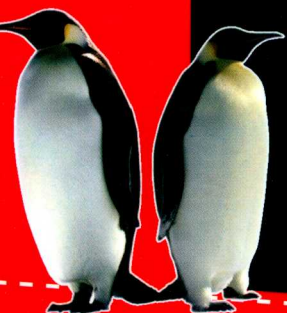


广受好评的Linux精品图书全面升级，ChinaUnix社区鼎力推荐  
资深程序员15年经验总结，深入探讨Linux应用层和内核层的网络编程  
详细讲解HTTP服务器、协议栈和防火墙三个典型案例的实际开发过程



# Linux

宋敬彬 等编著

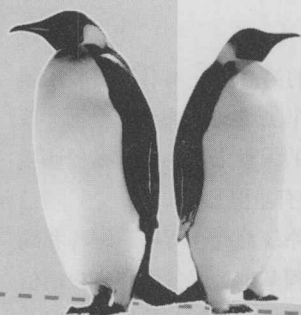
# 网络编程 (第2版)

本书源代码及PPT下载网址：[www.tup.com.cn](http://www.tup.com.cn)或[www.wanjuanchna.net](http://www.wanjuanchna.net)

- ◎ **内容全面**：涵盖Linux网络编程从基础到高级开发的方方面面知识
- ◎ **内容深入**：重点讲解了技术性较强的Linux用户空间网络编程及内核网络编程
- ◎ **注重原理**：对每个知识点都从原始概念和基本原理进行了详细和透彻的分析
- ◎ **插图丰富**：对比较复杂和难度较高的内容绘制了220余幅原理图进行讲解
- ◎ **代码经典**：书中的示例代码大多是从实际项目总结而来，有很强的实用性
- ◎ **实践性强**：结合450余个示例、70余个应用实例及3个项目案例进行讲解
- ◎ **案例典型**：详细介绍了HTTP网络服务器、协议栈和防火墙的实现过程



清华大学出版社



# Linux

宋敬彬 等编著

# 网络编程 (第2版)



浙江工业大学图书馆



7 2014984

清华大学出版社

北京

## 内 容 简 介

本书是获得大量读者好评的“Linux 典藏大系”中的《Linux 网络编程》的第2版。本书第1版出版后获得了读者的高度评价。本书循序渐进,从应用层到Linux内核,从基本知识点到综合案例,全面、系统地向读者介绍了如何在Linux下进行网络程序设计。本书涉及面广,从基本的编程工具介绍和编程环境搭建,到高级技术和核心原理,再到项目实战,几乎涉及Linux网络编程的所有重要知识。

本书共分4篇。第1篇介绍Linux操作系统概述、Linux编程环境、文件系统简介、程序、进程和线程;第2篇介绍TCP/IP协议族简介、应用层网络服务程序简介、TCP网络编程基础、服务器和客户端信息的获取、数据的IO和复用、基于UDP协议的接收和发送、高级套接字、套接字选项、原始套接字、服务器模型选择,以及IPv6的简介;第3篇介绍Linux内核中网络部分结构,以及分布和netfilter框架内报文处理;第4篇介绍三个网络编程的实例:Web服务器的例子SHTTPD、网络协议栈的例子SIP、防火墙的例子SIPFW。

本书适合所有想全面学习Linux网络编程的人员阅读,也适合已经从事Linux网络开发的工程技术人员使用。对于广大的Linux平台下的网络程序设计人员,本书更是一本不可多得的参考手册。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

Linux 网络编程/宋敬彬等编著.--2版.--北京:清华大学出版社,2014  
(Linux 典藏大系)  
ISBN 978-7-302-33528-3

I. ①L… II. ①宋… III. ①Linux 操作系统-程序设计 IV. ①TP316.89

中国版本图书馆CIP数据核字(2013)第199067号

责任编辑:夏兆彦  
封面设计:欧振旭  
责任校对:胡伟民  
责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:44

字 数:1100千字

版 次:2010年1月第1版

2014年2月第2版

印 次:2014年2月第1次印刷

印 数:6501~10500

定 价:89.00元

产品编号:050118-01

# 前 言

Linux 操作系统已经成为目前最流行的开源操作系统，在服务器、嵌入式系统有着广泛的应用，并且逐步走入个人电脑的桌面操作系统。Linux 的网络程序设在服务器领域、嵌入式领域有着广泛的应用。例如 Web 服务器、P2P 应用、嵌入式网络机顶盒、IPTV 机顶盒、手持设备等，上述产品大部分采用了开源的 Linux 系统。因此，熟悉并且能够编写网络程序代码，构建自己的网络架构程序是十分重要的。

本书是获得了大量读者好评的“Linux 典藏大系”中的《Linux 网络编程》的第 2 版。本书全面、系统地介绍了 Linux 网络编程技术，其中通过实例重点介绍了 Linux 的应用层网络设计、网络协议栈的实现原理和 Linux 内核防火墙的技术。学完本书之后，读者可以有编写比较复杂项目的本领。

## 关于“Linux 典藏大系”

“Linux 典藏大系”是清华大学出版社自 2010 年 1 月以来陆续推出的一个图书系列，截止 2013 年 1 月，已经出版了 10 余个品种。该系列图书涵盖了 Linux 技术的方方面面，可以满足各个层次和各个领域的读者学习 Linux 技术的需求。该系列图书自出版以来获得了广大读者的好评，已经成为 Linux 图书市场上最耀眼的明星品牌之一，其销量在同类图书中也名列前茅，其中一些图书还获得了“51CTO 读书频道”颁发的“最受读者喜爱的原创 IT 技术图书奖”。该系列图书出版过程中也得到了国内 Linux 领域最知名的技术社区 ChinaUnix（简称 CU）的大力支持和帮助，读者在 CU 社区中就图书的内容与活跃在 CU 社区中的 Linux 技术爱好者进行广泛交流，将会取得了良好的学习效果。

## 关于本书第 2 版

本书第 1 版出版后深受读者好评，并被 ChinaUNIX 技术社区所推荐。但是随着 Linux 技术的发展，本书第 1 版的内容与 Linux 各个新版本有一定出入，这给读者的学习造成了一些不便。应广大读者的要求，我们结合 Linux 技术的最新发展推出第 2 版图书。相比第 1 版，第 2 版图书在内容上的变化主要体现在以下几个方面：

- (1) 操作系统环境从原有的 Debian 改为更为通用的 Ubuntu。
- (2) Linux 内核介绍增加了 3.\* 系列。
- (3) 对 IT 业界的动态进行了更新。
- (4) 对一些专有名词的大小写进行了更正，如 VIM、Emacs。
- (5) 由于 Vim 区分大小写，尤其在快捷键上面。为了避免读者误操作，所以对原有的快捷键大小写进行了重新确认，并更正部分错误的大小写。
- (6) 更正了第 1 版中的部分描述错误，如 Objective-C。



- (7) 对 GCC 软件包进行了更新。
- (8) 为了便于读者阅读和使用代码, 对于完整的代码增加了行号。
- (9) 更正了部分调试选项的大小写错误。
- (10) 对部分 Shell 命令进行了更新, 如 fdisk。
- (11) 对 Linux 涉及的硬件信息进行了更新, 如对 Ext4 的支持。
- (12) 对需要重点注意的关键代码做了加粗。
- (13) 对部分代码缺少的库文件进行了补充。
- (14) 修改了部分函数库的包含关系。
- (15) 修改了部分变量的数据类型。
- (16) 修改了部分代码行号的说明错误。

## 本书的特点

### 1. 循序渐进, 由浅入深

为了方便读者学习, 本书首先介绍 Linux 的开发环境, 然后介绍基本的网路程序设计方法, 再进行 Linux 内核的网络设计方法。最后, 通过 3 个综合案例, 综合运用上述知识, 让读者更深刻地了解网络程序设计的知识。在每一部分的介绍中都是按照由浅入深的方式进行介绍, 先介绍基础知识, 再结合高级知识进行介绍。

### 2. 技术全面, 内容充实

本书基本涵盖了 Linux 网络程序设计的所有知识面, 特别对于高级网络编程、原始套接字等高级应用层网络程序设计给出了全面的介绍和丰富的例子程序。除了用户界面的网络程序设计外, 本书还对内核空间的网络程序设计进行了详细的介绍, 针对 netfilter 框架, 做了很细致的讲解, 并给出了一个全面使用 netfilter 框架的案例, 以方便读者深入了解。

### 3. 对比讲解, 理解深刻

由于 Linux 程序设计的知识用于空间和内核空间的代码和模块是相互作用的, 在多个主要函数介绍过程中, 本书对用户空间和内核空间进行交互式的对比介绍, 使读者在了解如何使用的情况下, 更深入地了解为什么这样用, 所谓“知其然并知其所以然”。

### 4. 案例精讲, 深入剖析

根据本人多年的项目经验, 只有实际接触案例和代码才能够对知识点更深入地了解。本书在介绍了 Linux 网络程序设计知识点的基础上, 通过具有典型意义的 3 个案例, 对各个知识点包括应用层的 HTTP 协议的 Web 服务器、协议栈原理的协议栈案例和内核网络的防火墙案例进行了深入剖析。

## 本书内容及体系结构

### 第 1 篇 Linux 网络开发基础 (第 1~4 章)

本篇主要内容包括: Linux 操作系统概述、Linux 编程环境、文件系统简介、程序、进程和线程。通过本篇的学习, 读者可以掌握 Linux 编程的基础知识, 以及编程环境。

## 第 2 篇 Linux 用户层网络编程（第 5~15 章）

本篇主要内容包括：TCP/IP 协议族简介、应用层网络服务程序简介、TCP 网络编程基础、服务器和客户端信息的获取、数据的 IO 和复用、基于 UDP 协议的接收和发送、高级套接字、套接字选项、原始套接字、服务器模型选择、IPv6 简介。通过本篇的学习，读者可以掌握 Linux 网络编程的大部分知识。

## 第 3 篇 Linux 内核网络编程（第 16 章和第 17 章）

本篇主要内容包括：Linux 内核中网络部分结构，以及分布和 netfilter 框架内报文处理。通过本篇的学习，读者可以初步了解 Linux 内核网络编程的知识。

## 第 4 篇 综合案例（第 18~20 章）

本篇主要内容包括：一个简单 Web 服务器的例子 SHTTPD、一个简单网络协议栈的例子 SIP、一个简单防火墙的例子 SIPFW。通过本篇的学习，读者可以全面了解一个完整可用的 Linux 网络程序是如何编写的。

## 本书学习建议

- 建议没有基础的读者，从前至后顺次阅读，尽量不要跳跃。
- 书中的实例和示例建议读者都要亲自上机动手实践，学习效果会更好。
- 第 4 篇的内容偏重于实战，这部分内容在初期可以不需要全面掌握，只要理解思想即可，等读者有了较多开发经验后可进一步研读。

## 本书读者对象

- 想全面学习 Linux 网络编程的人员；
- Linux 网络编程从业人员；
- Linux 网络编程爱好者；
- 大中专院校的学生；
- 社会培训班的学员；
- 需要一本案头必备手册的开发人员。

## 本书作者

本书由宋敬彬主笔编写。其他参与编写的人员有陈超、陈锴、陈佩霞、陈锐、黎华、李鹏钦、李森、李奕辉、李玉莉、刘仲义、卢香清、鲁木应、马向东、麦廷琮、米永刚、欧阳昉、綦彦臣、冉卫华、宋永强、滕科平、王秀丽、王玉芹、魏莹、魏宗寿、温本利。

虽然我们对书中所述的内容都尽量予以核实，并多次进行文字校对，但可能还存在疏漏和不足之处，恳请读者批评指正。

编著者

# 目 录

## 第 1 篇 Linux 网络开发基础

|                                 |    |
|---------------------------------|----|
| <b>第 1 章 Linux 操作系统概述</b> ..... | 2  |
| 1.1 Linux 发展历史 .....            | 2  |
| 1.1.1 Linux 的诞生和发展 .....        | 2  |
| 1.1.2 Linux 名称的由来 .....         | 3  |
| 1.2 Linux 的发展要素 .....           | 3  |
| 1.2.1 UNIX 操作系统 .....           | 3  |
| 1.2.2 Minix 操作系统 .....          | 4  |
| 1.2.3 POSIX 标准 .....            | 4  |
| 1.3 Linux 与 UNIX 的异同 .....      | 5  |
| 1.4 操作系统类型选择和内核版本的选择 .....      | 5  |
| 1.4.1 常见的不同公司发行的 Linux 异同 ..... | 5  |
| 1.4.2 内核版本的选择 .....             | 6  |
| 1.5 Linux 的系统架构 .....           | 7  |
| 1.5.1 Linux 内核的主要模块 .....       | 7  |
| 1.5.2 Linux 的文件结构 .....         | 8  |
| 1.6 GNU 通用公共许可证 .....           | 9  |
| 1.6.1 GPL 许可证的历史 .....          | 9  |
| 1.6.2 GPL 的自由理念 .....           | 10 |
| 1.6.3 GPL 的基本条款 .....           | 11 |
| 1.6.4 关于 GPL 许可证的争议 .....       | 12 |
| 1.7 Linux 软件开发的可借鉴之处 .....      | 12 |
| 1.8 小结 .....                    | 13 |
| <b>第 2 章 Linux 编程环境</b> .....   | 14 |
| 2.1 Linux 环境下的编辑器 .....         | 14 |
| 2.1.1 Vim 使用简介 .....            | 14 |
| 2.1.2 使用 Vim 建立文件 .....         | 15 |
| 2.1.3 使用 Vim 编辑文本 .....         | 16 |
| 2.1.4 Vim 的格式设置 .....           | 18 |
| 2.1.5 Vim 配置文件.vimrc .....      | 18 |
| 2.1.6 使用其他编辑器 .....             | 19 |

|              |                          |           |
|--------------|--------------------------|-----------|
| 2.2          | Linux 下的 GCC 编译器工具集..... | 19        |
| 2.2.1        | GCC 简介.....              | 19        |
| 2.2.2        | 编译程序的基本知识.....           | 21        |
| 2.2.3        | 单个文件编译成执行文件.....         | 21        |
| 2.2.4        | 编译生成目标文件.....            | 22        |
| 2.2.5        | 多文件编译.....               | 22        |
| 2.2.6        | 预处理.....                 | 24        |
| 2.2.7        | 编译成汇编语言.....             | 24        |
| 2.2.8        | 生成和使用静态链接库.....          | 25        |
| 2.2.9        | 生成动态链接库.....             | 26        |
| 2.2.10       | 动态加载库.....               | 29        |
| 2.2.11       | GCC 常用选项.....            | 31        |
| 2.2.12       | 编译环境的搭建.....             | 33        |
| 2.3          | Makefile 文件简介.....       | 33        |
| 2.3.1        | 一个多文件的工程例子.....          | 33        |
| 2.3.2        | 多文件工程的编译.....            | 35        |
| 2.3.3        | Makefile 的规则.....        | 37        |
| 2.3.4        | Makefile 中使用变量.....      | 39        |
| 2.3.5        | 搜索路径.....                | 42        |
| 2.3.6        | 自动推导规则.....              | 43        |
| 2.3.7        | 递归 make.....             | 44        |
| 2.3.8        | Makefile 中的函数.....       | 46        |
| 2.4          | 用 GDB 调试程序.....          | 47        |
| 2.4.1        | 编译可调试程序.....             | 48        |
| 2.4.2        | 使用 GDB 调试程序.....         | 49        |
| 2.4.3        | GDB 常用命令.....            | 52        |
| 2.4.4        | 其他的 GDB.....             | 59        |
| 2.5          | 小结.....                  | 60        |
| <b>第 3 章</b> | <b>文件系统简介.....</b>       | <b>61</b> |
| 3.1          | Linux 下的文件系统.....        | 61        |
| 3.1.1        | Linux 下文件的内涵.....        | 61        |
| 3.1.2        | 文件系统的创建.....             | 62        |
| 3.1.3        | 挂接文件系统.....              | 65        |
| 3.1.4        | 索引节点 inode.....          | 65        |
| 3.1.5        | 普通文件.....                | 66        |
| 3.1.6        | 设备文件.....                | 66        |
| 3.1.7        | 虚拟文件系统 VFS.....          | 68        |
| 3.2          | 文件的通用操作方法.....           | 72        |
| 3.2.1        | 文件描述符.....               | 72        |



|              |                             |            |
|--------------|-----------------------------|------------|
| 6.4          | NFS 协议和服务 .....             | 175        |
| 6.4.1        | 安装 NFS 服务器和客户端 .....        | 175        |
| 6.4.2        | 服务器端的设定 .....               | 176        |
| 6.4.3        | 客户端的操作 .....                | 176        |
| 6.4.4        | showmount 命令 .....          | 177        |
| 6.5          | 自定义网络服务 .....               | 177        |
| 6.5.1        | xinetd/inetd .....          | 177        |
| 6.5.2        | xinetd 服务配置 .....           | 178        |
| 6.5.3        | 自定义网络服务 .....               | 179        |
| 6.6          | 小结 .....                    | 180        |
| <b>第 7 章</b> | <b>TCP 网络编程基础 .....</b>     | <b>181</b> |
| 7.1          | 套接字编程基础知识 .....             | 181        |
| 7.1.1        | 套接字地址结构 .....               | 181        |
| 7.1.2        | 用户层和内核层交互过程 .....           | 183        |
| 7.2          | TCP 网络编程流程 .....            | 184        |
| 7.2.1        | TCP 网络编程架构 .....            | 184        |
| 7.2.2        | 创建网络插口函数 socket() .....     | 186        |
| 7.2.3        | 绑定一个地址端口对函数 bind() .....    | 189        |
| 7.2.4        | 监听本地端口 listen .....         | 192        |
| 7.2.5        | 接受一个网络请求函数 accept() .....   | 194        |
| 7.2.6        | 连接目标网络服务器函数 connect() ..... | 199        |
| 7.2.7        | 写入数据函数 write() .....        | 200        |
| 7.2.8        | 读取数据函数 read() .....         | 201        |
| 7.2.9        | 关闭套接字函数 .....               | 202        |
| 7.3          | 服务器/客户端的简单例子 .....          | 202        |
| 7.3.1        | 例子功能描述 .....                | 202        |
| 7.3.2        | 服务器网络程序 .....               | 202        |
| 7.3.3        | 服务器读取和显示字符串 .....           | 205        |
| 7.3.4        | 客户端的网络程序 .....              | 205        |
| 7.3.5        | 客户端读取和显示字符串 .....           | 206        |
| 7.3.6        | 编译运行程序 .....                | 206        |
| 7.4          | 截取信号的例子 .....               | 207        |
| 7.4.1        | 信号处理 .....                  | 207        |
| 7.4.2        | 信号 SIGPIPE .....            | 207        |
| 7.4.3        | 信号 SIGINT .....             | 208        |
| 7.5          | 小结 .....                    | 208        |
| <b>第 8 章</b> | <b>服务器和客户端信息的获取 .....</b>   | <b>209</b> |
| 8.1          | 字节序 .....                   | 209        |
| 8.1.1        | 大端字节序和小端字节序 .....           | 209        |
| 8.1.2        | 字节序转换函数 .....               | 211        |

|              |   |            |
|--------------|---|------------|
| 8.1.3        | 一个字节序转换的例子.....                         | 213        |
| 8.2          | 字符串 IP 地址和二进制 IP 地址的转换 .....            | 216        |
| 8.2.1        | inet_xxx()函数 .....                      | 216        |
| 8.2.2        | inet_pton()和 inet_ntop()函数 .....        | 218        |
| 8.2.3        | 使用 8.2.1 节地址转换函数的例子 .....               | 219        |
| 8.2.4        | 使用函数 inet_pton()和函数 inet_ntop()的例子..... | 221        |
| 8.3          | 套接字描述符判定函数 issockettype() .....         | 222        |
| 8.3.1        | 进行文件描述符判定的函数 issockettype().....        | 222        |
| 8.3.2        | main()函数 .....                          | 223        |
| 8.4          | IP 地址与域名之间的相互转换 .....                   | 223        |
| 8.4.1        | DNS 原理 .....                            | 223        |
| 8.4.2        | 获取主机信息的函数 .....                         | 224        |
| 8.4.3        | 使用主机名获取主机信息的例子 .....                    | 227        |
| 8.4.4        | 函数 gethostbyname()不可重入的例子 .....         | 229        |
| 8.5          | 协议名称处理函数 .....                          | 230        |
| 8.5.1        | xxxprotoxxx()函数 .....                   | 231        |
| 8.5.2        | 使用协议族函数的例子 .....                        | 232        |
| 8.6          | 小结 .....                                | 235        |
| <b>第 9 章</b> | <b>数据的 IO 和复用 .....</b>                 | <b>236</b> |
| 9.1          | IO 函数 .....                             | 236        |
| 9.1.1        | 使用 recv()函数接收数据 .....                   | 236        |
| 9.1.2        | 使用 send()函数发送数据 .....                   | 238        |
| 9.1.3        | 使用 readv()函数接收数据 .....                  | 239        |
| 9.1.4        | 使用 writev()函数发送数据 .....                 | 239        |
| 9.1.5        | 使用 recvmsg()函数接收数据 .....                | 241        |
| 9.1.6        | 使用 sendmsg()函数发送数据 .....                | 243        |
| 9.1.7        | IO 函数的比较 .....                          | 245        |
| 9.2          | 使用 IO 函数的例子 .....                       | 245        |
| 9.2.1        | 客户端处理框架的例子 .....                        | 245        |
| 9.2.2        | 服务器端程序框架 .....                          | 247        |
| 9.2.3        | 使用 recv()和 send()函数 .....               | 248        |
| 9.2.4        | 使用 readv()和 writev()函数 .....            | 250        |
| 9.2.5        | 使用 recvmsg()和 sendmsg()函数 .....         | 252        |
| 9.3          | IO 模型 .....                             | 255        |
| 9.3.1        | 阻塞 IO 模型 .....                          | 255        |
| 9.3.2        | 非阻塞 IO 模型 .....                         | 255        |
| 9.3.3        | IO 复用 .....                             | 256        |
| 9.3.4        | 信号驱动 IO 模型 .....                        | 256        |
| 9.3.5        | 异步 IO 模型 .....                          | 257        |

|               |                             |            |
|---------------|-----------------------------|------------|
| 9.4           | select()函数和 pselect()函数     | 258        |
| 9.4.1         | select()函数                  | 258        |
| 9.4.2         | pselect()函数                 | 260        |
| 9.5           | poll()函数和 ppoll()函数         | 261        |
| 9.5.1         | poll()函数                    | 261        |
| 9.5.2         | ppoll()函数                   | 262        |
| 9.6           | 非阻塞编程                       | 263        |
| 9.6.1         | 非阻塞方式程序设计介绍                 | 263        |
| 9.6.2         | 非阻塞程序设计的例子                  | 263        |
| 9.7           | 小结                          | 264        |
| <b>第 10 章</b> | <b>基于 UDP 协议的接收和发送</b>      | <b>265</b> |
| 10.1          | UDP 编程框架                    | 265        |
| 10.1.1        | UDP 编程框图                    | 265        |
| 10.1.2        | UDP 服务器编程框架                 | 267        |
| 10.1.3        | UDP 客户端编程框架                 | 267        |
| 10.2          | UDP 协议程序设计的常用函数             | 267        |
| 10.2.1        | 建立套接字 socket()和绑定套接字 bind() | 268        |
| 10.2.2        | 接收数据 recvfrom()/recv()      | 268        |
| 10.2.3        | 发送数据 sendto()/send()        | 273        |
| 10.3          | UDP 接收和发送数据的例子              | 277        |
| 10.3.1        | UDP 服务器端                    | 277        |
| 10.3.2        | UDP 服务器端数据处理                | 278        |
| 10.3.3        | UDP 客户端                     | 279        |
| 10.3.4        | UDP 客户端数据处理                 | 279        |
| 10.3.5        | 测试 UDP 程序                   | 280        |
| 10.4          | UDP 协议程序设计中的几个问题            | 280        |
| 10.4.1        | UDP 报文丢失数据                  | 280        |
| 10.4.2        | UDP 数据发送中的乱序                | 282        |
| 10.4.3        | UDP 协议中的 connect()函数        | 284        |
| 10.4.4        | UDP 缺乏流量控制                  | 285        |
| 10.4.5        | UDP 协议中的外出网络接口              | 287        |
| 10.4.6        | UDP 协议中的数据报文截断              | 288        |
| 10.5          | 小结                          | 289        |
| <b>第 11 章</b> | <b>高级套接字</b>                | <b>290</b> |
| 11.1          | UNIX 域函数                    | 290        |
| 11.1.1        | UNIX 域函数的地址结构               | 290        |
| 11.1.2        | 套接字函数                       | 291        |
| 11.1.3        | 使用 UNIX 域函数进行套接字编程          | 291        |
| 11.1.4        | 传递文件描述符                     | 293        |
| 11.1.5        | socketpair()函数              | 294        |

|               |   |            |
|---------------|---|------------|
| 11.1.6        | 传递文件描述符的例子.....                           | 295        |
| 11.2          | 广播.....                                   | 299        |
| 11.2.1        | 广播的 IP 地址.....                            | 300        |
| 11.2.2        | 广播与单播的比较.....                             | 300        |
| 11.2.3        | 广播的示例.....                                | 301        |
| 11.3          | 多播.....                                   | 307        |
| 11.3.1        | 多播的概念.....                                | 308        |
| 11.3.2        | 广域网的多播.....                               | 308        |
| 11.3.3        | 多播的编程.....                                | 308        |
| 11.3.4        | 内核中的多播.....                               | 310        |
| 11.3.5        | 一个多播例子的服务器端.....                          | 313        |
| 11.3.6        | 一个多播例子的客户端.....                           | 315        |
| 11.4          | 数据链路层访问.....                              | 317        |
| 11.4.1        | SOCK_PACKET 类型.....                       | 317        |
| 11.4.2        | 设置套接口以捕获链路帧的编程方法.....                     | 317        |
| 11.4.3        | 从套接口读取链路帧的编程方法.....                       | 318        |
| 11.4.4        | 定位 IP 包头的编程方法.....                        | 319        |
| 11.4.5        | 定位 TCP 报头的编程方法.....                       | 321        |
| 11.4.6        | 定位 UDP 报头的编程方法.....                       | 322        |
| 11.4.7        | 定位应用层报文数据的编程方法.....                       | 323        |
| 11.4.8        | 使用 SOCK_PACKET 编写 ARP 请求程序的例子.....        | 323        |
| 11.5          | 小结.....                                   | 326        |
| <b>第 12 章</b> | <b>套接字选项.....</b>                         | <b>328</b> |
| 12.1          | 获取和设置套接字选项 getsockopt()/setsockopt()..... | 328        |
| 12.1.1        | getsockopt()函数和 setsockopt()函数的介绍.....    | 328        |
| 12.1.2        | 套接字选项.....                                | 329        |
| 12.1.3        | 套接字选项简单示例.....                            | 330        |
| 12.2          | SOL_SOCKET 协议族选项.....                     | 334        |
| 12.2.1        | SO_BROADCAST 广播选项.....                    | 334        |
| 12.2.2        | SO_DEBUG 调试选项.....                        | 335        |
| 12.2.3        | SO_DONTROUTE 不经过路由选项.....                 | 335        |
| 12.2.4        | SO_ERROR 错误选项.....                        | 335        |
| 12.2.5        | SO_KEEPALIVE 保持连接选项.....                  | 336        |
| 12.2.6        | SO_LINGER 缓冲区处理方式选项.....                  | 337        |
| 12.2.7        | SO_OOBINLINE 带外数据处理方式选项.....              | 339        |
| 12.2.8        | SO_RCVBUF 和 SO_SNDBUF 缓冲区大小选项.....        | 340        |
| 12.2.9        | SO_RCVLOWAT 和 SO_SNDLOWAT 缓冲区下限选项.....    | 340        |
| 12.2.10       | SO_RCVTIMEO 和 SO_SNDTIMEO 收发超时选项.....     | 341        |
| 12.2.11       | SO_REUSEADDR 地址重用选项.....                  | 341        |
| 12.2.12       | SO_EXCLUSIVEADDRUSE 端口独占选项.....           | 342        |



|               |                                   |            |
|---------------|-----------------------------------|------------|
| 12.2.13       | SO_TYPE 套接字类型选项 .....             | 342        |
| 12.2.14       | SO_BSDCOMPAT 与 BSD 套接字兼容选项 .....  | 342        |
| 12.2.15       | SO_BINDTODEVICE 套接字网络接口绑定选项 ..... | 343        |
| 12.2.16       | SO_PRIORITY 套接字优先级选项 .....        | 344        |
| 12.3          | IPPROTO_IP 选项 .....               | 344        |
| 12.3.1        | IP_HDRINCL 选项 .....               | 344        |
| 12.3.2        | IP_OPTNIO 选项 .....                | 344        |
| 12.3.3        | IP_TOS 选项 .....                   | 344        |
| 12.3.4        | IP_TTL 选项 .....                   | 345        |
| 12.4          | IPPROTO_TCP 选项 .....              | 345        |
| 12.4.1        | TCP_KEEPAIVE 选项 .....             | 345        |
| 12.4.2        | TCP_MAXRT 选项 .....                | 346        |
| 12.4.3        | TCP_MAXSEG 选项 .....               | 346        |
| 12.4.4        | TCP_NODELAY 和 TCP_CORK 选项 .....   | 346        |
| 12.5          | 使用套接字选项 .....                     | 348        |
| 12.5.1        | 设置和获取缓冲区大小 .....                  | 348        |
| 12.5.2        | 获取套接字类型的例子 .....                  | 353        |
| 12.5.3        | 使用套接字选项的综合例子 .....                | 353        |
| 12.6          | ioctl()函数 .....                   | 358        |
| 12.6.1        | ioctl()函数的命令选项 .....              | 358        |
| 12.6.2        | ioctl()函数的 IO 请求 .....            | 360        |
| 12.6.3        | ioctl()函数的文件请求 .....              | 362        |
| 12.6.4        | ioctl()函数的网络接口请求 .....            | 362        |
| 12.6.5        | 使用 ioctl()函数对 ARP 高速缓存操作 .....    | 369        |
| 12.6.6        | 使用 ioctl()函数发送路由表请求 .....         | 371        |
| 12.7          | fcntl()函数 .....                   | 371        |
| 12.7.1        | fcntl()函数的选项 .....                | 372        |
| 12.7.2        | 使用 fcntl()函数修改套接字非阻塞属性 .....      | 372        |
| 12.7.3        | 使用 fcntl()函数设置信号属主 .....          | 372        |
| 12.8          | 小结 .....                          | 373        |
| <b>第 13 章</b> | <b>原始套接字 .....</b>                | <b>374</b> |
| 13.1          | 概述 .....                          | 374        |
| 13.2          | 原始套接字的创建 .....                    | 375        |
| 13.2.1        | SOCK_RAW 选项 .....                 | 375        |
| 13.2.2        | IP_HDRINCL 套接字选项 .....            | 376        |
| 13.2.3        | 不需要 bind()函数 .....                | 376        |
| 13.3          | 原始套接字发送报文 .....                   | 376        |
| 13.4          | 原始套接字接收报文 .....                   | 377        |
| 13.5          | 原始套接字报文处理时的结构 .....               | 377        |
| 13.5.1        | IP 头部的结构 .....                    | 377        |

|               |                                      |            |
|---------------|--------------------------------------|------------|
| 13.5.2        | ICMP 头部结构 .....                      | 378        |
| 13.5.3        | UDP 头部结构 .....                       | 381        |
| 13.5.4        | TCP 头部结构 .....                       | 382        |
| 13.6          | ping 的例子 .....                       | 384        |
| 13.6.1        | 协议格式 .....                           | 384        |
| 13.6.2        | 校验和函数 .....                          | 385        |
| 13.6.3        | 设置 ICMP 发送报文的头部 .....                | 386        |
| 13.6.4        | 剥离 ICMP 接受报文的头部 .....                | 387        |
| 13.6.5        | 计算时间差 .....                          | 388        |
| 13.6.6        | 发送报文 .....                           | 389        |
| 13.6.7        | 接收报文 .....                           | 390        |
| 13.6.8        | 主函数过程 .....                          | 391        |
| 13.6.9        | 主函数 main() .....                     | 393        |
| 13.6.10       | 编译测试 .....                           | 396        |
| 13.7          | 洪水攻击 .....                           | 396        |
| 13.8          | ICMP 洪水攻击 .....                      | 397        |
| 13.8.1        | ICMP 洪水攻击的原理 .....                   | 397        |
| 13.8.2        | ICMP 洪水攻击的例子 .....                   | 397        |
| 13.9          | UDP 洪水攻击 .....                       | 401        |
| 13.10         | SYN 洪水攻击 .....                       | 405        |
| 13.10.1       | SYN 洪水攻击的原理 .....                    | 405        |
| 13.10.2       | SYN 洪水攻击的例子 .....                    | 405        |
| 13.11         | 小结 .....                             | 409        |
| <b>第 14 章</b> | <b>服务器模型选择 .....</b>                 | <b>410</b> |
| 14.1          | 循环服务器 .....                          | 410        |
| 14.1.1        | UDP 循环服务器 .....                      | 410        |
| 14.1.2        | TCP 循环服务器 .....                      | 413        |
| 14.2          | 简单并发服务器 .....                        | 415        |
| 14.2.1        | 并发服务器的模型 .....                       | 416        |
| 14.2.2        | UDP 并发服务器 .....                      | 416        |
| 14.2.3        | TCP 并发服务器 .....                      | 419        |
| 14.3          | TCP 的高级并发服务器模型 .....                 | 421        |
| 14.3.1        | 单客户端单进程, 统一 accept() .....           | 422        |
| 14.3.2        | 单客户端单线程, 统一 accept() .....           | 425        |
| 14.3.3        | 单客户端单线程, 各线程独自 accept(), 使用互斥锁 ..... | 427        |
| 14.4          | IO 复用循环服务器 .....                     | 431        |
| 14.4.1        | IO 复用循环服务器模型介绍 .....                 | 431        |
| 14.4.2        | IO 复用循环服务器模型例子 .....                 | 432        |
| 14.5          | 小结 .....                             | 436        |

|                                     |     |
|-------------------------------------|-----|
| 第 15 章 IPv6 简介 .....                | 437 |
| 15.1 IPv4 的缺陷 .....                 | 437 |
| 15.2 IPv6 的特点 .....                 | 438 |
| 15.3 IPv6 的地址 .....                 | 439 |
| 15.3.1 IPv6 的单播地址 .....             | 439 |
| 15.3.2 可聚集全球单播地址 .....              | 439 |
| 15.3.3 本地使用单播地址 .....               | 440 |
| 15.3.4 兼容性地址 .....                  | 441 |
| 15.3.5 IPv6 多播地址 .....              | 441 |
| 15.3.6 IPv6 任播地址 .....              | 442 |
| 15.3.7 主机的多个 IPv6 地址 .....          | 442 |
| 15.4 IPv6 的头部 .....                 | 443 |
| 15.4.1 IPv6 头部格式 .....              | 443 |
| 15.4.2 与 IPv4 头部的对比 .....           | 444 |
| 15.4.3 IPv6 的 TCP 头部 .....          | 444 |
| 15.4.4 IPv6 的 UDP 头部 .....          | 444 |
| 15.4.5 IPv6 的 ICMP 头部 .....         | 445 |
| 15.5 IPv6 运行环境 .....                | 447 |
| 15.5.1 加载 IPv6 模块 .....             | 447 |
| 15.5.2 查看是否支持 IPv6 .....            | 447 |
| 15.6 IPv6 的结构定义 .....               | 448 |
| 15.6.1 IPv6 的地址族和协议族 .....          | 448 |
| 15.6.2 套接字地址结构 .....                | 448 |
| 15.6.3 地址兼容考虑 .....                 | 450 |
| 15.6.4 IPv6 通用地址 .....              | 450 |
| 15.7 IPv6 的套接字函数 .....              | 451 |
| 15.7.1 socket() 函数 .....            | 451 |
| 15.7.2 没有发生改变的函数 .....              | 451 |
| 15.7.3 发生改变的函数 .....                | 452 |
| 15.8 IPv6 的套接字选项 .....              | 452 |
| 15.8.1 IPv6 的套接字选项 .....            | 452 |
| 15.8.2 单播跳限 IPV6_UNICAST_HOPS ..... | 453 |
| 15.8.3 发送和接收多播包 .....               | 454 |
| 15.8.4 IPv6 中获得时间戳的 ioctl 命令 .....  | 455 |
| 15.9 IPv6 的库函数 .....                | 455 |
| 15.9.1 地址转换函数的差异 .....              | 455 |
| 15.9.2 域名解析函数的差异 .....              | 455 |
| 15.9.3 测试宏 .....                    | 458 |
| 15.10 IPv6 的编程的一个简单例子 .....         | 458 |
| 15.10.1 服务器程序 .....                 | 458 |

|         |            |     |
|---------|------------|-----|
| 15.10.2 | 客户端程序..... | 460 |
| 15.10.3 | 编译调试.....  | 461 |
| 15.11   | 小结.....    | 462 |

### 第 3 篇 Linux 内核网络编程

|               |                                 |            |
|---------------|---------------------------------|------------|
| <b>第 16 章</b> | <b>Linux 内核中网络部分结构以及分布.....</b> | <b>464</b> |
| 16.1          | 概述.....                         | 464        |
| 16.1.1        | 代码目录分布.....                     | 464        |
| 16.1.2        | 内核中网络部分流程简介.....                | 466        |
| 16.1.3        | 系统提供修改网络流程点.....                | 468        |
| 16.1.4        | sk_buff 结构.....                 | 469        |
| 16.1.5        | 网络协议数据结构 inet_protosw.....      | 471        |
| 16.2          | 软中断 CPU 报文队列及其处理.....           | 473        |
| 16.2.1        | Linux 内核网络协议层的层间传递手段——软中断.....  | 473        |
| 16.2.2        | 网络收发处理软中断的实现机制.....             | 475        |
| 16.3          | socket 数据如何在内核中接收和发送.....       | 476        |
| 16.3.1        | socket()的初始化.....               | 476        |
| 16.3.2        | 接收网络数据 recv().....              | 476        |
| 16.3.3        | 发送网络数据 send().....              | 477        |
| 16.4          | 小结.....                         | 477        |
| <b>第 17 章</b> | <b>netfilter 框架内报文处理.....</b>   | <b>478</b> |
| 17.1          | netfilter.....                  | 478        |
| 17.1.1        | netfilter 简介.....               | 478        |
| 17.1.2        | netfilter 框架.....               | 479        |
| 17.1.3        | netfilter 检查时的表格.....           | 480        |
| 17.1.4        | netfilter 的规则.....              | 480        |
| 17.2          | iptables 和 netfilter.....       | 481        |
| 17.2.1        | iptables 简介.....                | 481        |
| 17.2.2        | iptables 的表和链.....              | 481        |
| 17.2.3        | 使用 iptables 设置过滤规则.....         | 483        |
| 17.3          | 内核模块编程.....                     | 485        |
| 17.3.1        | 内核“Hello World!”程序.....         | 485        |
| 17.3.2        | 内核模块的基本架构.....                  | 487        |
| 17.3.3        | 内核模块加载和卸载过程.....                | 489        |
| 17.3.4        | 内核模块初始化和清理函数.....               | 490        |
| 17.3.5        | 内核模块初始化和清理过程的容错处理.....          | 490        |
| 17.3.6        | 内核模块编译所需的 Makefile.....         | 491        |
| 17.4          | 5 个钩子点.....                     | 492        |
| 17.4.1        | netfilter 的 5 个钩子点.....         | 492        |



|        |                          |     |
|--------|--------------------------|-----|
| 17.4.2 | NF_HOOK 宏 .....          | 493 |
| 17.4.3 | 钩子的处理规则 .....            | 494 |
| 17.5   | 注册/注销钩子 .....            | 494 |
| 17.5.1 | 结构 nf_hook_ops .....     | 494 |
| 17.5.2 | 注册钩子 .....               | 495 |
| 17.5.3 | 注销钩子 .....               | 496 |
| 17.5.4 | 注册注销函数 .....             | 497 |
| 17.6   | 钩子的简单处理例子 .....          | 498 |
| 17.6.1 | 功能描述 .....               | 498 |
| 17.6.2 | 需求分析 .....               | 498 |
| 17.6.3 | ping 回显屏蔽实现 .....        | 498 |
| 17.6.4 | 禁止向目的 IP 地址发送数据的实现 ..... | 499 |
| 17.6.5 | 端口关闭实现 .....             | 499 |
| 17.6.6 | 动态配置实现 .....             | 499 |
| 17.6.7 | 可加载内核实现代码 .....          | 501 |
| 17.6.8 | 应用层测试代码实现 .....          | 508 |
| 17.6.9 | 编译运行 .....               | 508 |
| 17.7   | 一点多个钩子的优先级 .....         | 508 |
| 17.8   | 校验和问题 .....              | 509 |
| 17.9   | 小结 .....                 | 510 |

## 第 4 篇 综合案例

|        |                               |     |
|--------|-------------------------------|-----|
| 第 18 章 | 一个简单 Web 服务器的例子 SHTTPD .....  | 512 |
| 18.1   | SHTTPD 的需求分析 .....            | 512 |
| 18.1.1 | SHTTPD 启动参数可动态配置的需求 .....     | 513 |
| 18.1.2 | SHTTPD 的多客户端支持的需求 .....       | 515 |
| 18.1.3 | SHTTPD 支持方法的需求 .....          | 515 |
| 18.1.4 | SHTTPD 支持的 HTTP 协议版本的需求 ..... | 516 |
| 18.1.5 | SHTTPD 支持头部的需求 .....          | 517 |
| 18.1.6 | SHTTPD 定位 URI 的需求 .....       | 517 |
| 18.1.7 | SHTTPD 支持 CGI 的需求 .....       | 518 |
| 18.1.8 | SHTTPD 错误代码的需求 .....          | 519 |
| 18.2   | SHTTPD 的模块分析和设计 .....         | 519 |
| 18.2.1 | SHTTPD 的主函数 .....             | 520 |
| 18.2.2 | SHTTPD 命令行解析的分析设计 .....       | 521 |
| 18.2.3 | SHTTPD 配置文件解析的分析设计 .....      | 523 |
| 18.2.4 | SHTTPD 的多客户端支持的分析设计 .....     | 523 |
| 18.2.5 | SHTTPD 头部解析的分析设计 .....        | 526 |
| 18.2.6 | SHTTPD 对 URI 的分析设计 .....      | 526 |