



HZ BOOKS
华章科技

PEARSON

Windows Sockets网络编程经典著作，Windows Sockets 2.0规范解释小组负责人亲自执笔，Windows Sockets工作组主席鼎力推荐

对Windows Sockets规范进行深刻解读，系统阐述Windows Sockets网络编程的核心概念、工作原理、操作模式以及开发方法和技巧

华章程序员书库

Windows Sockets Network Programming

Windows Sockets 网络编程

(美) Bob Quinn Dave Shute 著

徐磊 腾婧 张莹 何慧 译



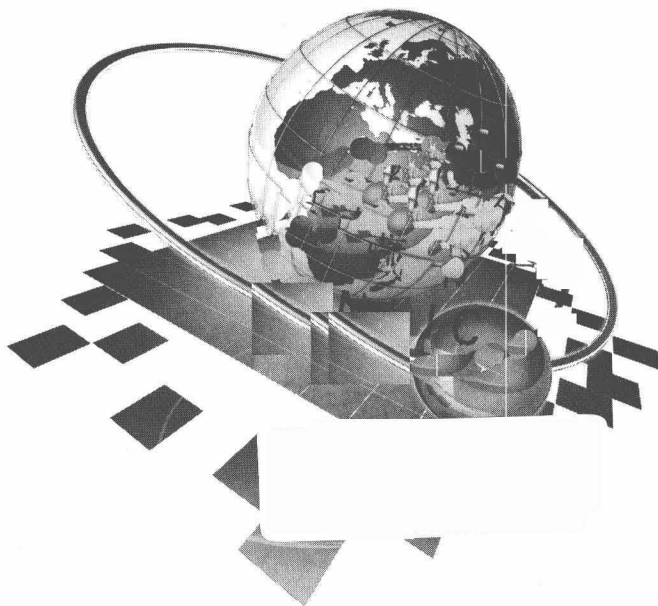
机械工业出版社
China Machine Press

Windows Sockets Network Programming

Windows Sockets 网络编程

(美) Bob Quinn Dave Shute 著

徐磊 腾婧 张莹 何慧 译



机械工业出版社
China Machine Press

本书是 Windows Sockets 网络编程领域公认的经典著作，由 Windows Sockets 2.0 规范解释小组负责人亲自执笔，权威性毋庸置疑。它结合大量示例，对 Windows Sockets 规范进行了深刻地解读，系统讲解了 Windows Sockets 网络编程及其相关的概念、原理、主要命令、操作模式，以及开发技巧和可能的陷阱，从程序员的角度给出了大量的建议和最佳实践，是学习 Windows Sockets 网络编程不可多得的参考书。

全书分为三部分：第一部分（第 1~6 章），提供了翔实的背景知识和框架方面的概念，借助于此框架，读者可理解 WinSock 的具体细节，包括 Windows Sockets 概述、OSI 网络参考模型、TCP/IP 协议簇中的协议和可用的服务、WinSock 网络应用程序的框架及其工作机制、WinSock 的三种操作模式、socket 通信机制等；第二部分（第 7~12 章），以 FTP 客户端实例为基础介绍了函数实例库，还介绍了客户端程序、服务器程序和 DLL 中间构件及它们的相应函数，并涵盖 socket 命令和选项及移植 BSD Sockets 相关事项等；第三部分（第 13~17 章），介绍了应用程序调试技术和工具，针对应用编程中的陷阱的建议和措施，WinSock API 的多种操作系统平台，WinSock 规范的可选功能和 WinSock 规范 2.0 中的所有新功能。另外，本书还提供了 4 个极具参考价值的附录。

Authorized translation from the English language edition, entitled *Windows Sockets Network Programming*, 9780768682328 by Bob Quinn, Dave Shute, published by Pearson Education, Inc., Copyright © 1996 Addison-Wesley Publishing Company, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and CHINA MACHINE PRESS Copyright © 2012.

本书封底贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2011-5303

图书在版编目（CIP）数据

Windows Sockets 网络编程 / (美) 奎因 (Quinn, B.), (美) 舒特 (Shute, D.) 著; 徐磊等译. —北京: 机械工业出版社, 2012. 8

(华章专业开发者丛书)

书名原文: Windows Sockets Network Programming

ISBN 978-7-111-39274-3

I. W… II. ①奎… ②舒… ③徐… III. Windows 操作系统—应用软件—网络软件—程序设计 IV. TP316.86

中国版本图书馆 CIP 数据核字 (2012) 第 167987 号

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: 高婧雅

薰城市京瑞印刷有限公司印刷

2012 年 8 月第 1 版第 1 次印刷

186mm × 240mm · 27.5 印张

标准书号: ISBN 978-7-111-39274-3

定价: 89.00 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

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

译者序

从最早实现 TCP/IP 协议体系的 UNIX (BSD 4.3 发布于 1983 年), 到后起直追的 Windows 系列操作系统, TCP/IP 协议栈已经成为各种主流操作系统必要和默认的组成部分, 而 Sockets API 就是网络应用和 TCP/IP 协议栈之间的桥梁 (WinSock 2 也支持其他下层协议, 但 TCP/IP 体系一直独领风骚)。

就像作者借 Cyril Connolly 的名言所隐喻的那样, 纵观 UNIX 和 Windows 两大操作系统阵营的发展历史, 两者也曾有很多结合点, Windows Sockets API 就是其中之一, 它脱胎于 UNIX 的 Berkeley Sockets API, 继承了它的基本框架, 包括最基本的流式 socket 和数据报 socket 的工作流程, WinSock 都遵循着 Berkeley Sockets 原创的机制。Windows Sockets 虽然脱胎于 Berkeley Sockets, 但是它也针对 Windows 消息驱动机制进行了扩展。在现实世界中, Windows Sockets 已成为 Windows 网络编程事实上的标准, 涉及微软所有桌面操作系统和嵌入式系统 (Windows CE、Windows Mobile), 因而得到了广大开发人员的重视。

时至今日, 很多开发平台都提供了基于组件和框架级别的网络 API, 然而, 对于网络应用开发者来说, Windows Sockets API 始终是实现高效和灵活的网络应用编程的不二选择。

这本书的作者 Bob Quinn 和 Dave Shute 属于最早涉足 Windows Sockets 技术的人之一, 其中, Bob Quinn 为 Windows Sockets 1.1 版本做出了重要贡献, 他也是 WinSock 2 规范工作解释小组的负责人, 曾负责 FTP Software 公司的 WinSock DLL 开发工作, Dave Shute 在 Windows Sockets 问世之初也曾服务于 FTP 公司, 这就使得他们既能为读者详尽而系统地分析 Windows Sockets 规范及其实现, 又能从应用开发的角度给读者以建议和指导。除此之外, 本书还具有以下特色:

1) 突出工作原理。一般介绍应用编程接口规范及其实现的书很容易写成本使用手册, 而本书对 Windows Sockets 的概念、网络应用程序的工作机制、网络操作的 3 种工作模式以及 TCP 和 UDP 客户与服务器的 workflows 等都有很生动、详细的讲解。

2) 示例丰富。既有客户端的程序示例, 也有服务器端的程序示例, 还有完整的网络服务程序示例。此外, 在介绍协议首部的附录 A 中, 作者也分别给出了同一子网和不同子网中客户机和服务器之间会话的示例, 从 ARP 解析到连接管理, 均给出图示, 详尽到每个协议字段的值, 体现作者严谨负责的写作态度。

3) 内容全面、系统、实用。全书共 17 章, 内容覆盖了从概念、原理、操作模式到实用编程的各个方面, 比较了 Windows Sockets 和 Berkeley Sockets 的差异和移植问题, 专门介绍 WinSock 可选特性和 WinSock 2 的扩展特性, 并从应用开发的角度对程序的调试和注意事项给出了详细的分析和建议。

该书出版时, WinSock 2 已经推出, 因此, 书中对于 Windows Sockets 1.1 和 WinSock 2 版本均展开了介绍。实际上, WinSock 2 是 Windows Sockets 1.1 的超集 (superset)。虽然现今所有的 Windows 系统 (包括 Windows CE 和 Windows Mobile) 都支持 WinSock 2, 但是仍然同时提供 Windows

Sockets 1.1 版本的 DLL，以帮助那些不需要 WinSock 2 扩展特性的用户将程序限定在基本集上。

若说有什么不足的话，那就是由于出版年代的原因，书中对于 16 位 Windows 平台下 WinSock 编程局限的讨论，在今天看来意义已经不大，32 位平台的用户可以忽略这一部分内容。瑕不掩瑜，全书对 Windows Sockets 全面而又详尽的阐释，仍然是指导 WinSock 应用开发的“真经”。

本书的翻译由 4 位译者合作完成，其中徐磊承担了第 1~6 章以及附录 A 的翻译工作，滕婧承担第 13~17 章以及附录 B 的翻译工作，张莹承担第 8~11 章以及附录 D 的翻译工作，何慧承担第 7 章、第 12 章和附录 C 的翻译工作。徐磊对全书的译文进行了必要的修改、润色和统稿。

感谢身边朋友们的不吝赐教，感谢家人对我的默默支持。由于水平和时间所限，译者对原书中一些知识的传达可能不够到位，若有翻译错误和不妥之处，敬请读者指正。

徐磊

2012 年 5 月

序

在 1991 年，人们已经能够明确地看出 Microsoft Windows 将在未来桌面计算方面发挥决定性的作用。同时，如果你密切关注的话，会听到 TCP/IP 渐行渐近的脚步声，它将给人们的生活带来颠覆式的影响。然而，在 Microsoft Windows 和 TCP/IP 的结合处，存在着一个问题，所有 PC 的 TCP/IP 产品供应商都为各自的产品推出了不同的编程接口。我作为世界上众多程序员中的一员，深夜坐在 PC 前，告诉自己必须有一种更好的解决办法。

这个“更好的方法”于 1991 年 10 月的 Interop 上的 BOF (“Bird Of Feather”) 座谈会上得以启动，在一个聚集了大约三十几个业界杰出人物的房间里，初步形成了这个构想：建立一个单一的、标准的传输接口，关键的衡量标准就是使其尽可能地接近现有的 Berkeley Sockets API，使得 Windows 基于消息的应用程序使用该接口时无须借助其他工具。对于技术人员来说，总是存在这样的诱惑，即追求完美和在技术设计中做到面面俱到。我们作为一个团队，所做出的重要决定之一就是：把这个接口的功能限制在能满足大多数程序员的需要上。经过互为竞争伙伴的厂商们的通力合作，最终为 Windows Sockets API 的成功创建奠定了基础，这就是人们所说的“WinSock”。

在其后的 15 个月里，WinSock 邮件列表 (mailing list) 中的很多人都为最初的规范工作做出了贡献，期间召开了若干次互操作性方面的座谈会，这些协同努力的结果就是 Windows Sockets API 规范的 1.1 版本。

目前，人们对于在 Microsoft Windows PC 上开发和使用通信应用程序的热情井喷式增长，WinSock 的推出可谓适逢其时。与此同时，致力于定义该编程标准的开发者群体也日益壮大，有更多的程序员加入这个群体，他们在全球范围内协同合作，希望通过使用 WinSock 来给自己的应用程序软件增加通信功能。简而言之，WinSock 已经变得极为流行，从而急需一本优秀的有关 WinSock 的书，至少迄今为止仍缺乏这样一本书。

能为该书写序，我感到非常荣幸。这本书为开发人员提供了情境相关的框架和开发高质量 WinSock 软件所需要的详细信息，因此，它填补了一个空白。无论你是一个 WinSock 的初学者，还是一个自诩的 WinSock 专家，都能够从此书中有所收获。Bob 和 Dave 凭借他们在 WinSock 方面深厚、坚实和实际的经验积累，给读者以清晰的指导，以帮助读者开发健壮的、基于 WinSock 的软件。本着 WinSock 团队所体现的务实精神，Bob 和 Dave 在构思此书时能够从程序员的角度出发，为他们着想。本书前面几章为读者提供了翔实的背景知识和框架方面的概念。借助于此框架，读者可理解 WinSock 的具体细节，并了解怎样能使其更好地应用于自己的编程。在编写客户端程序、服务器程序或 DLL 中间构件方面，后续几章做了很好的安排，所提供的内容准确地把握了细节和难易程度。最后，由于 WinSock 依然在不断地发展和完善，知道从哪里获取更多的信息是很重要的，因此，附录部分包含了其他信息资源访问途径。

VI

自创建 WinSock 伊始，Bob Quinn 和 Dave Shute 就投入其中，他们对 WinSock 可谓了如指掌，无所不知，可能睡梦中也是 WinSock 吧！我赞赏他们的梦想和努力，毫无保留地向读者推荐此书。

Martin Hall
WinSock 工作组主席
Stardust Technologies, Inc. 首席技术官
1995 年 5 月

前 言

本书讲述 Windows Sockets 应用编程接口 (API), 即通常所说的“WinSock”, 其定位是作为 Windows Sockets 规范 1.1 版本的指南而非代替它。书中内容提供了该规范的路线图和学习资料, 自顶向下地阐述了 Windows Sockets 规范的每个部分, 讨论了可选功能以及 Windows Sockets 规范 2.0 版本的许多功能。

书中内容侧重为读者提供指南, 帮助他们了解如何编写平台可支持的和可扩展的网络应用程序, 使其有效地运行于所有 Windows Sockets 的实现上。有一种令人沮丧的说法是: “除了最基本的‘hello world’, 要想编写能够在所有 WinSock 实现上运行的任何程序都是不可能的。”这种说法肯定是不对的。如果一个应用能在一种 WinSock 上运行, 而不能在另一种 WinSock 上运行, 那么十有八九是由于开发人员做了某种不正确的假定。他们想当然地认为 WinSock 能够做某种事情, 但是并没有确切的规范依据。也就是说, 这种应用程序的失败既不是 WinSock 实现方面的过错, 也不是 WinSock 规范的过错, 它是可以避免的, 我们就来告诉你应该怎么做。

此书适用于所有希望成功掌握 WinSock 应用编程的人。如果你正在做这些事情: 从零开始编程, 打算从 BSD Sockets (或任何其他网络 API) 移植程序, 编写网络动态链接库 (DLL), 或者只是打算升级别人编写的应用程序, 那么这本书就是为你而写的。我们既论述 16 位 Windows 平台 (Windows 3.1、Windows for Workgroup), 也覆盖 32 位 Windows 平台 (Windows NT 3.1、Windows NT 3.5、Windows 95)。我们还介绍其他支持 WinSock API 的平台: 那些与 Windows 相关并打算增补 Sockets 的平台。

内容安排

书的前半部分 (第 1~6 章) 包含网络编程的介绍, 因此不要求读者具有一定的基础。后半部分通过详细的讲解和代码实例展示更深入的内容。附录部分可供读者快速查阅。

第 1 章概括地介绍了 Windows Sockets。第 2 章给出了网络软件架构概览。第 3 章介绍 TCP/IP 协议簇中的协议, 侧重于你的网络应用程序可以使用的服务, 以及这些服务的优缺点。第 4 章开始详细介绍 WinSock 编程接口, 从基本网络功能调用的角度解析所有网络应用程序的框架。第 5 章的内容覆盖不同的操作模式。第 6 章讨论每个网络应用程序所包含的状态机。

第 7 章给出了一个 FTP 客户端的源代码, 这是书中最大的应用程序。第 8~10 章则对前面一直没有讨论过的所有的 WinSock 函数调用一一地展开详细讲述。第 11 章论述有关创建运行在 WinSock DLL 上的动态链接库的一些具体问题。第 12 章讲述将现有 BSD Sockets 源代码移植到 16 位或 32 位 Windows 平台上的一些问题和应对策略。

从第 13 章开始, 进入本书的收尾部分, 并解决一些遗留问题。第 13 章详细介绍 Windows Sockets 应用程序调试技术和工具。第 14 章针对 WinSock 应用编程中的陷阱和容易犯的错误给出一

般建议和具体措施。第 15 章介绍了目前提供 WinSock API 的多种不同的操作系统平台。第 16 章覆盖 WinSock 规范中的可选功能（有些是计划中的，有些则不是），并向读者介绍这些功能的应用场合和用法。最后，在第 17 章详细介绍 WinSock 规范 2.0 版本中的所有新功能。

附录 A 给出因特网协议簇（TCP/IP）的协议首部的简短说明。附录 B 包含全部 WinSock API 的快速参考，包括函数、结构和宏（包含一些被遗忘的）。附录 C 提供了一份详细的 WinSock 错误参考信息。附录 D 包含应用程序编译和链接机制的一些细节。

读者对象

虽然不要求读者具有任何网络、协议或者使用 socket 以及其他网络 API 进行网络编程的预备知识，但若有一些基础更好。本书既适用于初学者，也适用于有经验的网络应用程序开发人员，书中内容还包含了大量的 Windows Sockets 规范 1.1 版本中所没有的背景资料和说明信息，因此，即使 WinSock 应用开发高手也能通过阅读这本书而从中获益。

读者应具有 C 语言编程和 Microsoft Windows API（WinAPI 或者 Win32）的知识。

应用程序示例

书中的应用程序示例采用 Microsoft C 1.51 版本（16 位）和 Microsoft C 2.0 版本（32 位）创建，与 Borland C 4.0 版本兼容，并与源代码一起提供支持上述平台的 Makefile 文件。应用程序都经过测试，测试平台涵盖了几乎所有的商用和共享版本的 WinSock，网络测试环境为以太网和 PPP（点到点协议）网络，使用本地和远程连接。

致谢

这本书能够与读者见面得益于很多人的帮助和参与，在此向他们表示万分的感谢。特别要感谢读者，你们就是这本书存在的理由，祝愿你们成就不平凡的 Windows Sockets 应用。

最要感谢的是 Bob Quinn，是他的愿望孕育了这本书，而我只是为这本书的问世提供了一些必要的辅助工作。

Dave Shute
dks@world.std.com

Reading, MA
1995 年 5 月

有太多的人为本书的出版提供了帮助，任何一个致谢列表都不足以表达我的谢意，然而，我还是要特别地提到一些人。首先是我的家人，感谢他们的默默奉献，能与他们再次共度时光将是多么美好；其次是我的合著者兼朋友 Dave Shute，他的那支红笔和他的智慧一样敏锐和宝贵。

我十分感激 FTP Software 公司的 Larry Backman、Mike Khalandovsky 以及 John Keller 的支持和鼓励，特别要提到的是 FTP Software 公司的 crack SDK 支持人员 Dave Barnard、Kerry Hannigan 和 Helen Sylvester，正是他们持续不断地质疑，教会了我在 WinSock 应用编程中应该怎样做，不应该怎样

做，使我获益匪浅。

我在 WinSock 2.0 规范解释工作小组的共同管理人 Paul Brooks 和 Vikas Garg 功不可没，他们为把光明洒向 WinSock 的黑暗角落付出了不倦的努力，他们不仅阐明了规范中的很多东西，也使书中的一些内容更加明晰。

其他要提到的评阅人还有：Jim DeMarco、Fred Whiteside、Dave Andersen、Charlie Tai、Alun Jones、Eli Patashnik 以及我们的顾问编辑 Alan Feuer。

感谢 Addison-Wesley 出版社友善的朋友们：John Wait、Mike Hendrickson、Kim Dawley、Marty Rabinowitz、Simone Payment 和 Katie Duffy。

最后，我要向开创了 Windows Sockets 先河的 Martin Hall 和帮助他完成这项工作的合著者及其他奉献者致谢。一个好的想法的显著标志是它能促使你思考，“为什么我就没有想到!?”，这就是 WinSock，一个了不起的想法。这个不争的事实已被随之而来的成功所证明。

Bob Quinn
rcq@ftp.com
韦斯顿于马萨诸塞州

目 录

译者序
序
前言

第 1 章 Windows Sockets 概述	1	2.6 协议和 API	21
1.1 什么是 Windows Sockets	2	第 3 章 TCP/IP 协议服务	23
1.2 Windows Sockets 的发展历史	3	3.1 什么是 TCP/IP	23
1.3 Windows Sockets 的优势	4	3.2 TCP/IP 的发展历史	25
1.3.1 Windows Sockets 是一个开放的 标准	4	3.3 传输服务	26
1.3.2 Windows Sockets 提供源代码可 移植性	4	3.3.1 无连接的服务: UDP	26
1.3.3 Windows Sockets 支持动态链接 ...	5	3.3.2 面向连接的服务: TCP	28
1.3.4 Windows Sockets 的优点	6	3.3.3 传输协议的选择: UDP 与 TCP 的对比	29
1.4 Windows Sockets 的前景	7	3.4 网络服务	30
1.5 结论	7	3.4.1 IP 服务	30
第 2 章 Windows Sockets 的概念	9	3.4.2 ICMP 服务	31
2.1 OSI 网络模型	9	3.5 支持协议和服务	32
2.2 WinSock 网络模型	11	3.5.1 域名服务	32
2.2.1 信息与数据	12	3.5.2 地址解析协议	33
2.2.2 应用协议	12	3.5.3 其他支持协议	33
2.3 WinSock 中的 OSI 层次	12	3.6 TCP/IP 的发展前景	34
2.3.1 应用层	13	第 4 章 网络应用程序工作机制	35
2.3.2 表示层	14	4.1 客户端 - 服务器模型	35
2.3.3 会话层	14	4.2 网络程序概览	37
2.3.4 传输层	15	4.3 socket 的打开	37
2.3.5 网络层	16	4.4 socket 的命名	39
2.3.6 数据链路层	17	4.4.1 sockaddr 结构	40
2.3.7 物理层	17	4.4.2 sockaddr_in 结构	40
2.4 模块化的层次框	18	4.4.3 端口号	41
2.5 服务和协议	19	4.4.4 本地 IP 地址	41
		4.4.5 什么是 socket 名称	42
		4.4.6 客户端 socket 名称是可选的	42
		4.5 与另一个 socket 建立关联	43
		4.5.1 服务器如何准备建立关联	44
		4.5.2 客户端如何发起一个关联	45

4.5.3 服务器如何完成一个关联	47	5.6 平台比较	86
4.6 socket 之间的发送与接收	48	5.7 真正的阻塞	86
4.6.1 在“已连接的”socket 上 发送数据	48	5.7.1 线程化的应用程序	86
4.6.2 在“无连接的”socket 上 发送数据	49	5.7.2 其他适用性说明	87
4.6.3 接收数据	50	第 6 章 socket 状态	88
4.6.4 socket 解复用器中的关联	51	6.1 什么是 socket 状态	88
4.7 socket 的关闭	52	6.1.1 数据报 socket 的状态	89
4.7.1 closesocket()	52	6.1.2 流 socket 的状态	90
4.7.2 shutdown()	53	6.2 socket 状态的检测方法	91
4.8 客户端和服务端概览	54	6.2.1 根据函数调用的成功或失败 检测	92
第 5 章 操作模式	56	6.2.2 同步检测	93
5.1 什么是操作模式	57	6.2.3 异步检测	93
5.1.1 不挂机, 等待: 阻塞	57	6.3 WSAAsyncSelect()	93
5.1.2 挂机后再拨: 非阻塞	57	6.3.1 WSAAsyncSelect() 的工作原理	94
5.1.3 请求对方回拨: 异步	58	6.3.2 什么是 WSAAsyncSelect() 事件	96
5.2 阻塞模式	58	6.3.3 避免机会丢失的窗口	96
5.2.1 阻塞 socket	58	6.3.4 什么是重使能函数	97
5.2.2 阻塞函数	59	6.3.5 什么是通知消息	98
5.2.3 伪阻塞的问题	59	6.3.6 为函数失败做好准备	98
5.2.4 阻塞钩子函数	60	6.3.7 在响应时设置循环	99
5.2.5 阻塞情境	61	6.3.8 撤销异步通知	99
5.2.6 撤销阻塞操作	62	6.3.9 事件描述	99
5.2.7 阻塞操作中的超时	64	6.3.10 AS_Echo 应用程序	102
5.2.8 无最少接收限制值	65	6.4 select()	115
5.2.9 代码示例	65	6.4.1 怎样使用 select()	116
5.3 非阻塞模式	67	6.4.2 select() 的宏	117
5.3.1 怎样使 socket 成为非阻塞的	68	6.4.3 select() 的输入参数	118
5.3.2 成功与失败不是绝对的	68	6.4.4 不要把 select() 作为定时器	120
5.3.3 探询而非阻塞	69	6.4.5 注意 0 超时	120
5.3.4 显式地避让	69	6.4.6 代码示例	120
5.3.5 代码示例	70	6.5 查看数据	122
5.4 异步模式	72	6.5.1 ioctlsocket() FIONREAD 命令	122
5.4.1 认识异步函数	73	6.5.2 带有 MSG_PEEK 标志的 recv()	122
5.4.2 撤销异步操作	74	6.6 带外数据	123
5.4.3 代码示例	74	6.6.1 TCP 的局限	124
5.4.4 AU_Time 应用	76	6.6.2 ioctlsocket() 函数和 SIOCATMARK 命令	125
5.5 性能比较	85		

6.6.3 代码示例	125	8.5 协议族与地址族	204
6.7 监听状态	126	第9章 socket 信息与控制	205
第7章 应用程序和函数实例库	127	9.1 socket 控制	205
7.1 FTP 客户端实例	127	9.1.1 FIONBIO	206
7.1.1 文件传输协议概述	128	9.1.2 FIONREAD	207
7.1.2 FTP 有限状态机	130	9.1.3 SIOCATMARK	208
7.1.3 AC_FTP 应用程序	132	9.1.4 其他控制命令	208
7.1.4 操作中的 AC_FTP	168	9.2 socket 选项	209
7.2 函数实例库	169	9.2.1 getsockopt() 与 setsockopt()	209
7.2.1 CenterWnd()	170	9.2.2 SO_ACCEPTCONN	211
7.2.2 CloseConn()	171	9.2.3 SO_BROADCAST	211
7.2.3 CreateLclFile()	172	9.2.4 SO_DEBUG	212
7.2.4 Dlg_About()	173	9.2.5 SO_DONTLINGER	213
7.2.5 GetAddr()	174	9.2.6 SO_DONTROUTE	213
7.2.6 GetBuf()	175	9.2.7 SO_ERROR	214
7.2.7 GetHostID()	176	9.2.8 SO_KEEPAIVE	214
7.2.8 GetLclDir()	177	9.2.9 SO_LINGER	216
7.2.9 GetPort()	179	9.2.10 SO_OOBLINE	218
7.2.10 WSAErrStr()	179	9.2.11 SO_RCVBUF 与 SO_SNDBUF	218
7.2.11 WSAperror()	180	9.2.12 SO_REUSEADDR	219
7.3 WinSockx.h	182	9.2.13 SO_TYPE	219
7.3.1 Globals.c	183	9.2.14 TCP_NODELAY	220
7.3.2 WSA_Xtra.h	184	9.2.15 不支持的选项	221
第8章 主机名称与主机地址	187	9.2.16 未来会使用的选项	222
8.1 主机名称与主机地址解析	187	9.3 阻塞钩子	222
8.1.1 hostent 结构	188	9.3.1 WSASetBlockingHook()	222
8.1.2 主机名称解析	189	9.3.2 WSAUnhookBlockingHook()	223
8.1.3 地址解析	195	9.3.3 阻塞钩子函数	223
8.2 主机表、域名系统和网络信息服务	198	9.3.4 阻塞状态	224
8.2.1 主机表	198	9.4 socket 名称	224
8.2.2 域名系统	198	9.4.1 getsockname()	225
8.2.3 网络信息服务	199	9.4.2 getpeername()	226
8.3 本地主机信息	200	第10章 支持例程	228
8.4 网络地址与格式化	201	10.1 启动与清除	228
8.4.1 in_addr 结构	201	10.1.1 WSAStartup()	228
8.4.2 inet_addr()	202	10.1.2 WSACleanup()	234
8.4.3 inet_ntoa()	203	10.2 字节顺序	234

10.3 服务名称与端口号	236	12.1.8 数据库文件操作不可用	276
10.3.1 servent 结构	236	12.1.9 bcmp()、bcopy()和 bzero() 不可用	276
10.3.2 服务解析	237	12.1.10 本地 IPC 不可用	277
10.3.3 端口解析	240	12.1.11 OOB API 不同	277
10.3.4 服务数据库	241	12.1.12 setsockopt()SO_DEBUG 是 可选的	277
10.4 协议名称与协议号	242	12.2 16 位 Windows 的局限性	277
10.4.1 protoent 结构	243	12.2.1 16 位 Windows 不清理现场	277
10.4.2 协议名称解析	243	12.2.2 16 位 Windows 是非抢占式的	278
10.4.3 协议号解析	247	12.2.3 16 位 Windows 文件系统 不安全	278
10.4.4 协议数据库	247	12.2.4 16 位 Windows 使用分段 寻址	279
10.5 报错	248	12.2.5 fork()不可用	279
10.5.1 WSAGetLastError()	248	12.3 移植 Berkeley Sockets 的其他事项	279
10.5.2 WSASetLastError()	249	12.3.1 包含文件	279
10.5.3 错误文本显示	249	12.3.2 socket 域	279
第 11 章 WinSock 上的 DLL	250	12.3.3 socket 类型	279
11.1 创建一个新的 API	250	12.3.4 socket 句柄	280
11.1.1 仿真的 API	251	12.3.5 错误值	280
11.1.2 应用协议 API	252	12.4 函数列表	280
11.1.3 封装的 WinSock API	252	第 13 章 调试	284
11.2 DLL 的问题	253	13.1 问题类型	284
11.2.1 重入	254	13.1.1 安装问题	285
11.2.2 任务管理	255	13.1.2 网络问题	285
11.2.3 与 WinSock DLL 的链接	256	13.1.3 程序问题	285
11.3 DLL 实例	256	13.1.4 WinSock DLL 问题	285
11.3.1 WSASimpl 应用程序	257	13.2 故障表现及其定位	286
11.3.2 WSASimpl DLL	258	13.3 问题评估	286
第 12 章 移植 BSD Sockets	272	13.4 安装调试	287
12.1 Windows Sockets 与 Berkeley Sockets 的差异	272	13.5 网络调试	287
12.1.1 socket 不是文件句柄	273	13.5.1 通用网络调试	288
12.1.2 signal()不可用	274	13.5.2 主机解析调试	289
12.1.3 域名不可用	275	13.6 应用程序调试	289
12.1.4 地址信息和操作不可用	275	13.7 调试工具	290
12.1.5 syslog()不可用	276		
12.1.6 变量 errno 不可用	276		
12.1.7 perror()和 strerror()不可用	276		

13.7.1	网络调试工具	290	14.6.5	关闭 socket	315
13.7.2	应用程序调试工具	295	14.6.6	使用主机名称和地址	316
13.7.3	WinSock 符合性和性能测试 工具	297	14.6.7	使用 WSAsyncSelect() 函数	316
13.7.4	其他调试工具	297	14.6.8	使用阻塞 socket	317
第 14 章	该做和不该做的	299	14.6.9	退出某个应用程序	319
14.1	特征化应用程序	299	14.6.10	其他注意事项	320
14.2	应用程序数据流	300	第 15 章	系统平台	322
14.2.1	增大缓存	301	15.1	WOSA 网络模型	322
14.2.2	MTU 的考虑	301	15.2	32 位 WinSock	323
14.2.3	健壮性原则	302	15.2.1	Windows 95	324
14.2.4	避免查看	303	15.2.2	交叉执行	324
14.2.5	先接收后发送	303	15.2.3	32 位 WinSock 特征	324
14.3	流算法	303	15.2.4	32 位操作系统优势	328
14.3.1	发送结构化数据	303	15.3	其他系统平台	329
14.3.2	发送交互数据	305	第 16 章	可选特性	330
14.3.3	检测无效连接	306	16.1	自相矛盾的可选标准	330
14.3.4	验证数据传输	306	16.2	应该使用可选特性吗	331
14.4	数据报算法	307	16.3	原始 socket	332
14.4.1	实现超时机制	308	16.3.1	ICMP ping 应用	332
14.4.2	数据报排序	308	16.3.2	IP_TTL 路径跟踪	337
14.4.3	有节制地发送数据	308	16.4	多播	338
14.4.4	同步发送方和接收方	309	16.4.1	为何使用多播	338
14.5	好消息准则	309	16.4.2	多播 API	339
14.5.1	检查返回值	309	16.4.3	多播机制	341
14.5.2	检测部分成功	310	16.5	环回接口	345
14.5.3	准备应对任何阶段出现的 任何错误	310	16.6	共享 socket	345
14.5.4	检测非致命错误	310	16.7	可选项	346
14.5.5	恰当地处理错误	311	16.8	socket 作为文件句柄	346
14.5.6	对用户透明	311	16.9	在任何时候都准备好应对任何 错误	347
14.5.7	留好出口	311	16.10	其他可选特性	347
14.5.8	不忽视任何警告信息	311	第 17 章	WinSock 2	348
14.6	常见陷阱和缺陷	312	17.1	你需要 WinSock 2 吗	349
14.6.1	打开和命名 socket	312	17.2	特性概述	349
14.6.2	建立连接	313	17.3	多协议支持	351
14.6.3	监听和同意建立连接	313	17.3.1	WinSock 2 架构	351
14.6.4	发送和接收	314			

17.3.2 协议独立	352	A.6 TCP 首部	371
17.3.3 名称空间独立	353	A.7 UDP 首部	372
17.4 重叠 I/O	356	A.8 对话示例	373
17.5 分散和聚合	358	附录 B 快速参考	380
17.6 服务质量	358	B.1 结构	380
17.7 socket 组	360	B.2 函数	384
17.8 多点和多播	361	B.3 宏	391
17.9 有条件的接受	362	附录 C 错误信息参考	398
17.10 连接和断开数据	363	C.1 获取错误值的位置	398
17.11 socket 共享	364	C.2 可能出现的错误	399
17.12 协议相关的附加功能	364	C.3 用户可以解决的错误	400
17.12.1 WSAIoctl()	365	C.4 详细的错误描述	400
17.12.2 get/setsockopt()	365	C.5 按数值排序的错误代码表	419
附录 A TCP/IP 协议首部	366	附录 D 用户必备	421
A.1 TCP/IP 报文中的分层	366	D.1 重要文件	421
A.2 ARP 首部	367	D.2 编译与链接机制	422
A.3 IP 首部	368	D.3 各种 WinSock 的使用	423
A.4 ICMP 首部	369	D.4 各种编程语言的使用	424
A.5 IGMP 首部	370		

Windows Sockets 概述

真理就像一条河，总是被分成支流，然后又汇合到一起。在两条支流之间，岛上的人们终生争论不休：哪一支才是河的主流？

——Cyril Connolly

本章内容

- 什么是 Windows Sockets
- Windows Sockets 的发展历史
- Windows Sockets 的优势
- Windows Sockets 的前景
- 结论

如果我们向网络程序员询问有关 Windows Sockets 的问题，很可能你会得到两种截然不同的回答：有的程序员们会很恼火地列举他们碰到过的问题，有的则会热情洋溢地点评他们的业绩和收获。这取决于你问的是什么问题。如果问：“你喜欢 Windows Sockets 吗？”，你会得到前一种回答；如果问：“你打算放弃 Windows Sockets 吗？”，那么你将会得到后一种回答。

如果询问 Windows Sockets 问世前的状况，你肯定会被告知那时的网络编程简直就是一场噩梦。当谈起在 Windows 平台[⊖]上开发，并能在多个 TCP/IP 实现方案上运行的网络程序，并对它们进行打包和提供支持，程序员们普遍面露恐惧之色，这让他们回忆起那个“黑暗时期”——每个 TCP/IP 厂商都有各自的网络应用编程接口（API）。

Windows Sockets 规范的 1.1 版本（更多地被称为“WinSock”）极大地改进了网络编程的状况。它建立了一套能让网络软件厂商达成一致的网络 API。当然，它并不完美，不同的解释导致 WinSock 的实现之间互不兼容，规范中的一些可选功能并没有获得一致的支持。但是这些灰色地带属于 WinSock 规范的次要部分。

从整体上看，这是一个完整的、有着明确定义的规范，WinSock 的实现也是一致的。已经有很多能够成功地跨所有 WinSock 实现版本而运行的应用案例，包括各种类型的应用：例如，采用流和数据报协议，在客户端和服务端之间发送和接收批量数据或交互类数据的应用。这些应用能够运行在各种平台：16 位 Windows（Windows 3.x）、32 位 Windows（Windows NT 和 Windows 95）以及其他平台（OS/2 和 SUN 工作站）。它们成功的秘密很简单：远离灰色地带。

在本书中，我们将介绍如何编写能够在任何 WinSock 的实现上运行的应用程序。我们将强调

[⊖] Windows 是微软公司的注册商标。