8 并发服务器	91
4.9 close函数	93
4.10 getsockname和getpeername函数	94
4.11 小结	96
习题	96
第5章 TCP客户/服务器程序示例	97
5.1 概述	97
5.2 TCP回射服务器程序：main函数	97
5.3 TCP回射服务器程序：str_echo函数	98
5.4 TCP回射客户程序：main函数	99

5.5	TCP回射客户程序: str_cli函数	100	7.10	SCTP套接字选项	173
5.6	正常启动	101	7.11	fcntl函数	182
5.7	正常终止	102	7.12	小结	184
5.8	POSIX信号处理	103		习题	184
5.9	处理SIGCHLD信号	106	第8章 基本UDP套接字编程		186
5.10	wait和waitpid函数	108	8.1	概述	186
5.11	accept返回前连接中止	111	8.2	recvfrom和sendto函数	187
5.12	服务器进程终止	112	8.3	UDP回射服务器程序: main函数	187
5.13	SIGPIPE信号	113	8.4	UDP回射服务器程序: dg_echo函数	188
5.14	服务器主机崩溃	114	8.5	UDP回射客户程序: main函数	190
5.15	服务器主机崩溃后重启	115	8.6	UDP回射客户程序: dg_cli函数	190
5.16	服务器主机关机	116	8.7	数据报的丢失	191
5.17	TCP程序例子小结	116	8.8	验证接收到的响应	191
5.18	数据格式	117	8.9	服务器进程未运行	193
5.19	小结	120	8.10	UDP程序例子小结	194
	习题	120	8.11	UDP的connect函数	196
第6章 I/O复用: select 和poll 函数		122	8.12	dg_cli函数(修订版)	199
6.1	概述	122	8.13	UDP缺乏流量控制	200
6.2	I/O模型	122	8.14	UDP中的外出接口的确定	203
6.3	select函数	127	8.15	使用select函数的TCP和UDP回射服务器程序	204
6.4	str_cli函数(修订版)	132	8.16	小结	206
6.5	批量输入	133		习题	207
6.6	shutdown函数	136	第9章 基本SCTP套接字编程		208
6.7	str_cli函数(再修订版)	137	9.1	概述	208
6.8	TCP回射服务器程序(修订版)	138	9.2	接口模型	208
6.9	pselect函数	142	9.3	sctp_bindx函数	212
6.10	poll函数	144	9.4	sctp_connectx函数	213
6.11	TCP回射服务器程序(再修订版)	146	9.5	sctp_getpaddrs函数	213
6.12	小结	148	9.6	sctp_freepaddrs函数	213
	习题	149	9.7	sctp_getladdrs函数	214
第7章 套接字选项		150	9.8	sctp_freeladdrs函数	214
7.1	概述	150	9.9	sctp_sendmsg函数	214
7.2	getsockopt和setsockopt函数	150	9.10	sctp_rcvmsg函数	215
7.3	检查选项是否受支持并获取默认值	152	9.11	sctp_opt_info函数	215
7.4	套接字状态	156	9.12	sctp_peeloff函数	216
7.5	通用套接字选项	156	9.13	shutdown函数	216
7.6	IPv4套接字选项	168	9.14	通知	217
7.7	ICMPv6套接字选项	169	9.15	小结	221
7.8	IPv6套接字选项	169			
7.9	TCP套接字选项	171			

习题	222
第 10 章 SCTP 客户/服务器程序例子	223
10.1 概述	223
10.2 SCTP一到多式流分回射服务器程序: main函数	223
10.3 SCTP一到多式流分回射客户程序: main函数	225
10.4 SCTP流分回射客户程序: sctpstr_cli函数	226
10.5 探究头端阻塞	228
10.6 控制流的数目	233
10.7 控制终结	233
10.8 小结	234
习题	235
第 11 章 名字与地址转换	236
11.1 概述	236
11.2 域名系统	236
11.3 gethostbyname函数	239
11.4 gethostbyaddr函数	242
11.5 getservbyname和getservbyport 函数	242
11.6 getaddrinfo函数	245
11.7 gai_strerror函数	250
11.8 freeaddrinfo函数	251
11.9 getaddrinfo函数: IPv6	251
11.10 getaddrinfo函数: 例子	253
11.11 host_serv函数	254
11.12 tcp_connect函数	254
11.13 tcp_listen函数	257
11.14 udp_client函数	261
11.15 udp_connect函数	263
11.16 udp_server函数	264
11.17 getnameinfo函数	266
11.18 可重入函数	267
11.19 gethostbyname_r和 gethostbyaddr_r函数	270
11.20 作废的IPv6地址解析函数	271
11.21 其他网络相关信息	272
11.22 小结	273
习题	274

第三部分 高级套接字编程

第 12 章 IPv4 与 IPv6 的互操作性	278
12.1 概述	278
12.2 IPv4客户与IPv6服务器	278
12.3 IPv6客户与IPv4服务器	281
12.4 IPv6地址测试宏	283
12.5 源代码可移植性	284
12.6 小结	284
习题	285
第 13 章 守护进程和 inetd 超级服务器	286
13.1 概述	286
13.2 syslogd守护进程	286
13.3 syslog函数	287
13.4 daemon_init函数	289
13.5 inetd守护进程	293
13.6 daemon_inetd函数	297
13.7 小结	299
习题	299
第 14 章 高级 I/O 函数	300
14.1 概述	300
14.2 套接字超时	300
14.3 recv和send函数	305
14.4 readv和writev函数	306
14.5 recvmsg和sendmsg函数	307
14.6 辅助数据	310
14.7 排队的数量	313
14.8 套接字和标准I/O	313
14.9 高级轮询技术	316
14.10 T/TCP: 事务目的TCP	320
14.11 小结	322
习题	323
第 15 章 Unix 域协议	324
15.1 概述	324
15.2 Unix域套接字地址结构	324
15.3 socketpair函数	326
15.4 套接字函数	327
15.5 Unix域字节流客户/服务器程序	327
15.6 Unix域数据报客户/服务器程序	329

15.7	描述符传递	330	19.3	倾泻安全关联数据库	404
15.8	接收发送者的凭证	337	19.4	创建静态安全关联	407
15.9	小结	340	19.5	动态维护安全关联	412
	习题	340	19.6	小结	415
第 16 章	非阻塞式 I/O	341		习题	416
16.1	概述	341	第 20 章	广播	417
16.2	非阻塞读和写: <code>str_cli</code> 函数 (修订版)	342	20.1	概述	417
16.3	非阻塞 <code>connect</code>	351	20.2	广播地址	418
16.4	非阻塞 <code>connect</code> : 时间获取客户 程序	352	20.3	单播和广播的比较	419
16.5	非阻塞 <code>connect</code> : Web 客户程序	354	20.4	使用广播的 <code>dg_cli</code> 函数	422
16.6	非阻塞 <code>accept</code>	362	20.5	竞争状态	424
16.7	小结	363	20.6	小结	431
	习题	363		习题	432
第 17 章	<code>ioctl</code> 操作	365	第 21 章	多播	433
17.1	概述	365	21.1	概述	433
17.2	<code>ioctl</code> 函数	365	21.2	多播地址	433
17.3	套接字操作	366	21.3	局域网上多播和广播的比较	436
17.4	文件操作	367	21.4	广域网上的多播	438
17.5	接口配置	367	21.5	源特定多播	440
17.6	<code>get_ifi_info</code> 函数	369	21.6	多播套接字选项	441
17.7	接口操作	378	21.7	<code>mcast_join</code> 和相关函数	445
17.8	ARP 高速缓存操作	378	21.8	使用多播的 <code>dg_cli</code> 函数	450
17.9	路由表操作	380	21.9	接收 IP 多播基础设施会话声明	451
17.10	小结	381	21.10	发送和接收	454
	习题	381	21.11	SNTP: 简单网络时间协议	457
第 18 章	路由套接字	382	21.12	小结	461
18.1	概述	382		习题	461
18.2	数据链路套接字地址结构	382	第 22 章	高级 UDP 套接字编程	462
18.3	读和写	383	22.1	概述	462
18.4	<code>sysctl</code> 操作	390	22.2	接收标志、目的 IP 地址和接口索引	462
18.5	<code>get_ifi_info</code> 函数	394	22.3	数据报截断	467
18.6	接口名字和索引函数	397	22.4	何时用 UDP 代替 TCP	467
18.7	小结	401	22.5	给 UDP 应用增加可靠性	469
	习题	401	22.6	捆绑接口地址	478
第 19 章	密钥管理套接字	402	22.7	并发 UDP 服务器	482
19.1	概述	402	22.8	IPv6 分组信息	483
19.2	读和写	403	22.9	IPv6 路径 MTU 控制	486
			22.10	小结	487
				习题	488

第 23 章 高级 SCTP 套接字编程.....489	26.10 小结.....560
23.1 概述.....489	习题.....560
23.2 自动关闭的一到多式服务器程序.....489	第 27 章 IP 选项.....561
23.3 部分递送.....490	27.1 概述.....561
23.4 通知.....492	27.2 IPv4选项.....561
23.5 无序的数据.....495	27.3 IPv4源路径选项.....562
23.6 捆绑地址子集.....496	27.4 IPv6扩展首部.....569
23.7 确定对端和本端地址信息.....497	27.5 IPv6步跳选项和目的地选项.....569
23.8 给定IP地址找出关联ID.....500	27.6 IPv6路由首部.....573
23.9 心搏和地址不可达.....501	27.7 IPv6粘附选项.....577
23.10 关联剥离.....502	27.8 历史性IPv6高级API.....578
23.11 定时控制.....503	27.9 小结.....579
23.12 何时改用SCTP代替TCP.....505	习题.....579
23.13 小结.....506	第 28 章 原始套接字.....580
习题.....506	28.1 概述.....580
第 24 章 带外数据.....507	28.2 原始套接字创建.....580
24.1 概述.....507	28.3 原始套接字输出.....581
24.2 TCP带外数据.....507	28.4 原始套接字输入.....582
24.3 socketmark函数.....513	28.5 ping程序.....584
24.4 TCP带外数据小结.....519	28.6 traceroute程序.....596
24.5 客户/服务器心搏函数.....520	28.7 一个ICMP消息守护程序.....608
24.6 小结.....524	28.8 小结.....622
习题.....524	习题.....622
第 25 章 信号驱动式 I/O.....525	第 29 章 数据链路访问.....623
25.1 概述.....525	29.1 概述.....623
25.2 套接字的信号驱动式I/O.....525	29.2 BPF: BSD分组过滤器.....623
25.3 使用SIGIO的UDP回射服务器程序.....527	29.3 DLPI: 数据链路提供者接口.....625
25.4 小结.....532	29.4 Linux: SOCK_PACKET和 PF_PACKET.....626
习题.....533	29.5 libpcap: 分组捕获函数库.....627
第 26 章 线程.....534	29.6 libnet: 分组构造与输出函数库.....627
26.1 概述.....534	29.7 检查UDP的校验和字段.....628
26.2 基本线程函数: 创建和终止.....535	29.8 小结.....645
26.3 使用线程的str_cli函数.....537	习题.....645
26.4 使用线程的TCP回射服务器程序.....538	第 30 章 客户/服务器程序设计范式.....646
26.5 线程特定数据.....542	30.1 概述.....646
26.6 Web客户与同时连接.....549	30.2 TCP客户程序设计范式.....648
26.7 互斥锁.....552	30.3 TCP测试用客户程序.....649
26.8 条件变量.....555	30.4 TCP迭代服务器程序.....650
26.9 Web客户与同时连接(续).....558	

30.5	TCP并发服务器程序, 每个客户一个子进程.....	650
30.6	TCP预先派生子进程服务器程序, accept无上锁保护.....	653
30.7	TCP预先派生子进程服务器程序, accept使用文件上锁保护.....	659
30.8	TCP预先派生子进程服务器程序, accept使用线程上锁保护.....	662
30.9	TCP预先派生子进程服务器程序, 传递描述符.....	663
30.10	TCP并发服务器程序, 每个客户一个线程.....	667
30.11	TCP预先创建线程服务器程序, 每个线程各自accept.....	669
30.12	TCP预先创建线程服务器程序, 主线程统一accept.....	671
30.13	小结.....	673
	习题.....	674

第31章	流.....	675
31.1	概述.....	675
31.2	概貌.....	675
31.3	getmsg和putmsg函数.....	678
31.4	getpmsg和putpmsg函数.....	679
31.5	ioctl函数.....	680
31.6	TPI: 传输提供者接口.....	680
31.7	小结.....	689
	习题.....	689
附录A	IPv4、IPv6、ICMPv4 和 ICMPv6.....	690
附录B	虚拟网络.....	704
附录C	调试技术.....	708
附录D	杂凑的源代码.....	714
附录E	精选习题答案.....	726
参考文献	756
索引	763

第一部分

简介和TCP/IP



1.1 概述

要编写通过计算机网络通信的程序，首先要确定这些程序相互通信所用的协议（protocol）。在深入设计一个协议的细节之前，应该从高层次决断通信由哪个程序发起以及响应在何时产生。举例来说，一般认为Web服务器程序是一个长时间运行的程序（即所谓的守护程序，daemon），它只在响应来自网络的请求时才发送网络消息。协议的另一端是Web客户程序，如某种浏览器，与服务器进程的通信总是由客户进程发起。大多数网络应用就是按照划分成客户（client）和服务器（server）^①来组织的。在设计网络应用^②时，确定总是由客户发起请求往往能够简化协议和程序^③本身。当然一些较为复杂的网络应用还需要异步回调（asynchronous callback）通信，也就是由服务器向客户发起请求消息。然而坚持采纳图1-1所示的基本客户/服务器模型的网络应用毕竟要普遍得多。



图1-1 网络应用：客户和服务

通常客户每次只与一个服务器通信，不过以使用Web浏览器为例，我们也许在10分钟内就可以与许多不同的Web服务器通信。从服务器的角度来看，一个服务器同时与多个客户通信并不稀奇，见图1-2。本书后面将介绍若干种让一个服务器同时处理多个客户请求的方法。

可认为客户与服务器之间是通过某个网络协议通信的，但实际上，这样的通信通常涉及多个网络协议层。本书的焦点是TCP/IP协议族，也称为网际协议族。举例来说，Web客户与服务

① 本书英文原文通篇频繁使用client（客户）和server（服务器）这两个术语。实际上它们的具体含义随上下文而变化，有时指静态的源程序或可执行程序（客户程序和服务器程序），有时指动态进程（客户进程和服务器进程），有时指运行进程的主机（客户主机和服务器主机）。在不致引起混淆的前提下，我们简单地称客户进程为客户，称服务器进程为服务器。——译者注

② 应用（application）这个术语的具体含义随上下文而变化，有时指程序（应用程序），有时指进程（应用进程），有时作为名词性修饰词译为应用。本书有时把同处应用层的客户和服务器对也用应用表示，我们称之为应用系统、网络应用或应用。——译者注

③ Unix系统中程序（program）和进程（process）是在系统调用exec上衔接的。exec既可以由shell隐式调用（直接输入命令行执行程序属于这种情况），也可以在用户程序中显式调用。显式exec调用执行的程序在本书中称为新程序，以示与exec调用所在程序的区别。exec调用前后两个程序实际上在同一个进程环境下执行，不过往往使用新程序的名字来称呼这个进程。exec调用往往跟在某个fork调用之后，这样新程序将在新的进程环境中执行。客户程序和迭代服务器程序运行时通常只有一个进程，并发服务器程序运行时除主进程外，通常还为每个客户派生一个进程。程序和进程的密切关系使得两者有时相互渗透使用，不易区分。——译者注

器之间使用TCP (Transmission Control Protocol, 传输控制协议) 通信。TCP又转而使用IP (Internet Protocol, 网际协议) 通信, IP再通过某种形式的数据链路层通信。如果客户与服务器处于同一个以太网, 就有图1-3所示的通信层次。

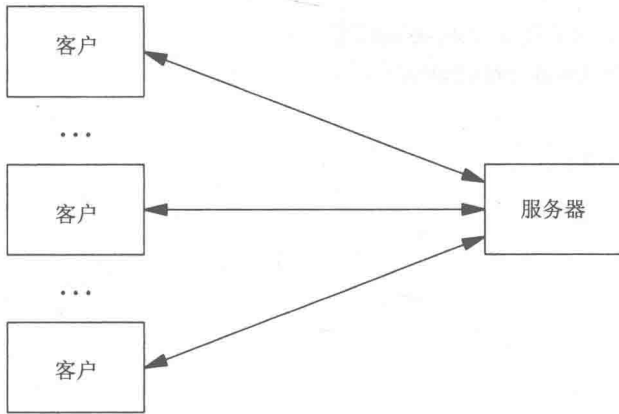


图1-2 一个服务器同时处理多个客户的请求

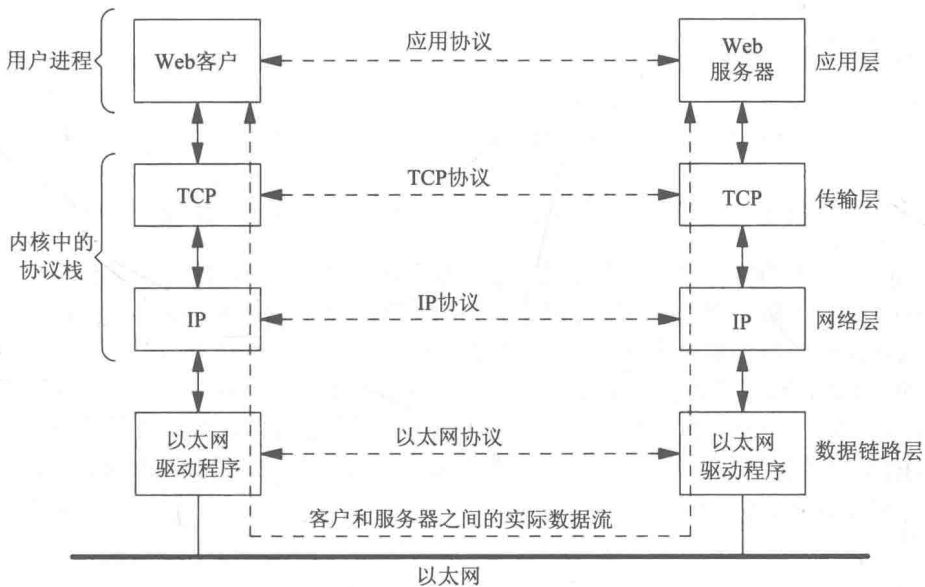


图1-3 客户与服务器使用TCP在同一个以太网中通信

尽管客户与服务器之间使用某个应用协议通信, 传输层却使用TCP通信。注意, 客户与服务器之间的信息流在其中一端是向下通过协议栈的, 跨越网络后, 在另一端则是向上通过协议栈的。另外注意, 客户和服务器通常是用户进程, 而TCP和IP协议通常是内核中协议栈的一部分。我们在图1-3右边标出了4个层。

本书讨论的协议不限于TCP和IP。有些客户和服务器改用UDP (User Datagram Protocol, 用户数据报协议) 而不是TCP, 第2章将详细介绍这两个协议。此外, 本书使用术语“IP”来称谓的那个协议, 自20世纪80年代早期以来一直在使用, 其实其正式名称是IPv4 (IP version 4, IP