



网络编程实例导学系列

Windows 网络编程之 VC 篇

萧秋水 文娟 编著



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



编程实例导学系列

Windows 网络编程之 VC 篇

萧秋水 文 娟 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Internet 技术无疑是当今计算机技术中的最大热点。本书以 Visual C++ 的最新版本作为开发工具, 尽最大可能包容网络现行的流行协议族, 讲解网络应用程序的开发原理及在 Windows 平台下的实现方法。

本书以编程实例为主线, 使读者针对某一项目可以立即参照实例中的步骤开始动手编程。虽然本书的基点在于实例, 但我们在介绍过程中又不局限于此, 而是更全面地介绍了该实例所需的各个技术要点, 以及尽量多的可选方法。本书附盘所带的所有代码都可以作为开发者二次开发的直接资源, 不必一切从头开始。

本书适用于使用 Visual C++ 6.0 进行 Windows 网络编程的中高级程序员。当然, 初级读者也可以从中受益。

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。

书 名: Windows 网络编程之 VC 篇

作 者: 萧秋水 文 娟

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编: 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 林庆嘉

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 16.75 字数: 408 千字

版 次: 2001 年 2 月第 1 版 2001 年 2 月第 1 次印刷

书 号: ISBN 7-900631-09-7

印 数: 0001 ~ 5000

定 价: 35.00 元

前 言

Windows 这个优秀的图形化操作系统,如今已深入人心,成了 PC 机市场中的绝对主流操作系统。Internet 的发展则是计算机技术的发展热点。如何在 Windows 平台下开发出出色的网络应用程序便成了一个很值得讨论的问题。有鉴于此,我们推出了《编程实例导学系列》丛书,本书便是其中的一本。

目前市面上讲解网络编程的书籍多是以内容为标准来分章论节,本书则是从一个崭新的角度向读者介绍网络编程的知识,这个角度对于读者来说也是最为实用的角度,即以编程实例为主线贯穿全书。如果读者正在进行一个网络程序项目的开发工作,本书是最适当不过的了。因为本书虽然基于实例,但我们在介绍过程中又不局限于此,而是全面地介绍了该实例所需的各个技术要点,以及尽量多的可选方法。更为有效的是,书中给出了实现项目的步骤,读者针对某一内容找到相应实例就可以立即开始动手编程了。读者可以利用本书原来的代码作为基础,不用一切都从头开始。

本书采用的开发工具是 Visual C++ 的最新版本。在内容上,力争覆盖 Windows 网络编程的大部分内容,适用于 Windows 95/98/NT/2000 平台。读者在使用本书时,既可以循序渐进地阅读,又可以直接查阅自己感兴趣的话题,下面对本书的内容做一个简单概述。

“实例一 用 WinSock 实现网上聊天”介绍目前已经成为时尚的聊天程序的编写方法,实例通过 WinSock 实现,它包括聊天服务器和客户端两个部分。

“实例二 获取主机名和 IP 地址”通过一个基于控制台界面的示例程序演示了如何获取本地计算机的主机名和 IP 地址的编程方法。

“实例三 枚举网上邻居”则比实例二更进一步,讲解了枚举局域网上所有计算机的方法。

“实例四 Ping 与 Trace Route”手把手地帮助你创建一个 PingPlus 实用工具,使用该工具,你既可以通过 Ping 检测远端计算机的可达性,又可以通过 Trace Route 列举出到达远端主机所经过的所有计算机。

“实例五 编程实现 Telnet”介绍了远程登录的编程原理,并通过一个 Telnet 服务器和一个 Telnet 客户机,演示了 Telnet 的具体实现步骤。

“实例六 邮件收发程序”给出了电子邮件收发程序的编程机理及具体的实现方法,其中给出的源代码很有参考价值。

“实例七 FTP 客户程序”为 Internet 的文件传送协议实现了一个客户端软件,使用这个工具,你可以从 Internet 的 FTP 服务器上自由地下载想要的文件。

“实例八 WWW 浏览器”演示的是一个简单版本的 Internet Explorer(即 Windows 自带的浏览器)的编写方法。当前,WWW 已经成为 Internet 发展最快的一个热点。

“实例九 RAS 拨号上网”为你讲解了 RAS(远程访问服务)客户机的编程原理,它是拨

号上网的各种编程方式中最为简单、有效的一种方法,是拨号编程必须掌握的一项技能。

“实例十 网络新闻组”实现了一个可以订阅网络新闻组的客户程序,你可以以此为起点,构建一个更为商业化的软件。

“实例十一 Finger 查询”使用 Finger 协议实现了一个客户端软件,用以获取特定服务器上的用户信息。

本书的一大特点是编程实例丰富,不仅在所附光盘中给出了作者在平时编码过程中积累的许多源代码,而且在书中给出了必要的、详细的讲解,引导读者逐步了解 Windows 网络编程的奥秘。

本书适用于使用 Visual C++ 6.0 进行 Windows 网络编程的中、高级程序员。当然,初级读者也可以从中受益。

本书主要由萧秋水编著,文娟完成了本书大部分内容的整理工作,蒋永波、唐明、刘政伟等也参与了本书的编写工作。限于作者水平,难免在内容选材和叙述上有不当之处。欢迎广大读者对本书提出批评和建议。

编 者

目 录

实例一 用 WinSock 实现网上聊天	1
📖 主要内容	1
?D 本例提要	1
?D 技术专题	2
※ WinSock 简介	2
※ WinSock API 主要函数的使用	3
※ MFC 对 WinSock API 的封装	5
📖 步骤之一——实现聊天服务器	6
?D 建立一个新项目	6
?D 添加新类 CListeningSocket	7
?D 添加代码实现固定端口的监听	8
?D 添加新类 CClientSocket	11
?D 添加代码实现与客户端的通信	13
?D 测试项目	17
📖 步骤之二——实现聊天客户端	17
?D 选择服务器	17
?D 添加新类 CChatSocket	20
?D 连接服务器	20
?D 与服务器通信	21
?D 与服务器联合测试	24
实例二 获取主机名和 IP 地址	25
📖 主要内容	25
?D 本例提要	25
?D 技术专题	25
※ 获取主机名和 IP 地址的机理	25
📖 步骤——获取本地计算机的主机名和 IP 地址	28

?D	建立一个新项目	28
?D	CIPEnum 类及其实现	28
?D	CMyIPEnum 类及其实现	31
?D	程序的主函数	32
实例三	枚举网上邻居	34
📖	主要内容	34
?D	本例提要	34
?D	技术专题	34
※	WNet API	34
※	WNet API 中的函数	36
📖	步骤——实现网上邻居枚举工具	41
?D	建立一个新项目	41
?D	添加 CNetwork 类对 WNet API 进行包装	41
?D	创建模板类 CNetSearch	48
?D	在对话框中完成枚举任务	50
实例四	Ping 与 Trace Route	52
📖	主要内容	52
?D	本例提要	52
?D	技术专题	53
※	Internet 控制报文协议 ICMP	53
※	ICMP 报文格式	54
※	ICMP 报文的分类	55
📖	步骤——实现一个实用工具 PingPlus	59
?D	建立一个新项目	59
?D	添加并实现 Clcmp 类	60
?D	添加并实现 RegKey 类	68
?D	添加并实现 SysTCPIP 类	71
?D	实现具体的用户操作过程	74
实例五	编程实现 Telnet	79
📖	主要内容	79

?D	本例提要	79
?D	技术专题	80
※	Telnet(远程登录)	80
※	NT 服务程序编程原理	82
☞	步骤之一——实现 Telnet 服务器	87
?D	建立一个新项目	87
?D	添加 CNTService 类包装 NT 服务器程序	87
?D	添加 SMRemoteService 类实现 Telnet 服务程序	95
?D	服务程序的启动细节	100
☞	步骤之二——实现 Telnet 客户机	101
?D	建立一个新项目	101
?D	实现服务器登录	101
?D	与服务器远程交互	102
实例六	邮件收发程序	106
☞	主要内容	106
?D	本例提要	106
?D	技术专题	107
※	邮件格式	108
※	SMTP 协议简介	108
※	SMTP 协议扩展	110
※	多用途网际邮件扩充	111
※	POP3 协议简介	112
☞	步骤之一——实现邮件发送程序	114
?D	建立一个新项目	114
?D	程序的整体结构	115
?D	CSMTP 类对 SMTP 协议的包装	116
?D	邮件的包装类	122
?D	MIME 邮件格式	125
?D	编码与解码	128
?D	在界面上为邮件添加附件	129
?D	发送邮件的过程	130

☞ 步骤之二——实现邮件接收程序	131
?D 建立一个新项目	131
?D CPop 类对 POP3 协议的实现	132
?D 任务条通知区的编程原理	139
?D 使用 CTrayIcon 管理任务条通知区	141
?D 在程序中监测服务器上的邮件	144
实例七 FTP 客户程序	147
☞ 主要内容	147
?D 本例提要	147
?D 技术专题	148
※ FTP 协议及其实现方法	148
※ 使用 Microsoft Internet Transfer 控件	150
※ 直接使用 WinInet API	154
※ 使用 MFC WinInet 类	159
☞ 步骤——使用 WinInet 实现 FTP 客户程序	172
?D 建立一个新项目	172
?D 添加代码连接到 FTP 服务器/断开连接	173
?D 添加代码定位/读写远程文件	176
实例八 WWW 浏览器	180
☞ 主要内容	180
?D 本例提要	180
?D 技术专题	180
※ World Wide Web	180
※ 统一资源定位符	181
※ 超文本传送协议	185
※ 使用 Microsoft WebBrowser 控件	188
※ 使用 CHtmlView 类	192
☞ 步骤——实现简单的 WWW 浏览器	192
?D 建立一个新项目	192
?D 在项目中添加 WebBrowser 控件	193
?D 在程序窗口创建时创建 WebBrowser 控件	193

?D 完成程序的浏览任务	193
实例九 RAS 拨号上网	195
□ 主要内容	195
?D 本例提要	195
?D 技术专题	196
※ RAS 简介	196
※ 拨号与挂断	198
※ 连接管理	208
※ 电话簿管理	211
□ 步骤——实现 RAS 客户机	220
?D 建立一个新项目	221
?D 枚举计算机上已有的拨号连接	222
?D 异步启动拨号连接	223
?D 处理拨号事件	225
?D 终止拨号	228
实例十 网络新闻组	229
□ 主要内容	229
?D 本例提要	229
?D 技术专题	231
※ Usenet 新闻组	231
※ 网络新闻传送协议	231
□ 步骤——实现 NNTP 客户端	233
?D 建立一个新项目	233
?D 实现主窗口的分割	233
?D 设定参数	235
?D 登录到 NNTP 服务器并获取新闻组列表	236
?D 检索新闻组的文章	238
?D 读取特定的文章	241
?D 投递自己的文章	242

实例十一 Finger 查询	246
📖 主要内容	246
?D 本例提要	246
?D 技术专题	246
※ Finger 协议	247
📖 步骤——实现 Finger 客户机	248
?D 建立一个新项目	248
?D 添加并实现 CFingerSocket 类	248
?D 添加并实现 CFingerThread 类	252
?D 响应“Finger”按钮	256
附录 光盘使用说明	258

实例一 用 WinSock 实现网上聊天

主要内容

本例提要

随着 Internet 技术的发展,网上聊天已经成了一大时尚,相信大家都已经有过这样的经历。所谓网上聊天,就是指很多人登录到一个众所周知的聊天服务器,通过服务器的中转,可以看到所有人向服务器发送的内容。

在这一个实例中,我们将自己动手,使用 WinSock 实现一个聊天服务器和客户端的组合,用户可以通过客户端连接到服务器并进行网上聊天。当然,聊天时需要同时启动多个客户端。

作为演示,我们的聊天服务器支持 4 个聊天主题,在服务器启动时应该选定主题,如图 1-1 所示。事实上如果需要的话,你可以将聊天的主题扩展到任意多个。

相应主题的服务器启动之后,你就可以从网上的其他机器上启动聊天客户端,在客户端与服务器建立起连接之前,你需要指定聊天所用的名称、登录到的服务器名称(或其 IP 地址)以及参加的聊天主题。如果指定的机器上已经启动了相关主题的服务器,客户端就可以连接到服务器上并开始与别的已经连接上的客户进行聊天。

连接建立起来之后,从服务器或者任何一个客户端,我们都可以见到所有客户曾经说过的话,并且还可以将它们保存下来。服务器和客户端分别如图 1-2 和图 1-3 所示。

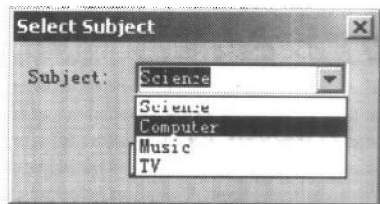


图 1-1 选定服务器的主题

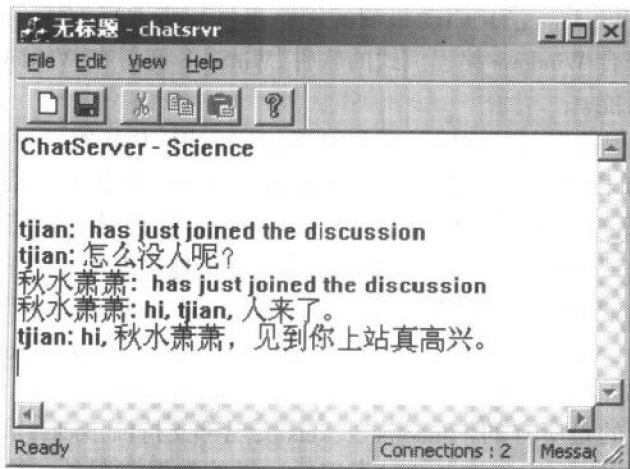


图 1-2 Science 主题的服务器

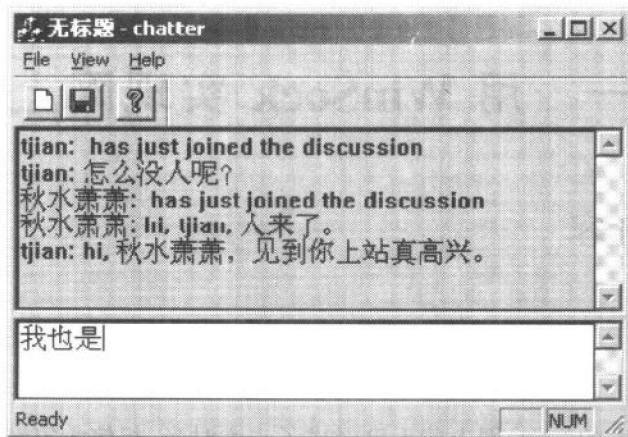


图 1-3 一个连接上的客户端

当然,我们的这个实例还是比较粗糙的,不过,如果你想做一个具有商业价值的聊天程序,只需要在这个实例的基础上进行一些改进就可以差强人意了。

? 技术专题

※ WinSock 简介

Socket(套接字)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。20 世纪 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,简称 WinSock。

Windows Sockets API 是 Microsoft Windows 的网络程序设计接口,它在继承了 Berkeley Sockets 主要特征的基础上,对其进行了重要扩充。这些扩充主要体现在它提供了一些异步函数,并增加了符合 Windows 消息驱动机制的网络事件异步选择方式。这些扩充有利于应用程序开发者编制符合 Windows 编程模式的软件,WinSock 使得在 Windows 下开发高性能的网络通信程序成为可能。

Windows Sockets 规范定义了应用程序开发者能够使用,并且网络软件供应商能够实现的一套库函数调用和相关语义。遵守这套 Windows Sockets 规范的网络软件,我们称之为 Windows Sockets 兼容的网络软件,而 Windows Sockets 兼容实现的提供者,我们称之为 Windows Sockets 提供者。一个网络软件供应商必须百分之百地实现 Windows Sockets 规范才能做到与 Windows Sockets 兼容。任何能够与 Windows Sockets 兼容实现协同工作的应用程序就被认为是具有 Windows Sockets 接口。我们称这种应用程序为 Windows Sockets 应用程序。

此外,该规范还定义了如何使用 API 与 Internet 协议族(IPS,现在最为流行的就是 TCP/IP)连接,尤其要指出的是所有的 Windows Sockets 实现都支持流 Socket 和数据报 Socket。流 Socket 提供了双向的、有序的、无重复并且无记录边界的数据流服务。数据报 Socket 支持双

向的数据流,但并不保证是可靠、有序、无重复的。也就是说,一个从数据报 Socket 接收信息的进程有可能发现信息重复了,或者和发出时的顺序不同。数据报 Socket 的一个重要特点是它保留了记录边界。对于这一特点,数据报 Socket 采用了与现在许多包交换网络(例如以太网)非常类似的模型。应用程序调用 Windows Sockets 的 API 实现相互之间的通信。Windows Sockets 又利用下层的网络通信协议功能和操作系统调用实现实际的通信工作。

Socket 实际上代表的是网络通信的一个端点,借助于它,用户所开发的 Socket 应用程序,可以通过网络与其他 Socket 应用程序进行通信。这与电话机之间的通话是十分相似的。

※ WinSock API 主要函数的使用

在 Windows(包括 Windows 95/98、Windows NT 和 Windows 2000)中进行 WinSock 应用程序的开发时,可供使用的编程语言很多,如 VC,Java,Delphi,VB 等。其中 VC 因为与 Windows 操作系统的结合紧密而被广泛使用。而且 VC(具体来说应该是 MFC)对原来的 Windows Sockets API 进行了一系列封装,使得 Socket 应用程序的开发变得更为容易,这一点我们将在后面介绍。但如果你是一个 WinSock 编程的初学者,那么了解一些最基本的 WinSock API 函数对将来的编程将会大有好处,因为它可以大大加深你对 WinSock 的了解。

在 VC 中进行 WinSock 的 API 编程开发,需要使用到下面三个文件:

- WinSock.h: 这是 Winsock API 的头文件。
- Wsock32.lib: WinSock API 链接库文件。在使用中,一定要把它作为项目的非默认的连接库包含到项目文件中去。
- WinSock.dll: WinSock 的动态链接库,位于 Windows 的安装目录下。

下面对 WinSock API 中主要函数的使用方法做一个简要说明:

1. 连接 WinSock 库

```
int WSASStartup (
    WORD wVersionRequested,
    LPWSADATA lpWSADATA
);
```

应用程序在使用 WinSock API 之前,必须调用该函数,只有该函数成功返回(表示应用程序与 WinSock 库成功地建立起连接),应用程序才可以调用其他 Windows Sockets DLL 中的函数。

2. 断开 WinSock 库

```
int WSACleanup (void);
```

WSACleanup 函数可以结束 Windows Sockets DLL 的使用。当应用程序不再需要使用 Windows Sockets DLL 时,必须调用此函数来注销使用,以便释放其占用的资源。

3. 套接字建立函数

```
SOCKET socket (
    int af,
    int type,
```

```
int protocol  
);
```

使用该函数可以建立使用特定协议的网络套接字。例如对于 UDP 协议,可以这样写:

```
SOCKET s;  
s = socket(AF_INET, SOCK_DGRAM, 0);  
或 s = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
```

4. 套接字绑定函数

```
int bind(  
    SOCKET s,  
    const struct sockaddr FAR * name,  
    int namelen  
);
```

s 是使用 socket 函数创建好的套接字, name 指向描述通信对象的结构体的指针, namelen 是该结构体的长度。该结构体中的分量包括:

- IP 地址: 对应 name.sin_addr.s_addr。
- 端口号: 对应 name.sin_port。端口号用于表示同一台计算机上不同的进程(即应用程序),其分配方法有两种。

第一种分配方法是,进程让系统为套接字自动分配一端口号,这只要在调用 bind 前将端口号指定为 0 即可。由系统自动分配的端口号位于 1024 ~ 5000 之间,而 1 ~ 1023 之间的任一 TCP 或 UDP 端口都是保留的,系统不允许任一进程使用保留端口,除非其有效用户 ID 是 0(即超级用户)。

第二种分配方法是,进程为套接字指定一特定端口。这对于需要给套接字分配一众所周知的端口的服务器是很有用的。指定范围在 1024 ~ 65536 之间。

- 地址类型: 对应 name.sin_family,一般都赋成 AF_INET,表示是 Internet 地址(即 IP 地址)。IP 地址的通常使用点分表示法表示,但它事实上是一个 32 位的长整数,这两者之间可以通过 inet_addr()函数转换。

5. 套接字监听函数

```
int listen(  
    SOCKET s,  
    int backlog  
);
```

listen 函数用来设定 Socket 为监听状态,这种状态表明 Socket 准备被连接。注意,此函数一般在服务程序上使用,其中的 backlog 参数可以设定最多可有多少个在未真正完成连接前的客户端的连接要求(目前最大值限制为 5,最小值为 1)。

6. 接受连接请求

```
SOCKET accept(  
    SOCKET s,  
    struct sockaddr FAR * addr,  
    int FAR * addrlen  
);
```

服务端应用程序调用此函数来接受客户端 Socket 连接请求, `accept()` 函数的返回值为一个新的 Socket, 新 Socket 就可用来完成服务端和客户端之间的信息传递与接收, 而原来 Socket 仍然可以接收其他客户端的连接要求。

7. 接收信息

```
int recv (
    SOCKET s,
    char FAR* buf,
    int len,
    int flags
);
```

此函数被用于面向连接的 Socket 中, 可以从连接的另一端接收信息。

8. 发送信息

```
int send (
    SOCKET s,
    const char FAR * buf,
    int len,
    int flags
);
```

此函数被用于面向连接的 Socket 中, 可以向连接的另一端发送信息。

了解了这些基本的函数之后, 我们还需要明确另一个重要的问题, 即服务端 Socket 与客户端 Socket 是如何交互的。目前最为常用的方式如下: 服务程序在一个众所周知的地址(其中包括端口的信息)监听对服务的请求, 也就是说, 服务进程一直处于休眠状态, 直到一个客户对这个服务的地址提出了连接请求。在这个时刻, 服务程序被唤醒并且对客户请求作出适当的反应。注意, 服务器与客户机之间的交互既可以是面向连接的(即基于流套接字), 也可以是无连接的(即基于数据报套接字)。其编程模型如图 1-4 所示。

※ MFC 对 WinSock API 的封装

在 MFC 中 VC 为 Socket 提供了相应的类 `CAsyncSocket` 和 `CSocket`。`CAsyncSocket` 提供基于异步通信的 Socket 封装功能, `CSocket` 则是由 `CAsyncSocket` 派生, 提供更加高层次的功能, 例如可以将 Socket 上发送和接收的数据和一个文件对象(`CSocketFile`)关联起来, 通过读写文件来达到发送和接收数据的目的, 而且 `CSocket` 提供的通信为同步通信, 数据未接收到或是未发送完之前调用不会返回。此外通过 MFC 类开发者可以不考虑网络字节顺序和忽略掉更多的通信细节。总之, MFC 对 WinSock API 的封装使得 WinSock 应用程序的开发变得简单, 大大减轻了程序员的负担。

我们的聊天程序就是使用 `CSocket` 类来实现的, 关于这些类的详细用法将在后面的例程中加以说明, 此处不再赘述。

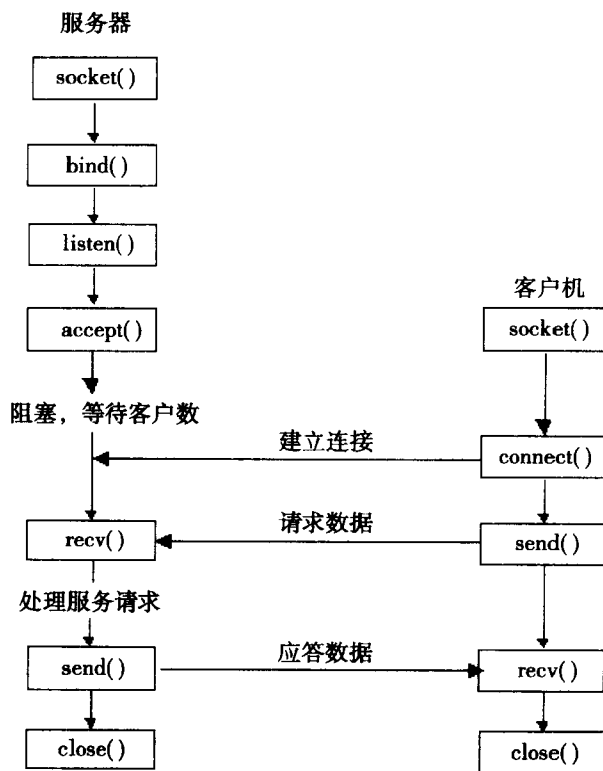


图 1-4 Socket 交互的编程模型

步骤之一——实现聊天服务器

下面我们将进行开发任务的第一步，即实现聊天服务器。根据前面介绍的编程模型，我们对服务器应该有一个框架性的认识，即我们的聊天服务器应该在一个客户机知道的端口上进行监听，并在有客户要求连接时接受请求并为其创建一个通话通道。

由于本书并不打算成为 Visual C++ 的入门教程，所以关于“如何创建一个项目”、“如何在 VC6 的项目中添加一个类”这样的问题，本书不会事无巨细地给出所有步骤，而只会指出一个大概，如果你忘了怎样做这样的事情，请参考相关的 VC 教程。

建立一个新项目

首先我们使用 Visual C++ 6.0 的 MFC Appwizard(exe)向导创建一个新的项目，注意，在向导的第一步选择创建单文档界面，第四步选中支持“Windows Sockets”特性，最后一步选择视类的基类为 CEditView 并将应用程序类、文档类和视类的名称分别改为 CServerApp、CServerDoc 和 CServerView，项目名称取为 ChatSrvr。