

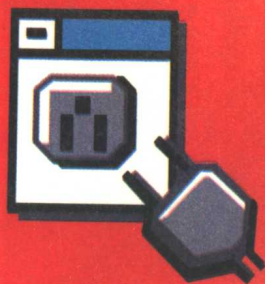
Windows Sockets

网络程序设计指南

蒋东兴 林鄂华 编著

石 晔

审校



Windows Sockets



清华大学出版社

Windows Sockets 网络程序设计指南

蒋东兴 林鄂华 编著
石 碧 审校

清华大学出版社

(京)新登字 158 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Windows Sockets 网络程序设计指南/蒋东兴,林鄂华编著. —
北京:清华大学出版社,1995

ISBN 7-302-01940-1

I. W... I. ①蒋... ②林... III. 窗口(软件)-计算机网络
程序-设计-指南 IV. TP311-62

中国版本图书馆 CIP 数据核字(95)第 12853 号

出 版 者: 清华大学出版社(北京清华大学校内,邮编 100084)

责任编辑: 刘明华

印 刷 者: 北京昌平环球印刷厂印刷

发 行 者: 新华书店总店北京科技发行所

开 本: 850×1168 1/32 印张: 7 1/8 字数: 183 千字

版 次: 1995 年 12 月第 1 版 1995 年 12 月第 1 次印刷

印 数: 0001—4000

书 号: ISBN 7-302-01940-1/TP·889

定 价: 8.80 元

内 容 简 介

Windows Sockets 是 Berkeley Sockets 在 Windows 环境下的扩充,它为 Windows 下网络异步通信提供了一种方便的开发和运行环境。本书主要介绍如何进行 Windows Sockets 网络程序设计,其中包括 TCP/IP 网络协议的基本情况,Berkeley Sockets 网络编程原理,Windows Sockets 基于消息的异步选择机制以及对 Berkeley Sockets 的扩充,并以范例的形式用一章的篇幅介绍了一个使用 Windows Sockets 进行 Windows 下网络异步通信的实现,读者通过修改扩充此范例即可获得符合自己要求的程序。另外,本书还提供了 Windows Sockets 库函数的使用说明,并以附录的形式给出了 Windows Sockets 版本 1.1 的头文件 winsock.h。

本书的读者是熟悉 C 语言和 Windows 的程序员,这本书为他们提供在 Windows 下实现网络程序设计方面的帮助。另外,本书也可作为大学本科生或研究生的参考资料,还可作为高等院校计算机网络课的教学参考书。

前 言

套接字(socket)最初是由加利福尼亚大学 Berkeley 分校为 UNIX 操作系统开发的网络通信接口,随着 UNIX 操作系统的广泛使用,套接字成为当前最流行的网络通信应用程序接口之一。Berkeley Sockets 只能用于 UNIX 操作系统,它不能支持微机 DOS 操作系统和 Microsoft Windows 环境。但是,socket 在 UNIX 中的成功应用使得将 socket 移植到 DOS 和 Windows 下成为一件很有意义的工作。90 年代初,由 Sun Microsystems,JSB Corporation,FTP software,Microdyne 和 Microsoft 等几家公司共同制定了一套标准,即 Windows Sockets 规范,他们试图使 Windows 下 Sockets 程序设计标准化。Windows Sockets 规范在 1993 年已推出了 1.1 版,现在许多公司都实现了 Windows Sockets,如 IBM 公司的 TCP/IP V2.1 for DOS 就提供了 Windows Sockets API。

Windows Sockets API 是 Microsoft Windows 的网络程序接口,它包括一个标准的 Berkeley Sockets 功能调用的集合,以及为 Windows 所作的重要扩充。扩充的功能调用都冠以 WSA(Windows Sockets Asynchronous)前缀,表明它们都允许异步的 I/O 操作,并且采用了符合 Windows 消息机制的网络事件异步选择机制。这些扩充有利于应用程序开发者更好地利用 Windows 的消息驱动特性,设计出高性能的网络程序。

本书是一本全面介绍 Windows Sockets 程序设计的书,介绍了 Berkeley Sockets 网络编程原理,Windows Sockets 网络程序设计,并提供了 Windows Sockets 规范 1.1 版定义的函数接口。全书共分 6 章。第 1 章介绍 TCP/IP 网络协议,Sockets 概念和本书使

用的一些专门术语。第2章介绍标准 Sockets 接口程序设计原理, 以及如何设计出 Sockets 典型的客户/服务器模式的应用程序。第3章介绍 Windows Sockets 基于消息的异步选择机制, 阻塞处理方法, 异步支持函数以及 Windows 下网络程序设计。第4章分类介绍了 Windows Sockets 库函数的作用。第5章用一个完整的例子说明 Windows 下网络异步通信程序的实现。第6章提供了 Windows Sockets 库函数。另外, 附录1提供了 Windows Sockets 的错误码。附录2提供了 Windows Sockets 的头文件 winsock.h, 以供设计应用程序时参考。

由于本书不但介绍了 Windows Sockets 网络程序设计, 还较详细地介绍了标准 socket 编程原理, 因此 UNIX 环境下的 socket 网络程序设计者也可以从中获得帮助。

本书第1章和第2章由林鄂华编写, 其余各章由蒋东兴编写, 全书由石焜审校。在本书的编写过程中, 得到了朱莉、张伟、郭茜、苏军等同志的许多帮助, 在此表示感谢。

由于 Windows Sockets 资料非常新颖, 尽管我们做了最大的努力, 本书仍难免存在一些不足之处, 希望各位同行专家批评指正。

作者

1995年3月

于清华园

目 录

目录.....	I
前言.....	V
第1章 TCP/IP 简介	1
1.1 TCP/IP 协议的起源和发展	1
1.2 TCP/IP 的体系结构和特点	4
1.3 术语.....	7
1.3.1 套接字	7
1.3.2 Windows Sockets 实现	8
1.3.3 阻塞处理例程	8
第2章 Sockets 编程原理	10
2.1 问题的引入	10
2.2 套接字编程基本概念	11
2.2.1 网间进程通信.....	11
2.2.2 服务方式.....	14
2.2.3 客户/服务器模式	16
2.2.4 套接字类型.....	17
2.3 基本套接字系统调用	18
2.3.1 创建套接字——socket()	18
2.3.2 指定本地地址——bind()	18
2.3.3 建立套接字连接——connect()与 accept().....	19
2.3.4 监听连接——listen()	21
2.3.5 数据传输——send()与 recv()	23
2.3.6 输入/输出多路复用——select()	24
2.3.7 关闭套接字——closesocket()	25
2.4 典型套接字调用过程举例	25

第3章 Windows Sockets 程序设计	33
3.1 Windows Sockets 简介	33
3.1.1 什么是 Windows Sockets	33
3.1.2 Windows Sockets 组成部分	33
3.1.3 Windows Sockets 对 Berkeley Sockets 的扩充	34
3.2 异步选择机制	35
3.3 阻塞处理方法	39
3.4 Windows Sockets 网络程序设计	41
3.4.1 启动与终止	41
3.4.2 异步请求服务	45
3.4.3 异步数据传输	47
3.4.4 出错处理	48
3.4.5 宏的使用	49
3.4.6 移植应用程序	51
3.5 较深入的问题	52
3.5.1 中间 DLL 设计	52
3.5.2 多线程环境下的 Windows Sockets	53
第4章 Windows Sockets 库函数综述	55
4.1 套接字函数	55
4.2 数据库函数	57
4.3 Windows Sockets 专用的增设函数	58
第5章 Windows 下网络异步通信实例	62
第6章 Windows Sockets 库函数	91
6.1 标准 Socket 函数	91
6.1.1 accept()	91
6.1.2 bind()	93
6.1.3 closesocket()	96
6.1.4 connect()	97
6.1.5 getpeername()	100
6.1.6 getsockname()	101
6.1.7 getsockopt()	103

6.1.8	htonl()	105
6.1.9	htons()	106
6.1.10	inet_addr()	106
6.1.11	inet_ntoa()	107
6.1.12	ioctlsocket()	108
6.1.13	listen()	110
6.1.14	ntohl()	112
6.1.15	ntohs()	112
6.1.16	recv()	113
6.1.17	recvfrom()	115
6.1.18	select()	118
6.1.19	send()	121
6.1.20	sendto()	123
6.1.21	setsockopt()	126
6.1.22	shutdown()	131
6.1.23	socket()	132
6.2	数据库函数	135
6.2.1	gethostbyaddr()	135
6.2.2	gethostbyname()	137
6.2.3	gethostname()	138
6.2.4	getprotobyname()	139
6.2.5	getprotobynumber()	140
6.2.6	getservbyname()	141
6.2.7	getservbyport()	143
6.3	Windows Sockets 专用的增设函数	144
6.3.1	WSAAsyncGetHostByAddr()	144
6.3.2	WSAAsyncGetHostByName()	147
6.3.3	WSAAsyncGetProtoByName()	149
6.3.4	WSAAsyncGetProtoByNumber()	150
6.3.5	WSAAsyncGetServByName()	152
6.3.6	WSAAsyncGetServByPort()	153

6.3.7	WSAAsyncSelect()	155
6.3.8	WSACancelAsyncRequest()	161
6.3.9	WSACancelBlockingCall()	163
6.3.10	WSACleanup()	164
6.3.11	WSAGetLastError()	166
6.3.12	WSAIsBlocking()	166
6.3.13	WSASetBlockingHook()	167
6.3.14	WSASetLastError()	168
6.3.15	WSAStartup()	168
6.3.16	WSAUnhookBlockingHook()	173
6.4	Windows Sockets 定义的宏	174
6.4.1	FD_CLR	174
6.4.2	FD_ISSET	174
6.4.3	FD_SET	174
6.4.4	FD_ZERO	175
6.4.5	WSAGETASYNCBUFLen	175
6.4.6	WSAGETASYNCERROR	175
6.4.7	WSAGETSELECTERROR	176
6.4.8	WSAGETSELECTEVENT	176
6.4.9	WSAMAKEASYNCREPLY	176
6.4.10	WSAMAKESELECTREPLY	177
附录 1	Windows Sockets 错误码	178
附录 2	Windows Sockets 头文件	182
参考文献		216

第 1 章 TCP/IP 简介

1.1 TCP/IP 协议的起源和发展

微电子技术、计算机技术、通信技术的迅猛发展,促进了计算机网络的实现和发展。从 1969 年第一个分组交换计算机网 ARPANET 的出现,随着计算机硬件技术的飞速发展,计算机硬件价格的急骤下降,至今涌现了许多大型计算机网络,如在美国 NFS 支持下设计的计算机科学研究网 CSNET,租用 9600 bps (波特)速率电话线作传输介质连接 400 多所大学的开放式计算机网络 BITNET,为全美医学科学家提供研究合作环境的斯坦福大学医学实验计算机网 SUMEX-AIM,英国的联合科学网 JANET,台湾学术网络 TANET 等。计算机网络的发展为提高信息工业的生产力,提供了一种全社会的、经济的、快速的存取手段。

这些网络建立的目标是:

1. 网络用户的资源(包括软件、硬件、信息等)共享;
2. 强化网络用户间的合作;
3. 加速学术研究成果的传播与流通;
4. 提供对网络技术的研究。

因此,如何实现不同网络及计算机间的互操作成为计算机联网的最关键问题。经过近 20 年的研究,到 80 年代末 90 年代初,有了肯定的答案:这就是采用 TCP/IP 协议。

TCP/IP 协议只是众多比较完善的网络协议中的一种。许多其它网络协议,如 Xerox 的 XNS,DEC 的 DECNET 和 IBM 的 SNA,虽然功能强大且拥有很多用户,但它们在异种机互联方面

功能很弱。国际标准化组织(ISO)为实现计算机网络互联制定了开放系统互联标准(OSI),但 OSI 目前还缺乏足够多的产品支持,而且 OSI 的许多标准仍在制定中。于是,在 80 年代初,人们选择了 TCP/IP 作为实现异种机互联的工业标准。这是一个在国际标准 ISO/OSI 尚未完全被采纳时,用户和厂家共同承认的标准。

在 TCP/IP 协议成为工业标准之前,TCP/IP 协议经历了近 12 年的实际测试。早在 70 年代中期,为了支持研究工作,美国国防部高级计划署(以下简称高研署)就开始着手全美范围内异种计算机间的连接。那时,计算机与计算机之间的连接使用的还只是点对点专用线路,计算机与计算机的通信规约采用的是各厂家自行定义的专门协议。针对当时的现状,高研署与许多机构共同讨论制订了开放的通信协议标准,以满足日益迫切的基于异种操作系统的异种网络之间的通信连接,即 TCP/IP 开放协议。

1980 年,高研署在它的网络上首先采用 TCP/IP 协议。随后,又命令它所赞助的 ARPANET 网上的计算机都必须遵循 TCP/IP 协议。1983 年 1 月,ARPANET 网向 TCP/IP 的转换全部结束。美国国防通信署(Defense Communication Agency, DCA)将 ARPANET 分为独立的两部分,一部分仍叫 ARPANET,用于进一步的研究工作;另一部分稍大一些,成为著名的 MILNET,用于军方的非机密通信。为推广 TCP/IP 协议,高研署又首先与许多公司达成协议,低价出售 TCP/IP 的实现,将 TCP/IP 模块安装在政府部门通用的计算机操作系统上;并通过资助 BBN(Bolt Beranek and Newman, Inc.)实现用于其 UNIX 操作系统的 TCP/IP 协议;还通过资助加州伯克利大学,将 TCP/IP 协议融入 UNIX BSD,促成 TCP/IP 与当时多数大学中流行的 UNIX BSD 的结合。对于各厂家开发的 TCP/IP 产品,由于有国防部通信署的检查和验证,保证产品符合标准,具有良好的互操作性,因而极大地推广了 TCP/IP 的应用。

1983年,伯克利推出内含TCP/IP的第一个UNIX BSD版本,满足了当时许多大学院系联网的要求,提供了一种联网的手段,以建立各自的局域网。使得未加入ARPANET网的用户也可以使用TCP/IP。随着安装UNIX BSD操作系统的SUN工作站的普及,TCP/IP的需求量迅速增加。UNIX BSD在网络方面的成功有以下几个原因:首先,除提供标准的TCP/IP应用程序外,它还支持一组网络服务工具程序(utilities)。这些工具的调用格式与UNIX命令调用格式相似,深受UNIX用户欢迎。第二,UNIX BSD提供一种供应用程序访问通信协议的操作系统调用:socket。socket是一种进程间通信机制,是UNIX输入/输出机制的推广。socket的出现使程序员可以很方便地访问TCP/IP,从事网络的研究开发工作。第三,UNIX BSD的开发者在其操作系统中还实现了地址分析协议(ARP)。地址分析协议是TCP/IP协议集中的一个协议,它能将Ethernet地址映射成Internet地址。Internet是ARPANET,NFSNET,MILNET等一组网络的集合,它用TCP/IP协议集来实现一个统一可互操作的网络,这样,便使得TCP/IP和Ethernet紧紧地联系在一起。现在,Ethernet网成为介质访问的工业标准并日渐流行,也使TCP/IP在当今局域网上得以流行,成为事实上的工业标准。

随着TCP/IP协议的普及和流行,今天,对TCP/IP的支持又出现了一些新的趋势,即许多外部设备安装有TCP/IP软件并通过网络直接与UNIX主机通信,达到资源共享的目的。这一趋势是用户对网络需求的增加和TCP/IP日益流行的结果。

人们一致认为,随着计算机网络和通信技术的不断发展,今后的计算机网络标准必将是ISO/OSI。但由于OSI推出的时间短、产品少,目前还不能取代TCP/IP。面对这种情况,TCP/IP将会有什么样的发展呢?

首先,应充实TCP/IP的上层协议,如网络管理、多媒体通信

等。与 TCP/IP 相比,OSI 的应用层协议功能更为丰富,但它还不能覆盖所有应用。因此,可开发具有 TCP/IP 自身特点的应用层协议。

其次,支持 TCP/IP 向外部设备的扩充,研制在外部设备上容易实现该协议的方法。

第三,与 OSI 结合,有效地利用 OSI 的服务,为 OSI 高层协议提供接口,从而保护现有投资,实现从 TCP/IP 向 OSI 的平滑过渡。

总之,当前采用 TCP/IP 实现异种机联网没有任何异议。今后,即使 OSI 取代 TCP/IP,由于 TCP/IP 庞大的装机量,会迫使厂家提供由 TCP/IP 向 OSI 过渡的策略。

1.2 TCP/IP 的体系结构和特点

TCP/IP 协议实际上就是在物理网上的一组完整的网络协议。用与 OSI 同样的层次模型来描述 TCP/IP 网络协议组,则 TCP 是提供传输层服务,而 IP 是提供网络层服务。此外,由于 TCP/IP 是一组协议的代名词,所以它还包含许多别的协议,其层次结构图见图 1.1。

Hardware Interface 相当于 OSI 的第 1—2 层,表示 TCP/IP 的实现基础,如 Ethernet,Token Ring,Token Bus 等;

IP,ARP,RARP,ICMP 相当于 OSI 第 3 层,其中:

IP 网间协议(Internet Protocol)。负责主机间数据的路由和网络上数据的存储。同时为 ICMP,TCP,UDP 提供分组发送服务。用户进程通常不需要涉及这一层。

ARP 地址分析协议(Address Resolution Protocol)。此协议将网络地址映射到硬件地址。

RARP 反向地址分析协议(Reverse Address Resolution

Protocol),此协议将硬件地址映射到网络地址。

ICMP 网间报文控制协议(Internet Control Message Protocol)。此协议处理信关和主机间的差错和传送控制。ICMP 报文使用 IP 数据报进行传送,这些报文通常由 TCP/IP 网络软件本身来保证正确性。

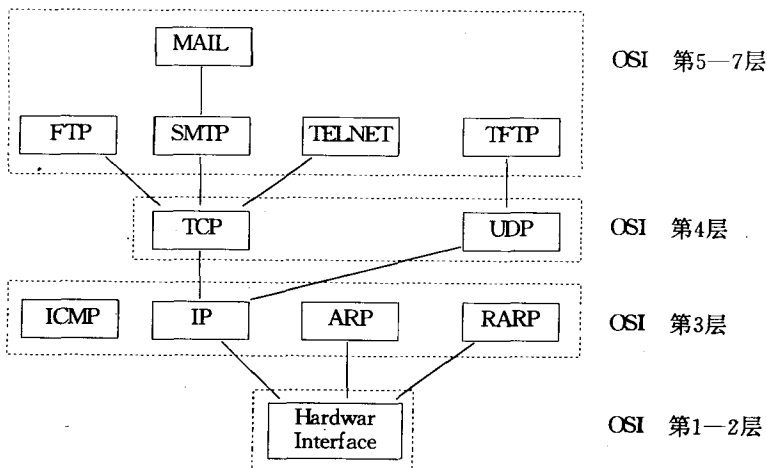


图 1.1 TCP/IP 协议组层次结构图

TCP,UDP 相当于 OSI 第 4 层,其中:

TCP 传送控制协议(Transmission Control Protocol),这是一种提供给用户进程的可靠的全双工字节流面向连接的协议。它要为用户进程提供虚电路服务,并为数据可靠传输建立检查。大多数网络用户程序使用 TCP。

UDP 用户数据报协议(User Datagram Protocol),这是提供给用户进程的无连接协议,用于传送数据而不执行正确性检查。

FTP,SMTP,TELNET,TFTP 相当于 OSI 第 5—7 层,其中:

FTP 文件传输协议(File Transfer Protocol),允许用户以文

件操作的方式(文件的增、删、改、查、传送等)与另一主机相互通信。

SMTP 简单邮件传送协议(Simple Mail Transfer Protocol), SMTP 协议为系统之间传送电子邮件。

TELNET 终端协议(Telnet Terminal Protocol)。允许用户以虚终端方式访问远程主机。

TFTP 简单文件传输协议(Trivial File Transfer Protocol), 是 FTP 的一种简化版本。

TCP/IP 协议的核心部分是传输层协议(TCP, UDP)、网络层协议(IP)和物理接口层,这三层通常在操作系统内核中实现。操作系统的内核是不能直接为一般用户所感受到的。一般用户感受到的只有应用程序(包括系统应用程序),即各种应用程序构成了操作系统的用户视图。那么应用程序通过什么样的界面与内核打交道呢?通过的是编程界面(即程序员界面)。各种应用程序,包括系统应用程序都是在此界面上开发的。编程界面有两种形式,一种是由内核直接提供的系统调用,另一种是以库函数方式提供的各种函数。前者在核内实现,后者在核外实现。因此,内核中实现 TCP/IP 协议的操作系统可以叫做 TCP/IP 操作系统,其核心协议 TCP, UDP, IP 等向外提供的只是原始的编程界面,而不是直接的用户服务。用户服务要靠核外的应用程序。TCP/IP 网络环境下的应用程序是通过网络系统编程界面套接字(socket)实现的。网间应用程序之间的作用方式为客户/服务器模式。TCP/IP 协议核心与应用程序的关系如图 1.2。

因此,网络的应用程序也不是能直接与 TCP/IP 核心打交道,而是与网络应用编程界面(套接字)打交道,编程界面构成了核心协议的用户视图。TCP/IP 核心协议连同网络物理介质一起,都是提供网络应用程序间相互通信的设施。我们把它叫做网际网设施。该设施比物理通信设施抽象,通用,功能更强。在多任务分时操作

系统中(如 UNIX),一个应用程序的一个实例对应于一个进程,因此网际网设施也就是网络系统中进程间通信的设施。

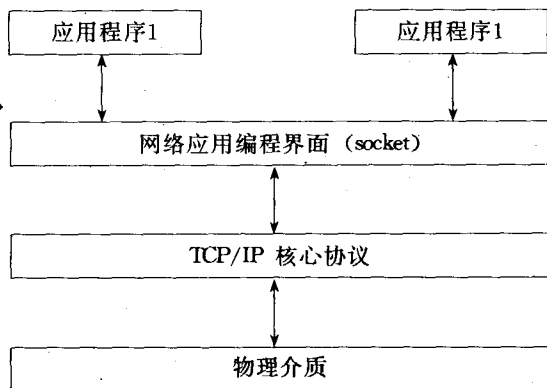


图 1.2 TCP/IP 协议核心与应用程序关系图

需要强调的是,TCP/IP 协议并没有确切地规定应用程序应该怎样与协议软件相互作用,也就是说没有对应用程序接口进行标准化,因此在原理上必须区分 TCP/IP 协议与接口。本书中讨论的 TCP/IP 编程界面(socket, Windows Sockets)源于 UNIX 操作系统,并已扩展到 DOS, Windows 操作系统。

1.3 术 语

1.3.1 套接字

套接字是从英文单词 socket 翻译而来,它是网络通信的基本构件。套接字是可以被命名和寻址的通信端点,使用中的每一个套接字都有其类型和一个与之相连的进程。

套接字存在于通信区域中。通信区域也叫地址族,它是一个抽