



Linux服务器编程领域的经典著作，资深Linux软件开发工程师撰写

从网络协议、服务器编程核心要素、原理机制、工具框架等多角度全面阐述编写高性能Linux服务器应用的方法、技巧和思想，包含两个综合性案例，极具实战意义



附赠项目源代码



游双 著

High Performance Linux Server Programming

Linux 高性能服务器编程



机械工业出版社
China Machine Press

013043418

TP316.81
604



High Performance Linux Server Programming

Linux

高性能服务器编程



游双 著



北航

C1651810



机械工业出版社
China Machine Press

TP316.81
604

813840810

图书在版编目 (CIP) 数据

Linux 高性能服务器编程 / 游双著. —北京: 机械工业出版社, 2013.5

ISBN 978-7-111-42519-9

I. L… II. 游… III. Linux 操作系统 IV. TP316.89

中国版本图书馆 CIP 数据核字 (2013) 第 101235 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书是 Linux 服务器编程领域的经典著作, 由资深 Linux 软件开发工程师撰写, 从网络协议、服务器编程核心要素、原理机制、工具框架等多角度全面阐释了编写高性能 Linux 服务器应用的方法、技巧和思想。不仅理论全面、深入, 抓住了重点和难点, 还包含两个综合性案例, 极具实战意义。

全书共 17 章, 分为 3 个部分: 第一部分对 Linux 服务器编程的核心基础——TCP/IP 协议进行了深入的解读和阐述, 包括 TCP/IP 协议族、TCP/IP 协议, 以及一个经典的 TCP/IP 通信案例; 第二部分对高性能服务器编程的核心要素进行了全面深入的剖析, 包含 Linux 网络编程 API、高级 I/O 函数、Linux 服务器程序规范、高性能服务器程序框架、I/O 复用、信号、定时器、高性能 I/O 框架库 Libevent、多进程编程、多线程编程、进程池和线程池等内容, 原理、技术与方法并重; 第三部分从侧重实战的角度讲解了高性能服务器的优化与监测, 包含服务器的调制、调试和测试, 以及各种实用系统监测工具的使用等内容。

本书另外免费赠送一个负载均衡服务器程序的完整实际项目的源代码!

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: 孙海亮

北京市荣盛彩色印刷有限公司印刷

2013 年 6 月第 1 版第 1 次印刷

186mm×240mm·22.5 印张

标准书号: ISBN 978-7-111-42519-9

定 价: 69.00 元

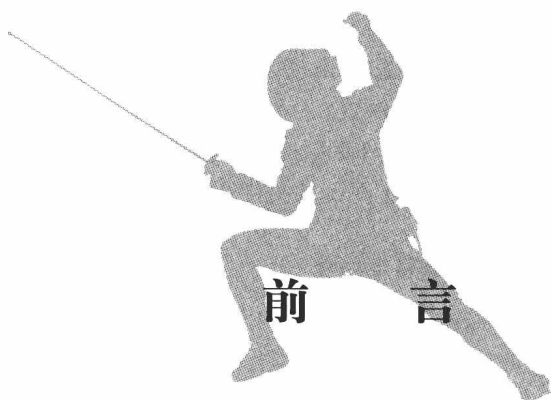
凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com



为什么要写这本书

目前国内计算机书籍的一个明显弊病就是内容宽泛而空洞。很多书籍长篇大论，恨不得囊括所有最新的技术，但连一个最基本的技术细节也无法解释清楚。有些书籍给读者展现的是网络上随处可见的知识，基本没有自己的观点，甚至连一点自己的总结都没有。反观大师们的经典书籍，整本书只专注于一个问题，而且对每个技术细节的描述都是精雕细琢。最关键的是，我们在阅读这些经典书籍时，似乎是在用心与一位编程高手交流，这绝对是一种享受。

我们把问题缩小到计算机网络编程领域。关于计算机网络编程的相关书籍，不得不提的是已故网络教育巨匠 W·Richard Stevens 先生的《TCP/IP 协议详解》（三卷本），以及《UNIX 网络编程》（两卷本）。作为一名网络程序员，即使没有阅读过这几本书，也应该听说过它们。但这几本书中的内容实在是太庞大了，没有耐心的读者根本不可能把它们全部读完。而且对于英文不太好的朋友来说，选择阅读其翻译版本又有失真原汁原味。

基于以上两点原因，笔者编写了这本《Linux 高性能服务器编程》。本书是笔者多年来学习网络编程之总结，是在充分理解大师的作品并融入自己的理解和见解后写成的。本书讨论的主题和定位很明确。简单来说就是：如何通过各种手段编写高性能的服务器程序。

网络技术是在不断向前发展的，比如 Linux 提供的 `epoll` 机制就是在内核 2.6 版本之后才正式引入的。但是，编程思想却可以享用一辈子。我们在不断学习并使用新技术，不断适应新环境的同时，书中提到的网络编程思想能让我们看得更远，想得更多。笔者相信，没有谁会认为 W·Richard Stevens 先生的网络编程书籍过时了。

读者对象

阅读本书之前，读者需要了解基本的计算机网络知识，并具有一定的 Linux 系统编程和 C++ 编程基础，否则阅读起来会有些困难。本书读者对象主要包括：

- Linux 网络应用程序开发人员
- Linux 系统程序开发人员
- C/C++ 程序开发人员
- 对网络编程技术感兴趣，或希望参与网络程序开发的人员
- 开设相关课程的大专院校师生

本书特色

本书的特点：不求内容宽泛，但求专而精，深入地剖析服务器编程的要素；不求内容精准，但求融入笔者自己的理解和观点，可谓“另眼”看服务器编程。

如何提高服务器程序性能是本书要着重讨论的。第 6、8、9、11、12、15、16 等章都用了相当的篇幅讨论这一主题。其论述方法是：首先，探讨提高服务器程序性能的一般原则，比如使用“池”以牺牲空间换取效率，使用零拷贝函数以避免内核和用户空间的切换等；其次，介绍一些高效的编程模式及其应用，比如使用有限状态机来分析用户数据，使用进程池或线程池来处理用户请求；最后，探讨如何通过调整系统参数来从服务器程序外部提高其整体性能。

光说不练假把式。如果没有实例，或者只是给出几个“Hello World”，那么本书就真没有出版的必要了。笔者要做的是让读者能真正把理论和实践完美地结合起来。在写作本书之前，笔者阅读了不少开源社区的优秀服务器软件的源代码，自己也写过相当多的小型服务器程序。这些软件中那些最精彩的部分，在书中都有充分的体现。比如第 15 章给出的两个实例——用进程池实现的简单 CGI 服务器和用线程池实现的简单 Web 服务器，就充分展现了如何利用各种提高服务器性能的手段来高效地解决实际问题。

此外，为了帮助读者进一步把书中的知识融汇到实际项目中，笔者还特意编写了一个较为完整的负载均衡服务器程序 `springsnail`。该程序能从所有逻辑服务器中选取负荷最小的一台来处理新到的客户连接。在这个程序中，使用了进程池、有限状态机、高效数据结构来提高其性能；同时，细致地封装了每个函数和模块，使之更符合实际工程项目。由于篇幅的限制，笔者未将该程序的源代码列在书中，读者可从华章网站[⊖]上下载它。

如何阅读本书

本书分为三篇：

第一篇（第 1～4 章）介绍 TCP/IP 协议族及各种重要的网络协议。只有很好地理解了底层 TCP/IP 通信的过程，才能编写出高质量的网络应用程序。毕竟，坚实的基础设施造就稳固的上层建筑。

第二篇（第 5～15 章）细致地剖析了服务器编程的各主要方面，其中对每个重要的概

⊖ 参见华章网站 www.hzbook.com。——编辑注

念、模型以及函数等都以实例代码的形式加以阐述。这一篇又可细分为如下四个部分：

- ❑ 第一部分（第 5 ~ 7 章）介绍 Linux 操作系统为网络编程提供的众多 API。这些 API 就像是基本的音符，我们通过组织它们来谱写优美的旋律。
- ❑ 第二部分（第 8 章）探讨高性能服务器程序的一般框架。在这一部分中，我们将服务器程序解构为 I/O 单元、逻辑单元和存储单元三个部件，并重点介绍了 I/O 单元、逻辑单元的几种高效实现模式。此外，我们还探讨了提高服务器性能的其他建议。
- ❑ 第三部分（第 9 ~ 12 章）深入剖析服务器程序的 I/O 单元。我们将探讨 I/O 单元需要处理的 I/O 事件、信号事件和定时事件，并介绍一款优秀的开源 I/O 框架库——Libevent。
- ❑ 第四部分（第 13 ~ 15 章）深入剖析服务器程序的逻辑单元。这一部分我们要讨论多线程、多进程编程，以及高性能逻辑处理模型——进程池和线程池，并给出相应的实例代码。

第三篇（第 16 ~ 17 章）探讨如何从系统的角度优化和监测服务器性能。本篇的内容涉及服务器程序的调制、调试和测试，以及诸多常用系统监测工具的使用。

勘误和支持

由于作者的水平有限，加之编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。书中的全部源文件都可以从华章网站下载。如果您有更多的宝贵意见或建议，也欢迎发送邮件至邮箱 pjhq87@gmail.com，期待能够得到您的真挚反馈。

致谢

首先要感谢伟大的网络教育导师 W·Richard Stevens 先生，他的书籍帮助了无数的网络程序开发人员，也给笔者指明了学习的道路。

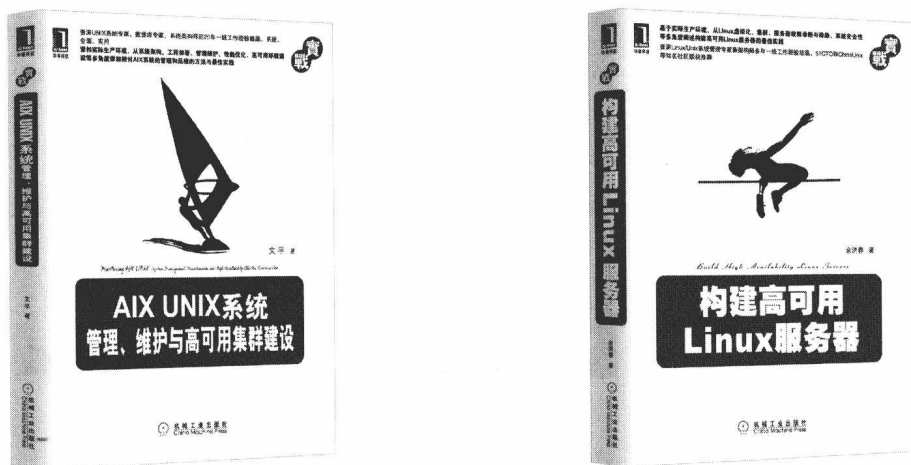
感谢机械工业出版社华章公司的编辑杨福川老师和孙海亮老师，是他们在这两年多的时间中始终支持着我的写作，因为有了他们的鼓励、帮助和引导，笔者才能顺利完成全部书稿。

感谢好友史正政，他对编程充满热爱，也无私地为本书提供了原材料。

谨以此书献给我最亲爱的家人和朋友，以及那些为计算机网络教育做出巨大贡献的大师们，还有正在为自己的未来努力拼搏、充满朝气和活力的 IT 工程师们！

游双
于北京

推荐阅读



AIX UNIX系统：管理、维护与高可用集群建设

作者：文平 ISBN: 978-7-111-35951-7 定价：79.00元

**资深UNIX系统专家、数据库专家、
系统架构师近20年一线工作经验结晶，系统、全面、实用**

**紧扣实际生产环境，从系统架构、工程部署、管理维护、性能优化、
高可用环境建设等多角度详细探讨AIX系统的管理和运维的方法与最佳实践**

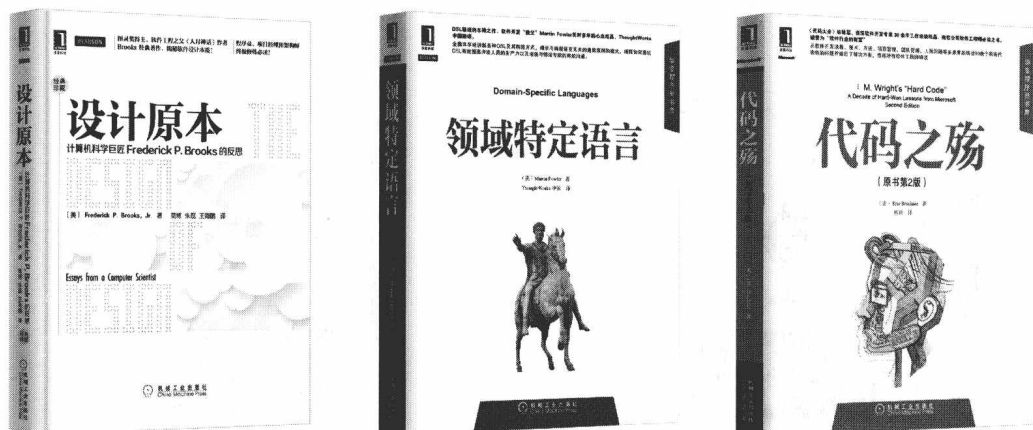
构建高可用Linux服务器

作者：余洪春 ISBN: 978-7-111-36055-1 定价：79.00元

**基于实际生产环境，从Linux虚拟化、集群、服务器故障诊断与排除、
系统安全性等多角度阐述构建高可用Linux服务器的最佳实践**

**资深Linux/Unix系统管理专家兼架构师多年一线工作经验结晶，
51CTO和ChinaUnix等知名社区联袂推荐**

推荐阅读



设计原本（精装本）

如果说《人月神话》是近40年来所有软件开发工程师和项目经理们必读的一本书，那么本书将会是未来数十年内从事软件行业的程序员、项目经理和架构师必读的一本书。它是《人月神话》作者、著名计算机科学家、软件工程教父、美国两院院士、图灵奖和IEEE计算机先驱奖得主Brooks在计算机软硬件架构与设计、建筑和组织机构的架构与设计等领域毕生经验的结晶，是计算机图书领域的又一史诗级著作。

领域特定语言

本书是DSL领域的丰碑之作，由世界级软件开发大师和软件开发“教父”Martin Fowler历时多年写作而成。全面详尽地讲解了各种DSL及其构造方式，揭示了与编程语言无关的通用原则和模式，阐释了如何通过DSL有效提高开发人员的生产力以及增进与领域专家的有效沟通，能为开发人员选择和使用DSL提供有效的决策依据和指导方法。

代码之殇

本书是《代码大全》的姊妹篇，被誉为“软件行业的财富”，资深软件开发专家30余年工作经验的结晶，是微软公司软件工程师必读之书。它从软件开发流程、技术、方法、项目管理、团队管理、人际沟通等多角度总结出90余个具有代表性的问题（大多数问题都可能给公司或软件项目带来毁灭性灾难），并给出了问题的解决方案和最佳实践，值得所有软件开发工程师和项目管理者研读。



北航

C1651810



前言

第一篇 TCP/IP协议详解

第1章 TCP/IP协议族 / 2

- 1.1 TCP/IP协议族体系结构以及主要协议 / 2
 - 1.1.1 数据链路层 / 2
 - 1.1.2 网络层 / 3
 - 1.1.3 传输层 / 4
 - 1.1.4 应用层 / 5
- 1.2 封装 / 6
- 1.3 分用 / 7
- 1.4 测试网络 / 8
- 1.5 ARP协议工作原理 / 9
 - 1.5.1 以太网ARP请求/应答报文详解 / 9
 - 1.5.2 ARP高速缓存的查看和修改 / 10
 - 1.5.3 使用tcpdump观察ARP通信过程 / 10
- 1.6 DNS工作原理 / 12
 - 1.6.1 DNS查询和应答报文详解 / 12
 - 1.6.2 Linux下访问DNS服务 / 14
 - 1.6.3 使用tcpdump观察DNS通信过程 / 15
- 1.7 socket和TCP/IP协议族的关系 / 16

第2章 IP协议详解 / 17

- 2.1 IP服务的特点 / 17
- 2.2 IPv4头部结构 / 18
 - 2.2.1 IPv4头部结构 / 18
 - 2.2.2 使用tcpdump观察IPv4头部结构 / 20
- 2.3 IP分片 / 21
- 2.4 IP路由 / 22
 - 2.4.1 IP模块工作流程 / 23
 - 2.4.2 路由机制 / 24
 - 2.4.3 路由表更新 / 25
- 2.5 IP转发 / 25
- 2.6 重定向 / 26
 - 2.6.1 ICMP重定向报文 / 26
 - 2.6.2 主机重定向实例 / 27
- 2.7 IPv6头部结构 / 27
 - 2.7.1 IPv6固定头部结构 / 28
 - 2.7.2 IPv6扩展头部 / 29

第3章 TCP协议详解 / 30

- 3.1 TCP服务的特点 / 30
- 3.2 TCP头部结构 / 32
 - 3.2.1 TCP固定头部结构 / 32
 - 3.2.2 TCP头部选项 / 33
 - 3.2.3 使用tcpdump观察TCP头部信息 / 35
- 3.3 TCP连接的建立和关闭 / 37
 - 3.3.1 使用tcpdump观察TCP连接的建立和关闭 / 37
 - 3.3.2 半关闭状态 / 39
 - 3.3.3 连接超时 / 39
- 3.4 TCP状态转移 / 40
 - 3.4.1 TCP状态转移总图 / 41
 - 3.4.2 TIME_WAIT状态 / 43
- 3.5 复位报文段 / 44
 - 3.5.1 访问不存在的端口 / 44
 - 3.5.2 异常终止连接 / 45

- 3.5.3 处理半打开连接 / 45
- 3.6 TCP交互数据流 / 46
- 3.7 TCP成块数据流 / 48
- 3.8 带外数据 / 50
- 3.9 TCP超时重传 / 51
- 3.10 拥塞控制 / 53
 - 3.10.1 拥塞控制概述 / 53
 - 3.10.2 慢启动和拥塞避免 / 54
 - 3.10.3 快速重传和快速恢复 / 55

第4章 TCP/IP通信案例：访问Internet上的Web服务器 / 57

- 4.1 实例总图 / 57
- 4.2 部署代理服务器 / 58
 - 4.2.1 HTTP代理服务器的工作原理 / 58
 - 4.2.2 部署squid代理服务器 / 59
- 4.3 使用tcpdump抓取传输数据包 / 60
- 4.4 访问DNS服务器 / 62
- 4.5 本地名称查询 / 63
- 4.6 HTTP通信 / 64
 - 4.6.1 HTTP请求 / 65
 - 4.6.2 HTTP应答 / 66
- 4.7 实例总结 / 68

第二篇 深入解析高性能服务器编程

第5章 Linux网络编程基础API / 70

- 5.1 socket地址API / 70
 - 5.1.1 主机字节序和网络字节序 / 70
 - 5.1.2 通用socket地址 / 71
 - 5.1.3 专用socket地址 / 72
 - 5.1.4 IP地址转换函数 / 73
- 5.2 创建socket / 74
- 5.3 命名socket / 75
- 5.4 监听socket / 76

- 5.5 接受连接 / 78
- 5.6 发起连接 / 80
- 5.7 关闭连接 / 80
- 5.8 数据读写 / 81
 - 5.8.1 TCP数据读写 / 81
 - 5.8.2 UDP数据读写 / 85
 - 5.8.3 通用数据读写函数 / 86
- 5.9 带外标记 / 87
- 5.10 地址信息函数 / 87
- 5.11 socket选项 / 87
 - 5.11.1 SO_REUSEADDR选项 / 89
 - 5.11.2 SO_RCVBUF和SO_SNDBUF选项 / 89
 - 5.11.3 SO_RCVLOWAT和SO_SNDLOWAT选项 / 93
 - 5.11.4 SO_LINGER选项 / 93
- 5.12 网络信息API / 94
 - 5.12.1 gethostbyname和gethostbyaddr / 94
 - 5.12.2 getservbyname和getservbyport / 95
 - 5.12.3 getaddrinfo / 96
 - 5.12.4 getnameinfo / 98

第6章 高级I/O函数 / 100

- 6.1 pipe函数 / 100
- 6.2 dup函数和dup2函数 / 101
- 6.3 readv函数和writev函数 / 103
- 6.4 sendfile函数 / 106
- 6.5 mmap函数和munmap函数 / 107
- 6.6 splice函数 / 108
- 6.7 tee函数 / 110
- 6.8 fcntl函数 / 112

第7章 Linux服务器程序规范 / 114

- 7.1 日志 / 114
 - 7.1.1 Linux系统日志 / 114
 - 7.1.2 syslog函数 / 115
- 7.2 用户信息 / 116

- 7.2.1 UID、EUID、GID和EGID / 116
- 7.2.2 切换用户 / 117
- 7.3 进程间关系 / 118
 - 7.3.1 进程组 / 118
 - 7.3.2 会话 / 118
 - 7.3.3 用ps命令查看进程关系 / 119
- 7.4 系统资源限制 / 119
- 7.5 改变工作目录和根目录 / 120
- 7.6 服务器程序后台化 / 121

第8章 高性能服务器程序框架 / 123

- 8.1 服务器模型 / 123
 - 8.1.1 C/S模型 / 123
 - 8.1.2 P2P模型 / 124
- 8.2 服务器编程框架 / 125
- 8.3 I/O模型 / 126
- 8.4 两种高效的事件处理模式 / 127
 - 8.4.1 Reactor模式 / 128
 - 8.4.2 Proactor模式 / 128
 - 8.4.3 模拟Proactor模式 / 129
- 8.5 两种高效的并发模式 / 130
 - 8.5.1 半同步/半异步模式 / 131
 - 8.5.2 领导者/追随者模式 / 134
- 8.6 有限状态机 / 136
- 8.7 提高服务器性能的其他建议 / 144
 - 8.7.1 池 / 144
 - 8.7.2 数据复制 / 145
 - 8.7.3 上下文切换和锁 / 145

第9章 I/O复用 / 146

- 9.1 select系统调用 / 146
 - 9.1.1 select API / 146
 - 9.1.2 文件描述符就绪条件 / 148
 - 9.1.3 处理带外数据 / 148

- 9.2 poll系统调用 / 150
- 9.3 epoll系列系统调用 / 151
 - 9.3.1 内核事件表 / 151
 - 9.3.2 epoll_wait函数 / 152
 - 9.3.3 LT和ET模式 / 153
 - 9.3.4 EPOLLONESHOT事件 / 157
- 9.4 三组I/O复用函数的比较 / 161
- 9.5 I/O复用的高级应用一：非阻塞connect / 162
- 9.6 I/O复用的高级应用二：聊天室程序 / 165
 - 9.6.1 客户端 / 165
 - 9.6.2 服务器 / 167
- 9.7 I/O复用的高级应用三：同时处理TCP和UDP服务 / 171
- 9.8 超级服务xinetd / 175
 - 9.8.1 xinetd配置文件 / 175
 - 9.8.2 xinetd工作流程 / 176

第10章 信号 / 178

- 10.1 Linux信号概述 / 178
 - 10.1.1 发送信号 / 178
 - 10.1.2 信号处理方式 / 179
 - 10.1.3 Linux信号 / 179
 - 10.1.4 中断系统调用 / 181
- 10.2 信号函数 / 181
 - 10.2.1 signal系统调用 / 181
 - 10.2.2 sigaction系统调用 / 181
- 10.3 信号集 / 182
 - 10.3.1 信号集函数 / 182
 - 10.3.2 进程信号掩码 / 183
 - 10.3.3 被挂起的信号 / 183
- 10.4 统一事件源 / 184
- 10.5 网络编程相关信号 / 188
 - 10.5.1 SIGHUP / 188
 - 10.5.2 SIGPIPE / 189
 - 10.5.3 SIGURG / 190

第11章 定时器 / 193

- 11.1 socket选项SO_RCVTIMEO和SO_SNDTIMEO / 193
- 11.2 SIGALRM信号 / 195
 - 11.2.1 基于升序链表的定时器 / 195
 - 11.2.2 处理非活动连接 / 200
- 11.3 I/O复用系统调用的超时参数 / 205
- 11.4 高性能定时器 / 206
 - 11.4.1 时间轮 / 206
 - 11.4.2 时间堆 / 211

第12章 高性能I/O框架库Libevent / 218

- 12.1 I/O框架库概述 / 218
- 12.2 Libevent源码分析 / 220
 - 12.2.1 一个实例 / 220
 - 12.2.2 源代码组织结构 / 222
 - 12.2.3 event结构体 / 224
 - 12.2.4 往注册事件队列中添加事件处理器 / 226
 - 12.2.5 往事件多路分发器中注册事件 / 230
 - 12.2.6 eventop结构体 / 233
 - 12.2.7 event_base结构体 / 235
 - 12.2.8 事件循环 / 236

第13章 多进程编程 / 239

- 13.1 fork系统调用 / 239
- 13.2 exec系列系统调用 / 240
- 13.3 处理僵尸进程 / 240
- 13.4 管道 / 241
- 13.5 信号量 / 243
 - 13.5.1 信号量原语 / 243
 - 13.5.2 semget系统调用 / 244
 - 13.5.3 semop系统调用 / 245
 - 13.5.4 semctl系统调用 / 247
 - 13.5.5 特殊键值IPC_PRIVATE / 249
- 13.6 共享内存 / 251

- 13.6.1 shmget系统调用 / 251
- 13.6.2 shmat和shmdt系统调用 / 252
- 13.6.3 shmctl系统调用 / 253
- 13.6.4 共享内存的POSIX方法 / 254
- 13.6.5 共享内存实例 / 254
- 13.7 消息队列 / 263
 - 13.7.1 msgget系统调用 / 263
 - 13.7.2 msgsnd系统调用 / 264
 - 13.7.3 msgrcv系统调用 / 264
 - 13.7.4 msgctl系统调用 / 265
- 13.8 IPC命令 / 266
- 13.9 在进程间传递文件描述符 / 267

第14章 多线程编程 / 269

- 14.1 Linux线程概述 / 269
 - 14.1.1 线程模型 / 269
 - 14.1.2 Linux线程库 / 270
- 14.2 创建线程和结束线程 / 271
- 14.3 线程属性 / 273
- 14.4 POSIX信号量 / 275
- 14.5 互斥锁 / 276
 - 14.5.1 互斥锁基础API / 276
 - 14.5.2 互斥锁属性 / 277
 - 14.5.3 死锁举例 / 278
- 14.6 条件变量 / 279
- 14.7 线程同步机制包装类 / 280
- 14.8 多线程环境 / 282
 - 14.8.1 可重入函数 / 282
 - 14.8.2 线程和进程 / 283
 - 14.8.3 线程和信号 / 284

第15章 进程池和线程池 / 287

- 15.1 进程池和线程池概述 / 287
- 15.2 处理多客户 / 288
- 15.3 半同步/半异步进程池实现 / 289

- 15.4 用进程池实现的简单CGI服务器 / 298
- 15.5 半同步/半反应堆线程池实现 / 301
- 15.6 用线程池实现的简单Web服务器 / 304
 - 15.6.1 http_conn类 / 304
 - 15.6.2 main函数 / 318

第三篇 高性能服务器优化与监测

第16章 服务器调制、调试和测试 / 324

- 16.1 最大文件描述符数 / 324
- 16.2 调整内核参数 / 325
 - 16.2.1 /proc/sys/fs目录下的部分文件 / 325
 - 16.2.2 /proc/sys/net目录下的部分文件 / 325
- 16.3 gdb调试 / 326
 - 16.3.1 用gdb调试多进程程序 / 326
 - 16.3.2 用gdb调试多线程程序 / 328
- 16.4 压力测试 / 329

第17章 系统监测工具 / 333

- 17.1 tcpdump / 333
- 17.2 lsof / 334
- 17.3 nc / 336
- 17.4 strace / 338
- 17.5 netstat / 341
- 17.6 vmstat / 342
- 17.7 ifstat / 344
- 17.8 mpstat / 344

参考文献 / 346