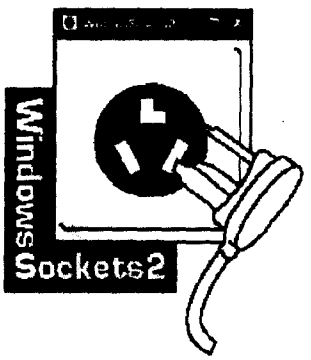
Windows Sockets
网络程序设计大全

蒋东兴 林鄂华 陈棋德
印 敏 刘启新 编著
石 磊 审校

清华大学出版社
http://www.tup.tsinghua.edu.cn

Windows Sockets
网络程序设计大全

蒋东兴 林鄂华 陈棋德 编著
印 敏 刘启新
石 磊 审校



清华大学出版社

(京)新登字 158 号

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。现在，Windows Sockets 已经成为 Windows 下标准的网络程序设计接口，Windows 下各种开发平台都支持 Windows Sockets 上的网络程序设计，提供了网络编程接口。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 1.1 程序设计原理、应用实例与库函数参考，Windows Sockets 2 的扩展特性、应用实例与库函数参考，MFC Sockets 程序设计原理与实例，Java Sockets 程序设计原理与实例，Delphi Sockets 程序设计原理与实例，PowerBuilder Sockets 程序设计原理与实例。

本书的读者是在Windows环境下使用 Sockets 进行网络程序设计的程序员，它为读者提供在Windows下使用各种平台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。

图书在版编目(CIP)数据

Windows Sockets 网络程序设计大全/蒋东兴等编著．—北京：清华大学出版社，1999
ISBN 7-302-03374-9

Ⅰ．W… Ⅱ．蒋… Ⅲ．计算机网络-程序设计 Ⅳ．TP393

中国版本图书馆 CIP 数据核字(1999)第 06550 号

出版者：清华大学出版社（北京清华大学校内，邮编 100084）
http://www.tup.tsinghua.edu.cn
印刷者：中国科学院印刷厂
发行者：新华书店总店北京发行所
开 本：787×1092 1/16 印张：24.75 字数：588 千字
版 次：1999 年 4 月 第 1 版 1999 年 4 月 第 1 次印刷
书 号：ISBN 7-302-03374-9/TP·1825
印 数：0001～5000
定 价：29.50 元

前 言

套接字(socket)最初是由加利福尼亚大学 Berkeley 学院为 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 等几家公司共同参与制定了一套标准,即 Windows Sockets 规范,他们试图使 Windows 下 Sockets 程序设计标准化。1993 年 1 月,他们制定了 Windows Sockets 1.1 版规范,定义了 16 位 Windows 下的网络标准编程接口。随着形式的发展,特别是 32 位 Windows 平台的发展,Windows Sockets 1.1 已不能满足需要。1994 年 5 月,WinSock 小组开始启动 Windows Sockets 版本 2 (WinSock 2)规范的制定工作。这一次,数以百计的公司、组织和个人参与了这一工作。1997 年 5 月,WinSock 2 的正式规范版本 2.2.1 发布。

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。

本书是在蒋东兴、林鄂华编著的《Windows Sockets 网络程序设计指南》(清华大学出版社,1995 年)的基础上改编扩充而成,是一本全面介绍 Windows Sockets 网络程序设计的书。本书除了保留原书介绍的 Berkeley Sockets 网络编程原理,Windows Sockets 1.1 网络程序设计,以及 Windows Sockets 1.1 版规范定义的函数接口外,还新增了 WinSock 2 的扩展特性、扩展库函数和应用实例,以及 MFC,Java,Delphi 和 PowerBuilder 等开发环境下的 Sockets 网络程序设计。另外,本书还扩充了原书的标准 Sockets 程序设计实例和Windows Sockets 1.1程序设计实例。

全书共分 12 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,并给出了一个通用的实例程序。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 章介绍 3 个实际的通信实例程序:点对点实时通信程序、广播通信程序和阻塞处理程

序。第5章提供了Windows Sockets 1.1库函数参考。第6章介绍WinSock 2体系结构和重叠I/O、服务质量等扩展特性。第7章提供了Windows Sockets 2扩展库函数。第8章用实例介绍了WinSock 2扩展库函数的使用及多址广播程序设计。第9章至第12章分别介绍了MFC,Java,Delphi和PowerBuilder等Windows应用开发环境下的Sockets网络程序设计。另外,附录A提供了Windows Sockets的错误码。附录B提供了Windows Sockets多点通信与多址广播的语义。

本书第1章(不含1.3节)、第2章(不含2.5节)由林鄂华编写,第9章由陈棋德编写,第10章由印敏编写,第11章由刘启新编写,其余各章由蒋东兴编写。全书由蒋东兴统稿,石晶审校。

在本书的编写中,得到了很多朋友和同行的帮助,作者表示深深的谢意。特别地,计算中心总工程师石晶老师在百忙中审阅了全稿;责任编辑刘明华老师长期以来的热情鼓励与支持;计算中心领导与同事的大力支持。另外,我还要感谢我的朋友陈棋德、印敏、刘启新,他们都是工作在第一线的教师与网络程序员,应作者之邀各自为读者献上了精彩的一章。最后,我还要将本书献给我的妻子与女儿!在本书创作的两年中,正好是女儿文菁孕育与出生的前后两年,妻子李霞林几乎承担了抚育女儿的全部工作,本书才能及时与读者见面。

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。

蒋东兴

1998年7月

于清华园

目 录

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，如由美国NFS支持设计的计算机科学研究网CSNET，租用9600 b/s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，以及台湾学术网络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机会；

(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讯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讯。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间联网的要求，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(套接字)。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解析协议(ARP)。地址解析协议是 TCP/IP 协议集中的一个协议，它能将 Ethernet(以太网)地址映射成 Internet(因特网)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，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。

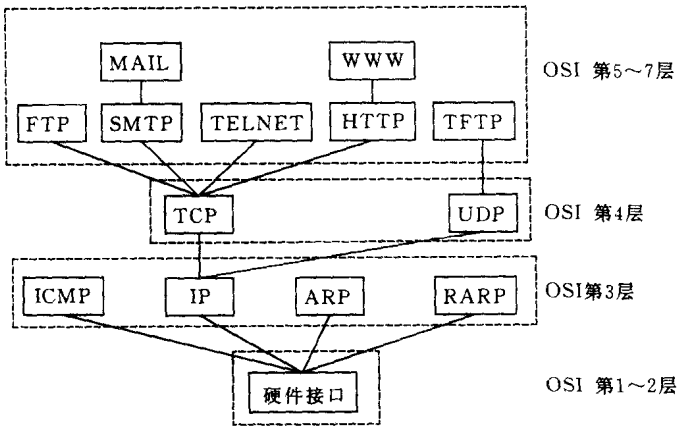


图 1.1 TCP/IP 协议组层次结构图

硬件接口(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 地址解析协议(address resolution protocol)。此协议将网络地址映射到硬件地址。
- RARP 反向地址解析协议(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。

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,HTTP 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。
- HTTP　超文本传输协议(hypertext transfer protocol)。是万维网 WWW 的基础,它使丰富多彩的 Internet 以简单的方式展现给用户。
- TFTP　简单文件传输协议(trivial file transfer protocol)。FTP 的一种简化版本。

TCP/IP 协议的核心部分是传输层协议(TCP 与 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(应用)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机/服务器模式。TCP/IP 协议核心与应用程序的关系如图 1.2 所示。

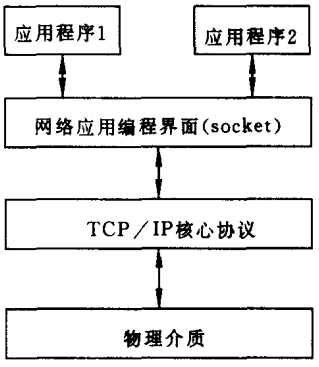


图 1.2　TCP/IP 协议核心与应用程序关系图

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。我们把它叫做因特网设施,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。因此因特网设施也就是网络系统中进程间通信的设施。

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，它是一个抽象概念，主要用于将通过套接字通信的进程的共有特性综合在一起。套接字通常只与同一区域中的套接字交换数据（也有可能跨越区域通信，但这只在执行了某种转换进程后才能实现）。Windows Sockets 只支持一个通信区域：网际域（AF-INET），这个域被使用网际协议簇通信的进程使用。

套接字都具有类型，它是根据用户可见的通信特征进行分类的。应用程序被假定为只在同一类型的套接字间通信，不过只要依据的通信协议支持，也完全可以在不同类型的套接字间通信。Windows Sockets 版本 1.1 支持两种套接字：流套接字（SOCK_STREAM）和数据报套接字（SOCK_DGRAM）。

1.3.2 Windows Sockets 实现

一个 Windows Sockets 实现是指实现了 Windows Sockets 规范所描述的全部功能的一套软件。一般来说，在 Windows 下实现 Windows Sockets 功能都是通过 DLL 实现的，并且很多实现是纯粹通过 DLL 实现的，因此，本书中的 Windows Sockets 实现的提法可以说等价于 Windows Sockets DLL（WINSOCK.DLL）。

Windows Sockets 版本 1.1 的实现必须支持 TCP 和 UDP 两种类型的套接字（即流套接字和数据报套接字），某些实现可能提供原始套接字（或 SOCK_RAM 类型），但这不是强制的。原始套接字是不鼓励使用的，为了兼容性，应用程序最好不要使用它们。

1.3.3 阻塞处理例程

阻塞处理例程（blocking hook，阻塞钩子）是 Windows Sockets 实现为了支持阻塞套接字函数调用而提供的一种机制。对单线程环境下的 Windows Sockets 来说，它有一个默认的阻塞处理例程。当应用程序引出一个阻塞的 Windows Sockets API 操作时，Windows Sockets 实现在让操作等待完成时调用此例程来模拟阻塞机制。默认的阻塞处理例程只是简单地获取并发送 Windows 消息，应用程序可以使用 WSASetBlockingHook()函数来安装自己定义的阻塞处理例程，它在函数阻塞时代替默认的阻塞处理例程执行。

1.3.4 多址广播

多址广播（multicast，也译作多点传送或组播）是一种一对多的传输方式，传输发起者通过一次传输就将信息传送到一组接收者，与单点传送（unicast）和广播（broadcast）相对应。

多址广播使用最广泛的是 IP multicast，它是标准 IP 网络层协议的扩展，由 Steve