

<<TCP/IP详解 卷2:实现(英文版)>>

图书基本信息

书名：<<TCP/IP详解 卷2:实现(英文版)>>

13位ISBN编号：9787115222480

10位ISBN编号：7115222487

出版时间：201003

出版时间：人民邮电出版社

作者：Gary R.Wright,,W.Richard Stevens

页数：1174

字数：1440000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

本书介绍并给出了TCP / IP的常见参考实现的源代码，即由加州大学伯克利分校计算机系统研究组（CSRG）研发的实现。

历史上该实现是随4.x BSD（Berkeley Software Distribution）系统一起发布的。

最早的发布版本出现于1982年，随后经过了多次大改、多次微调，并实现了映射到其他unix及非Unix系统的众多端口。

这个实现不可小觑，它是世界范围内无数系统中日常运行的TCP / IP实现的基础。

该实现还提供了路由器功能，使我们可以展示出TCP / IP的主机实现与路由器的区别。

本书将描述该实现并给出TCP / IP核心实现的全部源代码，大约有15000行C代码。

书中介绍的Berkeley代码的版本是4.4BSD-Lite发布版本。

这个代码于1994年4月公布，它在1988年的4.3BSD Tahoe发布版本、1990年的4.3BSD Reno发布版本以及1993年的4.4BSD发布版本基础上增加了大量的网络增强技术（附录B讲述了如何获取这些源代码）。

4.4BSD发布版本提供了最新的TCP / IP特性，例如对多播（multicast）和长肥管道（10ng fat pipe）的支持（用于高带宽、长延迟的通道）。

图1-1（第4页）提供了各版本的Berkeley网络代码的更多细节。

## <<TCP/IP详解 卷2:实现(英文版)>>

### 内容概要

本书是TCP/IP领域的经典之作！

书中完整而详细地介绍了TCP/IP协议是如何实现的。

本书介绍了一个实际的TCP/IP实现，并给出了这一实现的完整源代码，帮助读者全面掌握TCP/IP的实现。

本书内容详尽且具权威性，几乎每章都提供精选的习题，并在附录中提供了部分习题的答案。

本书适合任何希望了解TCP/IP协议如何实现的读者阅读，更是TCP/IP领域研究人员和开发人员的权威参考书。

## <<TCP/IP详解 卷2:实现(英文版)>>

### 作者简介

Gary.Wright, 研究TCP / IP多年。

他是Connix公司的董事长, Cotlnix公司的总部在康涅狄格州, 主要提供Internet接入和咨询服务。

W.Richard Stevens ( 1951-1999 ) 国际知名的LJNIX和网络专家, 备受赞誉的技术作家。

生前著有《TCP / IP详解》(三卷)、《UNIX环境高级编程》和《UNIX网络编程》(两卷), 均为不朽的经典著作。

<<TCP/IP详解 卷2:实现(英文版)>>

书籍目录

Chapter 1. Introduction Chapter 2. Mbufs:Memory Buffers Chapter 3. Interface Layer Chapter 4. Interfaces:Ethernet Chapter 5. Interfaces:SLIP and Loopback Chapter 6. IP Addressing Chapter 7. Domains and Protocols Chapter 8. IP:Internet Protocol Chapter 9. IP Option Processing Chapter 10. IP Fragmentation and Reassembly Chapter 11. ICMP:Internet Control Message Protocol Chapter 12. IP Multicasting Chapter 13. IGMP:Internet Group Management Protocol Chapter 14. IP Multicast Routing Chapter 15. Socket Layer Chapter 16. Socket I/O Chapter 17. Socket Options Chapter 18. Radix Tree Routing Tables Chapter 19. Routing Requests and Routing Messages Chapter 20. Routing Sockets Chapter 21. ARP:Address Resolution Protocol Chapter 22. Protocol Control Blocks Chapter 23. UDP:User Datagram Protocol Chapter 24. TCP:Transmission Control Protocol Chapter 25. TCP Timers Chapter 26. TCP Output Chapter 27. TCP Functions Chapter 28. TCP Input Chapter 29. TCP Input(Continued) Chapter 30. TCP User Requests Chapter 31. BPF:BSD Packet Filter Chapter 32. Raw IP Epilogue Appendix A. Solutions to Selected Exercises Appendix B. Source Code Availability Appendix C. RFC 1122 Compliance Bibliography Index

## 章节摘录

When a process executes a system call such as `socket`, the kernel has access to the process table structure. The entry `pfds` in this structure points to the `filedesc` structure for the process. There are two members of this structure that interest us now: `fdofiles` is a pointer to an array of characters (the per-descriptor flags for each descriptor), and `fdofiles` is a pointer to an array of pointers to file table structures. The per-descriptor flags are 8 bits wide since only 2 bits can be set for any descriptor: the close-on-exec flag and the mapped-from-device flag. We show all these flags as `fdoflags`. We purposely call this section “Descriptors” and not “File Descriptors” since Unix descriptors can refer to lots of things other than files: sockets, pipes, directories, devices, and so on. Nevertheless, much of Unix literature uses the adjective file when talking about descriptors, which is an unnecessary qualification. Here the kernel data structure is called `filedesc` even though we were about to describe socket descriptors. We'll use the unqualified term descriptor whenever possible. The data structure pointed to by the `fdofiles` entry is shown as `files` since it is an array of pointers to file structures. The index into this array and the array of descriptor flags is the nonnegative descriptor itself: 0, 1, 2, and so on. In Figure 1.5 we show the entries for descriptors 0, 1, and 2 pointing to the same file structure at the bottom of the figure (since all three descriptors refer to our terminal). The entry for descriptor 3 points to a different file structure for our socket descriptor.

The `type` member of the file structure specifies the descriptor type as either `DTYPE_SOCKET` or `DTYPE_VNODE`. V-nodes are a general mechanism that allows the kernel to support different types of filesystems—a disk filesystem, a network filesystem (such as NFS), a filesystem on a CD-ROM, a memory-based filesystem, and so on. Our interest in this text is not with v-nodes, since TCP/IP sockets always have a type of `DTYPE_SOCKET`. The `data` member of the file structure points to either a socket structure or a vnode structure, depending on the type of descriptor. The `ops` member points to a vector of five function pointers. These function pointers are used by the `read`, `readv`, `write`, `writv`, `ioctl`, `select`, and `close` system calls, since these system calls work with either a socket descriptor or a nonsocket descriptor. Rather than look at the `type` value each time one of these system calls is invoked and then jump accordingly, the implementors chose always to jump indirectly through the corresponding entry in the `fileops` structure instead. Notationally we use a fixed-width font (`fo_read`) to show the name of a structure member and a slanted fixed-width font (`soo_read`) to show the contents of a structure member. Also note that sometimes we show the pointer to a structure arriving at the top left corner (e.g., the `filedesc` structure) and sometimes at the top right corner (e.g., both file structures and both `fileops` structures). This is to simplify the figures. Next we come to the socket structure that is pointed to by the file structure when the descriptor type is `DTYPE_SOCKET`. In our example, the socket type (`SOCK_DGRAM` for a datagram socket) is stored in the `so_type` member. An Internet protocol control block (PCB) is also allocated: an `inpcb` structure. The `so_pcb` member of the socket structure points to the `inpcb`.

## <<TCP/IP详解 卷2:实现(英文版)>>

### 编辑推荐

“我早就知道这《TCP/IP 详解 卷2：实现(英文版)》很好，但它比我之前了解的还要好。你可以在这《TCP/IP 详解·卷2：实现(英文版)》中找到任何与IP相关的信息。

” “上大学的时候就对网络充满了好奇，偶然的机，接触了《TCP / IP详解》的第1卷，立刻被它透彻的分析所深深吸引。

但总还是有很多地方不甚理解。

工作了，因为项目的需要，又曾多次拜读《TCP / IP详解》第1卷，每一次都有新的收获。

今天，购买了这本第2卷，我相信它的精彩。

” “作为一名软件程序员，我一直在寻找一本能清晰阐释网际协议工作方式的书。

《TCP / IP详解》通过逐行解释的源代码和清晰的图示，给出了关于TCP / IP如何实现的精彩且深入的细节。

这《TCP/IP 详解·卷2：实现(英文版)》绝对是每一位网络程序员以及任何对TCP / IP运行方式感兴趣的技术人员的必备读物。

” “这本书非常详尽地介绍了BSD / Linu x 系统上TCP / IP协议的实现。

对于任何对网络协议软件开发感兴趣的人来说，《TCP/IP 详解·卷2：实现(英文版)》都是必读之作、必备参考。

书中详细介绍了协议的核心实现以及基于用户的套接字API / ioctls。

无论你是初学者还是有经验的专业人士，我都向你强烈推荐这本书。

” “最近正在编写网络通信程序，这本书既可以加深我对TCP / IP协议的理解，又为我编程提供了很好的实例。

“绝对的经典！

希望所有想深入研究TCP / IP的人，都能有幸获得此书，读懂此书。

“我觉得《TCP / IP详解》真是学习TCP / IP协议的超级好书。

通俗易懂，对于初学者非常适合。

通过阅读这三本书，我对整个TCP / IP有了更深层的认识和理解。

“这本书难得如此细致。

相对于第1卷来讲，第2卷更适合做相关工作的人，或者对TCP / IP协议感兴趣的人。

读这样一本书需要耐心。

难得的好书！

” 《TCP / IP详解》是已故网络专家、著名技术作家W.Rictlard Steverls的传世之作。

内容详尽且具权威性，被誉为TCP / IP领域的不朽名著。

《TCP/IP 详解·卷2：实现(英文版)》是《TCP / IP详解》三卷本的第2卷，重点关注TCP / IP协议的实现问题。

书中介绍了一个实际的TCP / IP实现，并给出了这一实现的完整源代码，大约有15000行C代码。

此外，几乎每章都提供精选的习题。

并在附录中提供了部分习题的答案。

这一卷要求读者对TCP / IP协议的工作原理以及操作系统原理有初步的了解。

对TCP / IP协议不是很熟悉的读者应先阅读《TCP / IP详解》的第1卷。

该书对TCP / IP协议族有比较透彻的描述。

《TCP / IP详解》对于网络应用的开发人员、网络管理员以及任何想了解TCP / IP协议运行原理的人员来说，都是极好的权威参考书。

无论是初学者还是功底深厚的网络领域高手。

这套书都应是案头必备。

版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>