大学计算机教育丛书（影印版）

UNIX NETWORK PROGRAMMING

Networking APIs: **Sockets and XTI**

Volume 1

SECOND EDITION

UNIX网络编程 卷1：

连网的API：套接字与XTI

第 2 版

W. RICHARD STEVENS

清华大学出版社 · PRENTICE HALL

http://www.tup.tsinghua.edu.cn

UNIX Network Programming
Volume 1
Second Edition
Networking APIs: Sockets and XTI

UNIX 网络编程
卷 1
连网的 API：套接字与 XTI
第 2 版

W. Richard Stevens

清华大学出版社
Prentice-Hall International, Inc.

(京) 新登字 158 号

UNIX network Programming Volume I 2nd ed.
networking APIs: Sockets and XTI/W. Richard Stevens

图书在版编目(CIP)数据

UNIX 网络程序设计:卷 I 第 2 版:英文/史蒂文斯(Stevens, W.R.)著.
-影印版.-北京:清华大学出版社,1998.7
(大学计算机教育丛书)
ISBN 7-302-02942-3

Ⅰ.U… Ⅱ.史… Ⅲ.UNIX 操作系统-程序设计-教材-英文 Ⅳ.TP316

中国版本图书馆 CIP 数据核字(98)第 09265 号

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)
http://www.tup.tsinghua.edu.cn
印刷者: 清华大学印刷厂
发行者: 新华书店总店北京发行所
开 本: 850×1168 1/32 印张: 32.5
版 次: 1998 年 7 月第 1 版 2000 年 10 月第 5 次印刷
书 号: ISBN 7-302-02942-3/TP·1554
印 数: 15001~17000
定 价: 43.00 元

Function and Macro Definitions

(**Bold** page numbers indicate source code implementation)

Function and Macro Definitions

(Bold page numbers indicate source code implementation)

出版者的话

今天，我们的大学生、研究生和教学、科研工作者，面临的是一个国际化的信息时代。他们将需要随时查阅大量的外文资料；会有更多的机会参加国际性学术交流活动；接待外国学者；走上国际会议的讲坛。作为科技工作者，他们不仅应有与国外同行进行口头和书面交流的能力，更为重要的是，他们必须具备极强的查阅外文资料获取信息的能力。有鉴于此，在国家教委所颁布的“大学英语教学大纲”中有一条规定：专业阅读应作为必修课程开设。同时，在大纲中还规定了这门课程的学时和教学要求。有些高校除开设“专业阅读”课之外，还在某些专业课拟进行英语授课。但教、学双方都苦于没有一定数量的合适的英文原版教材作为教学参考书。为满足这方面的需要，我们陆续精选了一批国外计算机科学方面最新版本的著名教材，进行影印出版。我社获得国外著名出版公司和原著作者的授权将国际先进水平的教材引入我国高等学校，为师生们提供了教学用书，相信会对高校教材改革产生积极的影响。

我们欢迎高校师生将使用影印版教材的效果，意见反馈给我们，更欢迎国内专家、教授积极向我社推荐国外优秀计算机教育教材，以利我们将《大学计算机教育丛书(影印版)》做得更好，更适合高校师生的需要。

清华大学出版社

《大学计算机教育丛书(影印版)》项目组

1999.6

Preface

Introduction

Network programming involves writing programs that communicate with other programs across a computer network. One program is normally called the *client* and the other the *server*. Most operating systems provide precompiled programs that communicate across a network—common examples in the TCP/IP world are Web clients (browsers) and Web servers, and the FTP and Telnet clients and servers—but this book describes how to write our own network programs.

We write network programs using an *application program interface* or *API*. We describe two APIs for network programming:

1. sockets, sometimes called "Berkeley sockets" acknowledging their heritage from Berkeley Unix, and
2. XTI (X/Open Transport Interface), a slight modification of the Transport Layer Interface (TLI) developed by AT&T.

All the examples in the text are from the Unix operating system, although the foundation and concepts required for network programming are, to a large degree, operating system independent. The examples are also based on the TCP/IP protocol suite, both IP versions 4 and 6.

To write network programs one must understand the underlying operating system and the underlying networking protocols. This book builds on the foundation of the my other four books in these two areas, and these books are abbreviated throughout this text as follows:

- 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], and
- TCPv3: *TCP/IP Illustrated, Volume 3* [Stevens 1996].

This second edition of *UNIX Network Programming* still contains information on both Unix and the TCP/IP protocols, but many references are made to these other four texts to allow interested readers to obtain more detailed information on various topics. This is especially the case for TCPv2, which 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 First Edition

This second edition is a complete rewrite of the first edition. These changes have been driven by the feedback I have received teaching this material about once a month during 1990–1996, and by following certain Usenet newsgroups during this same time, which lets one see the topics that are continually misunderstood. The following are the major changes with this new edition:

- This new edition uses ANSI C for all examples.
- The old Chapters 6 ("Berkeley Sockets") and 8 ("Library Routines") have been expanded into 25 chapters. Indeed this sevenfold expansion (based on a word count) of this material is probably the most significant change from the first to the second edition. Most of the individual sections in the old Chapter 6 have been expanded into an entire chapter with more examples added.
- The TCP and UDP portions from the old Chapter 6 have been separated and we now cover the TCP functions and a complete TCP client–server, followed by the UDP functions and a complete UDP client–server. This is easier for newcomers to understand than describing all the details of the `connect` function, for example, with its different semantics for TCP versus UDP.
- The old Chapter 7 ("System V Transport Layer Interface") has been expanded into seven chapters. We also cover the newer XTI instead of the TLI that it replaces.
- The old Chapter 2 ("The Unix Model") is gone. This chapter provided an overview of the Unix system in about 75 pages. In 1990 this chapter was needed because few books existed that adequately described the basic Unix programming interface, especially the differences between the Berkeley and System V implementations that existed in 1990. Today, however, more readers have a fundamental understanding of Unix, so concepts such as a process ID, password files, directories, and group IDs, need not be repeated. (My APUE book is a 700-page expansion of this material for readers desiring additional Unix programming details.)

Some of the advanced topics from the old Chapter 2 are covered in this new edition, but their coverage is moved to where the feature is used. For example, when showing our first concurrent server (Section 4.8) we cover the `fork` function. When we describe how to handle the `SIGCHLD` signal with our concurrent server (Section 5.9), we describe many additional features of Posix signal handling (zombies, interrupted system calls, etc.).

- Whenever possible this text describes the Posix interface. (We say more about the Posix family of standards in Section 1.10.) This includes not only the Posix.1 standard for the basic Unix functions (process control, signals, etc.), but also the forthcoming Posix.1g standard for the sockets and XTI networking APIs, and the 1996 Posix.1 standard for threads.

  The term "system call" has been changed to "function" when describing functions such as `socket` and `connect`. This follows the Posix convention that the distinction between a system call and a library function is an implementation detail that is often irrelevant for a programmer.

- The old Chapters 4 ("A Network Primer") and 5 ("Communication Protocols") have been replaced with Appendix A covering IP versions 4 (IPv4) and 6 (IPv6), and Chapter 2 covering TCP and UDP. This new material focuses on the protocol issues that network programmers are certain to encounter. The coverage of IPv6 was included, even though IPv6 implementations are just starting to appear, since during the lifetime of this text IPv6 will probably become the predominant networking protocol.

  I have found when teaching network programming that about 80% of all network programming problems have nothing to do with network programming, per se. That is, the problems are not with the API functions such as `accept` and `select`, but the problems arise from a lack of understanding of the underlying network protocols. For example, I have found that once a student understands TCP's three-way handshake and four-packet connection termination, many network programming problems are immediately understood.

  The old sections on XNS, SNA, NetBIOS, the OSI protocols, and UUCP have been removed, since it has become obvious during the early 1990s that these proprietary protocols have been eclipsed by the TCP/IP protocols. (UUCP is still popular and is not proprietary, but there is little we can show from a network programming perspective using UUCP.)

- The following new topics are covered in this second edition:
  - IPv4/IPv6 interoperability (Chapter 10),
  - protocol-independent name translation (Chapter 11),
  - routing sockets (Chapter 17),
  - multicasting (Chapter 19),
  - threads (Chapter 23),
  - IP options (Chapter 24),
  - datalink access (Chapter 26),

- client–server design alternatives (Chapter 27),
- virtual networks and tunneling (Appendix B), and
- network program debugging techniques (Appendix C).

Unfortunately, the coverage of the material from the first edition has been expanded so much that it no longer fits into a single book. Therefore at least two additional volumes are planned in the *UNIX Network Programming* series.

- Volume 2 will probably be subtitled *IPC: Interprocess Communication* and will be an expansion of the old Chapter 3, along with coverage of the 1996 Posix.1 realtime IPC mechanisms.
- Volume 3 will probably be subtitled *Applications* and will be an expansion of Chapters 9–18 of the first edition.

Even though most of the networking applications will be covered in Volume 3, a few special applications are covered in this volume: Ping, Traceroute, and `inetd`.

Readers

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 9) followed by whatever additional topics are of interest. Part 2 covers the basic socket functions, for both TCP and UDP, along with 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.

To aid in the use 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.

Although the sockets API has become the de facto standard for network programming, XTI is still used, sometimes with protocol suites other than TCP/IP. While the coverage of XTI in Part 4 is smaller than the coverage of sockets in Parts 2 and 3, much of the sockets coverage describes *concepts* that apply to XTI as well as sockets. For example, all of the concepts regarding the use of nonblocking I/O, broadcasting, multicasting, signal-driven I/O, out-of-band data, and threads, are the same, regardless of which API (sockets or XTI) is used. Indeed, many network programming problems are fundamentally similar, independent of whether the program is written using sockets or XTI, and there is hardly anything that can be done with one API that cannot be done with the other. The concepts are the same—just the function names and arguments change.

Source Code and Errata Availability

The source code for all the examples that appear in the book is available from `ftp://ftp.kohala.com/pub/rstevens/unpv12e.tar.gz`. 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 my home page, listed at the end of the Preface.

Acknowleagments

Supporting every author is an understanding family, or nothing would ever get written! I am grateful to my family, Sally, Bill, Ellen, and David, first for their support and understanding when I wrote my first book (the first edition of this book), and for enduring this "small" revision. Their love, support, and encouragement helped make this book possible.

Numerous reviewers provided invaluable feedback (totaling 190 printed pages or 70,000 words), catching lots of errors, pointing out areas that needed more explanation, and suggesting alternative presentations, wording, and coding: Ragnvald Blindheim, Jim Bound, Gavin Bowe, Allen Briggs, Joe Doupnik, Wu-chang Feng, Bill Fenner, Bob Friesenhahn, Andrew Gierth, Wayne Hathaway, Kent Hofer, Sugih Jamin, Scott Johnson, Rick Jones, Mukesh Kacker, Marc Lampo, Marty Leisner, Jack McCann, Craig Metz, Bob Nelson, Evi Nemeth, John C. Noble, Steve Rago, Jim Reid, Chung-Shang Shao, Ian Lance Taylor, Ron Taylor, Andreas Terzis, and Dave Thaler. A special thanks to Sugih Jamin and his students in EECS 489 ("Computer Networks") at the University of Michigan who beta tested an early draft of the manuscript during the spring of 1997.

The following people answered email questions of mine, sometimes lots of questions, which improved the accuracy and presentation of the text: Dave Butenhof, Dave Hanson, Jim Hogue, Mukesh Kacker, Brian Kernighan, Vern Paxson, Steve Rago, Dennis Ritchie, Steve Summit, Paul Vixie, John Wait, Steve Wise, and Gary Wright.

A special thanks to Larry Rafsky and the wonderful team at Gari Software for handling lots of details and for many interesting technical discussions. Thank you, Larry, for everything.

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.

- Meg McRoberts of SCO provided the latest releases of UnixWare, and Dion Johnson, Yasmin Kureshi, Michael Townsend, and Brian Ziel, provided support and answered questions.
- Mukesh Kacker of SunSoft provided access to a beta version of Solaris 2.6 and answered many questions about the Solaris TCP/IP implementation.

- Jim Bound, Matt Thomas, Mary Clouter, and Barb Glover of Digital Equipment Corp. provided an Alpha system and access to the latest IPv6 kits for Digital Unix.
- Michael Johnson of Red Hat Software provided the latest releases of Red Hat Linux.
- Steve Wise and Jessie Haug of IBM Austin provided an RS/6000 system and access to the latest IPv6 for AIX.
- Rick Jones of Hewlett-Packard provided access to a beta version of HP-UX 10.30 and he and William Gilliam answered many questions about it.

Many people helped with the Internet connectivity used throughout the text. My thanks once again to the National Optical Astronomy Observatories (NOAO), Sidney Wolff, Richard Wolff, and Steve Grandi, for providing access to their networks and hosts. Dave Siegel, Justis Addis, and Paul Lucchina answered many questions, Phil Kaslo and Jim Davis provided an MBone connection, Ran Atkinson and Pedro Marques provided a 6bone connection, and Craig Metz provided lots of DNS help.

The staff at Prentice Hall, especially my editor Mary Franz, along with Noreen Regina, Sophie Papanikolaou, and Eileen Clark, have been a wonderful asset to a writer. Many thanks for letting me do so many things "my way."

As usual, but contrary to popular fads, I produced camera-ready copy of the book using the wonderful Groff package written by James Clark. I typed in all 291,972 words using the `vi` editor, created the 201 illustrations using the `gpic` program (using many of Gary Wright's macros), produced the 81 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.

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

Tucson, Arizona
September 1997

W. Richard Stevens
rstevens@kohala.com
http://www.kohala.com/~rstevens

Contents