

TCP Sockets编程

Working with TCP Sockets

[加] Jesse Storimer 著 门佳 译

- 短小精悍的Sockets编程手册
- 代码示例详尽，供你举一反三
- 利用Ruby的语法糖实现高效构建



TP393.027

20142

TCP Sockets编程

Working with TCP Sockets

[加] Jesse Storimer 著 门佳 译



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (C I P) 数据

TCP Sockets编程 / (加) 斯托里默 (Storimer, J.)
著 ; 门佳译. — 北京 : 人民邮电出版社, 2013.10
(图灵程序设计丛书)
书名原文: Working with TCP Sockets
ISBN 978-7-115-33052-9

I. ①T… II. ①斯… ②门… III. ①计算机网络—程
序设计 IV. ①TP393.09

中国版本图书馆CIP数据核字(2013)第215050号

内 容 提 要

本书通过循序渐进的方式,从最基础的概念到高级别的 Ruby 封装器,再到更复杂的应用,提供了开发成熟且功能强大的应用程序所必备的知识 and 技巧,帮助读者掌握在 Ruby 语言环境下,用套接字实现项目开发的任务和技术。

本书适合对 TCP 套接字感兴趣的读者阅读。

◆ 著 [加] Jesse Storimer

译 门 佳

责任编辑 丁晓昀

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京天宇星印刷厂印刷

◆ 开本: 880×1230 1/32

印张: 5.125

字数: 125千字

2013年10月第1版

印数: 1-3 500册

2013年10月北京第1次印刷

著作权合同登记号 图字: 01-2013-3891号

定价: 29.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

站在巨人的肩上
Standing on Shoulders of Giants



www.ituring.com.cn

版 权 声 明

Copyright © 2012 Jesse Storimer. Original English language edition, entitled *Working with TCP Sockets*.

Simplified Chinese-language edition copyright © 2013 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由The Pragmatic Programmers, LLC.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。

版权所有，侵权必究。

前言

套接字 (socket) 连接起了数字世界。

回想一下计算机的早年吧。那时候它是专供科学家使用的器物，他们用它来做数学运算以及模拟，那真是阳春白雪的东西啊。

计算机真正将普通人相互联系起来，已经是多年之后的事了。如今，计算机更多是由普罗大众在使用，科学家占的比例很小。人们能够随时随地同他人共享信息、互相交流，计算机就越发变得引人入胜。

正是网络编程——更确切地说，是一组特定的套接字编程API——的出现，才使得这一切成真。正在阅读本书的你可能每天都同他人进行在线联系，每天都在使用这些由计算机互联的想法所催生的技术。

所以说网络编程归根结底是关于共享与通信的。本书的目的在于使你更深入地理解网络编程的底层机制，为网络互联贡献出自己的一份力量。

我的故事

我仍记得与套接字初次相遇时的情景。嗯，那可实在算不上美好。

作为Web开发人员，我使用过各种HTTP API，也已经习惯了诸如REST、JSON这类高层概念。

后来，我不得不去集成一个域名注册API。

我拿到API文档就蒙了。文档中要求在某些私有主机名的随机端口上打开一个TCP套接字。这和Twitter API的工作方式一点都不一样！

文档不仅要求建立TCP套接字，而且并没有将数据编码为JSON，甚至连XML都不是。我必须使用它们自己的那一套基于行的协议（line protocol）。通过套接字发送专门格式化过的文本行，然后发送一个空行，接着是用于参数的一对键-值，最后跟上两个空行表明请求发送完毕。

随后还得用同样的方式读入响应的内容。当时我就在想：“这搞的是哪出啊……”

我把这些东西拿给了我的同事看，他的反应和我一样。他也从来没用过这种API。他立刻提醒我道：“我以前只在C语言中用过套接字。你可得小心点。在退出前一定得把套接字关闭，不然它就会一直处于打开状态。程序退出后，就很难将其关闭了。”

什么?! 一直打开? 协议? 端口? 我懵了。

后来另一位同事看了一眼，然后说：“不是真的吧? 你不知道怎么用套接字? 要知道每次读取Web页面时，就是在使用套接字啊。你真应

该了解一下它的工作原理。”

我将此视为一次挑战。一下子理清这些概念实在有些吃不消，不过我也要全力以赴。尽管没少出错，但最终还是把它搞定了。就套接字而言，我对它的运用比以前更上了一层楼。对于工作中所依赖的那些技术也有了更深入的理解。这种感觉还真是不错。

借助于本书，我希望能够帮你减少一些我自己在初识套接字时所经历的烦恼，同时让你在深刻领悟这一系列技术后享受到神清气爽的感觉。

本书的读者对象

这本书的目标读者是工作在Unix或类Unix系统平台上的Ruby开发人员。

本书的读者应该熟悉Ruby，但不必掌握网络编程的相关概念，我会从网络编程的基础讲起。

本书所有实例代码使用Ruby 1.9编写，并没有在更早的版本上测试过。

本书的内容

本书包括三个主要部分。

第一部分介绍套接字编程的基础知识。你可以学到如何创建套接字，如何连接套接字以及如何共享数据。

第二部分阐述一些套接字编程的高级主题。在你学会了“Hello world”式的套接字编程之后，还需要掌握这些内容。

第三部分在真实场景中运用前两部分中学到的知识。这部分会教你如何在网络程序中使用并发技术。对于同一个问题，我们会采用多个架构模式解决，并对这些模式进行比较。

Berkeley 套接字 API

本书主要关注的是Berkeley套接字API及其用法。Berkeley套接字API最先于1983年出现在BSD操作系统4.2版本中。该操作系统是当时刚刚提出的TCP的首个实现。

Berkeley套接字API真正经受住了时间的检验。本书中所使用的API和被大多数现代编程语言所支持的API同1983年时的那套API如出一辙。

毫无疑问，Berkeley套接字API之所以能够屹立不倒的一个关键原因就是：你可以在无需了解底层协议的情况下使用套接字。这一点至关重要，我们会在后面详细探讨。

Berkeley套接字API是一种编程API，运作在实际的协议实现之上。它关注的是连接两个端点（endpoint）共享数据，而非处理分组和序列号。

已成业界标准的Berkeley套接字API是由C语言实现的，几乎所有用C编写的现代编程语言都会包含它的低层次接口。因此，我尽力使本书的大部分内容具有普适性。

也就是说，我并不是仅仅演示Ruby所提供的套接字API的包装类（wrapper class），而是先讲解低层次的API，然后再介绍Ruby的包装类，使你对套接字的理解不仅仅局限在Ruby中。

当使用其他的编程语言时，你仍然可以运用这里所学到的基础知识，利用低层次结构实现所需要的一切。

本书没有讲述的内容

之前我提到过Berkeley套接字API的优点就是你无需了解任何底层协议的细节。本书也会彻底贯彻这一点。

有些有关网络互联的书籍重点关注底层协议及其错综复杂的细节，甚至会在诸如UDP协议或是原始套接字上重新实现TCP。本书可不会讲这些内容。

我们采纳了Berkeley套接字API所奉行的观点，即使用者无需了解底层协议的实现。本书将重点放在如何使用API实现有用的功能，并尽可能关注如何完成实战任务。

不过有时候（例如从事性能优化），不够了解底层协议会使你无法正确地使用某种特性。这种情况下，我会解释一些必要的细节，以帮助你理解某些概念。

将话题再转回到协议上。我已经说过不会详细讨论TCP，同样也不会详细介绍其他诸如HTTP、FTP等应用层协议。在随后的章节中，我会在示例中用到一些协议，但并不会深入探讨它们。

如果你对协议感兴趣，我推荐Stevens的著作《TCP/IP详解》(*TCP/IP Illustrated*)^①。

netcat

书中有很多地方都使用netcat工具创建了一些随意的连接，用来测试我们编写的各色程序。netcat(在终端下通常是nc)是一个Unix工具，可以用于创建TCP(以及UDP)连接并进行侦听。在使用套接字时，它可是你工具箱中必备的一件利器。

如果你的工作平台是Unix系统，那么netcat可能已经安装好了，运行本书的示例应该不会碰上什么问题。

致谢

首先我要感谢我的家人Sara和Inara。虽然她们并没有参与本书的编写，但是却以其独特的方式作出了贡献：给予我时间和空间从事写作，提醒我重要的事项。如果不是她们，此书断然不能成形。

接下来要感谢出色的审稿人。他们阅读了草稿，对每一页都提出了想法和意见，这一切都增进了本书的质量。非常感谢Jonathan Rudenberg、Henrik Nyh、Cody Fauser、Julien Boyer、Joshua Wehner、Mike Perham、Camilo Lopez、Pat Shaughnessy、Trevor Bramble、Ryan LeCompte、Joe James、Michael Bernstein、Jesus Castello以及Pradepto Bhattacharya。

^① <http://www.amazon.com/TCP-Illustrated-Vol-Addison-Wesley-Professional/dp/0201633469>.

目 录

第 1 章 建立套接字	1
1.1 Ruby 的套接字库	1
1.2 创建首个套接字	1
1.3 什么是端点	2
1.4 环回地址	3
1.5 IPv6	3
1.6 端口	4
1.7 创建第二个套接字	5
1.8 文档	6
1.9 本章涉及的系统调用	7
第 2 章 建立连接	8
第 3 章 服务器生命周期	9
3.1 服务器绑定	9
3.1.1 该绑定到哪个端口	10
3.1.2 该绑定到哪个地址	11
3.2 服务器侦听	12
3.2.1 侦听队列	13

3.2.2	侦听队列的长度	13
3.3	接受连接	14
3.3.1	以阻塞方式接受连接	15
3.3.2	accept 调用返回一个数组	15
3.3.3	连接类	17
3.3.4	文件描述符	17
3.3.5	连接地址	18
3.3.6	accept 循环	18
3.4	关闭服务器	19
3.4.1	退出时关闭	19
3.4.2	不同的关闭方式	20
3.5	Ruby 包装器	22
3.5.1	服务器创建	22
3.5.2	连接处理	24
3.5.3	合而为一	25
3.6	本章涉及的系统调用	25
第 4 章	客户端生命周期	27
4.1	客户端绑定	28
4.2	客户端连接	28
4.3	Ruby 包装器	30
4.4	本章涉及的系统调用	32
第 5 章	交换数据	33
第 6 章	套接字读操作	36
6.1	简单的读操作	36
6.2	没那么简单	37
6.3	读取长度	38
6.4	阻塞的本质	39

6.5	EOF 事件	39
6.6	部分读取	41
6.7	本章涉及的系统调用	43
第 7 章	套接字写操作	44
第 8 章	缓冲	45
8.1	写缓冲	45
8.2	该写入多少数据	46
8.3	读缓冲	47
8.4	该读取多少数据	47
第 9 章	第一个客户端/服务器	49
9.1	服务器	49
9.2	客户端	51
9.3	投入运行	52
9.3	分析	52
第 10 章	套接字选项	54
10.1	SO_TYPE	54
10.2	SO_REUSE_ADDR	55
10.3	本章涉及的系统调用	56
第 11 章	非阻塞式 IO	57
11.1	非阻塞式读操作	57
11.2	非阻塞式写操作	60
11.3	非阻塞式接收	62
11.4	非阻塞式连接	63
第 12 章	连接复用	65
12.1	select(2)	66

12.2 读/写之外的事件	68
12.2.1 EOF	69
12.2.2 accept	69
12.2.3 connect	69
12.3 高性能复用	72
第 13 章 Nagle 算法	74
第 14 章 消息划分	76
14.1 使用新行	77
14.2 使用内容长度	79
第 15 章 超时	81
15.1 不可用的选项	81
15.2 IO.select	82
15.3 接受超时	83
15.4 连接超时	83
第 16 章 DNS 查询	85
第 17 章 SSL 套接字	87
第 18 章 紧急数据	92
18.1 发送紧急数据	93
18.2 接受紧急数据	93
18.3 局限	94
18.4 紧急数据和 IO.select	95
18.5 SO_OOBINLINE 选项	96
第 19 章 网络架构模式	97

第 20 章	串行化	101
20.1	讲解	101
20.2	实现	101
20.3	思考	105
第 21 章	单连接进程	107
21.1	讲解	107
21.2	实现	108
21.3	思考	111
21.4	案例	111
第 22 章	单连接线程	112
22.1	讲解	112
22.2	实现	113
22.3	思考	116
22.4	案例	117
第 23 章	Preforking	118
23.1	讲解	118
23.2	实现	119
23.3	思考	123
23.4	案例	124
第 24 章	线程池	125
24.1	讲解	125
24.2	实现	125
24.3	思考	129
24.4	案例	130
第 25 章	事件驱动	131

25.1 讲解	131
25.2 实现	133
25.3 思考	140
25.4 案例	142
第 26 章 混合模式	143
26.1 nginx	143
26.2 Puma	144
26.3 EventMachine	145
第 27 章 结语	147