文章编号:1000 - 582X(2004)04 - 0049 - 04

基于 WinSock 的文件互传*

马永琴,李正文,汪刘艳
(成都理工大学 信息工程学院,四川 成都 610059)

摘 要:WinSock 是一组 API,用于 Internet 传输数据和交换信息。用户所开发的 Socket 应用程序可以通过网络与其它 Socket 应用程序进行通信;Delphi 中提供了 Socket 组件,支持 TCP/IP 协议,可以使 Internet 编程获得更大的灵活性。作者探讨了如何利用 Delphi 的 Socket 组件来实现客户端和服务器之间在非阻塞方式下对任意格式文件的相互传送。

关键词:Windows Sock;Socket;TCP/IP;Internet 编程;文件传输
中图分类号:TP319 **文献标识码**:A

在编写后台数据维护系统或 Internet 软件时,经常会遇到客户端和服务器数据文件的交互传送问题。一种方法可以通过 FTP 编程实现,但是这种方法要求服务器端必须配置 FTP 服务,还需要用户的 ID 和 Password,这期间还要有权限的配置,而且调试极不方便,实现起来较为繁琐;另一种方法可以利用 Delphi 中提供的文件流传送的方式,但是这种方法只能将文件从客户端传送到网络的服务器端,而不能实现文件的交互传送。WinSock(Windows Sock,Windows 环境下的网络编程接口)是一组 API(Application Programme Interface,应用程序接口),用于在 Internet 传输数据和交换信息,通过 WinSock 编程就避免了上述这些问题,可以获得更大的灵活性,并且不需要关心网络连接的细节。

用 WinSock 编程有一定难度,但在 Delphi 中并不需要直接与 WinSock 的 API 打交道,因为 TClientSocket 组件和 TServerSocket 组件封装了 WinSock 的大部分 API,使得 WinSock 编程大大简化[1]。

1 Socket 工作原理

Windows Sockets API(Windows 环境下的网络编程接口)是 Microsoft Windows 的网络程序设计接口,它在继承了 Berkeley Sockets(加利福利亚大学 Berkeley 分校为 UNIX 操作系统开发的网络通信接口)主要特征的基础上,又对它进行了重要扩充。这些扩充主要是提供了一些异步函数,并增加了符合 Windows 消息驱动特征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编制符合 Windows 编程模式的软件,它使在 Windows 下开发高性能的网络通信程序成为可能。

Socket(意为插座)实际上是指一个通信端口,借助于它,用户所开发的 Socket 应用程序可以通过网络与其它 Socket 应用程序进行通信,能够读写通过它连接的其他机器,而不用担心实际的网络软件的相关细节。Socket 连接建立在 TCP/IP 协议的基础上,同时也支持其他相关的协议,如 XNS(Xerox Network System)、DECnet 或 Novell IPX/SPX 协议。

通过 Internet 传输数据,至少需要一对 Socket。其中一个 Socket 在客户端,另一个 Socket 在服务器端。一旦客户端和服务器端的 Socket 都接通了,客户端和服务器端就可以相互通信了。两个 Socket 之间的连接既可以是面向连接的,也可以是面向无连接的。根据连接发起的方式及本地 Socket 连接目标,Socket 之间的连接可以分为 3 种类型:客户端连接、监听连接和服务器连接[2]。当网络通信需要一个 Socket 时,必须规定该 Socket 的属性,并使用一个函数请求网络软件提供一个标识该 Socket 的句柄。服务器和客户端 Socket 正是通过该句柄进行网络通信。Socket 数据结

* 收稿日期:2003 - 12 - 20
基金项目:国家自然科学基金资助项目(批准号:49874030)
作者简介:马永琴(1979 -),女,河北人,成都理工大学研究生,主要从事地球物理信号与信息处理技术领域的研究。

构必须包含本地主机和远程主机正确的协议端口和IP地址。

Delphi6分别用TClientSocket组件和TServerSocket组件来操纵客户端Socket与服务器端Socket的连接和通信。这两个组件用于管理客户端和服务器端的连接,本身并不是Socket对象,操纵Socket对象的是TcustomWinSocket类及其派生类,如TclienWinSocket类、TserverWinSocket类和TserverClientWinSocket类等。

2 Socket配置及实现方法

2.1 建立服务器端Socket

在一个应用程序的窗体中或数据摸块上放置一个TServerSocket组件,该程序就变成了一个TCP/IP服务器。当服务器处于监听状态时,就可以用TServerWinSocket来操纵服务器端的Socket对象。当客户端提出连接请求时,服务器接受了请求并建立了连接,此时就可以用TServerClientWinSocket来操纵服务器端Socket与客户端Socket的连接。

建立服务器监听客户的连接请求时,首先要设置TServerSocket的Port属性或Server属性,由此来指定端口号;确定了端口号后,调用TServerSocket的Open()方法或在设计期阶段把Active特性设为True来开始监听;进入监听状态后,通过TServerSocket组件的Socket属性返回服务器端的Socket对象(TServerWinSocket),并由此对象获得有关连接的信息。例如:Socket对象的SocketHandle属性可以获得Socket的Windows句柄,在直接调用WinSock的API时需要用到这个句柄,Socket对象的Handle属性可以访问从Socket连接中检索消息的窗口。

服务器(TServerSocket)进入监听状态后,当客户提出连接请求时,服务器就自动接受请求,然后建立连接并触发OnClientConnect事件。此时服务器端Socket与客户端Socket的连接由TServerClientWinSocket来操纵,TServerWinSocket的Connections属性可以返回一个数组,该数组由服务器中当前活动的连接(TServerClientWinSocket)所组成。如果要断开连接,只要在服务器端调用Close()方法或者把Active属性设置为False,就可以断开服务器对所有客户的连接,并退出监听状态。如果在客户端调用Close()方法,将只断开该客户与服务器的连接,不影响其他客户的连接。

2.2 建立客户端Socket

在一个应用程序的窗体中或数据摸块上放置一个TClientSocket组件,该程序就变成了一个TCP/IP客户,此时就可以用TClientSocket来操纵客户端的Socket对象。

客户连接服务器前首先要设置主机名或IP地址以及端口号。TClientSocket组件的Host属性用于指定服务器的主机名,如:http://www.sohu.com ; Address属性用于指定主机的IP地址,如:202.115.132.23;Port属性或是Service属性用于指定服务器端所使用的端口号[3]。指定好上述主机名或IP地址和端口号后,调用TClientSocket的Open()方法或在设计期阶段把Active属性设为True,客户端Socket就会向服务器端Socket提出连接请求。如果服务器端Socket正好处于监听状态,就会自动接受请求并建立连接。

客户与服务器建立连接后,通过TClientSocket的Socket属性返回客户端Socket对象(TClientWinSocket),并由此对象获得有关连接的信息。例如:Socket对象的SocketHandle属性可以获得Socket的Windows句柄,在直接调用WinSock的API时需要用到这个句柄,通过Socket对象的Handle属性可以访问从Socket连接中检索消息的窗口;通过Socket对象的AsyncStyles属性可以设置要捕捉那些信息。在客户端断开连接调用TClientSocket组件的Close()方法即可,此时服务器端将触发OnClientDisConnect事件,而客户端将触发OnDisConnect事件。

2.3 在网络上传输数据

当服务器端和客户端建立连接以后就可以通过Internet传输数据和文件,通常有两种方式:阻塞方式和非阻塞方式。

在Windows环境下,一般采用非阻塞方式。对于服务器端Socket,将其ServerType属性设为stNonBlocking,表示采用非阻塞方式进行连接。当客户端的Socket试图进行读或写时,服务器端的Socket就会得到通知,即OnClientRead事件或OnClientWrite事件。对于客户端Socket,将其ClientType属性设为ctNonBlocking,表示采用非阻塞方式进行连接。当服务器端的Socket试图进行读或写时,客户端的Socket就会得到通知,即OnRead事件或OnWrite事件。

与非阻塞方式不同的是,在阻塞方式下没有诸如OnRead或OnWrite等异步事件,Socket必须主动去读或写数据。在读或写操作完成之前,其他代码无法执行,应用程序处于等待状态,有可能影响整个应用程序的性能。对于服务器端Socket,将其ServerType属性设为stBlocking,表示采用阻塞方式进行连接。Depphi5为每个阻塞方式的连接自动分配一个新线程,当有一个客户正在进行读写操作时其他客户不必等待。

对于客户端 Socket，将其 ClientType 属性设为 ctBlocking，表示采用阻塞方式进行连接。为减少阻塞方式的副作用，将所有涉及到读写的操作放在一个单独的线程中，这样在 32 位的 Windows 环境下其他线程可以继续得到执行。

3　实例说明：

本程序主要借助于 TClientSocket 组件和 TServerSocket 组件实现网上交谈(chat)。在 chat 程序中只有一个 Form，在 Form 上同时放了 TClientSocket 组件和 TServerSocket 组件，表示这个程序既是 TCP/IP 服务器，也是 TCP/IP 客户。同时放置了两个 Memo 组件，用于显示双方交谈的内容。网上交谈是一种非标准的服务，因此交谈双方约定端口号(Port 属性)为 1024，Socket 之间连接类型为非阻塞方式(NonBlocking)。

1) 在 chat 程序刚开始运行的时候，首先要使服务器进入监听状态(TServerSocket 组件的 Active 属性初始设为 False)，在窗体的 OnCreat 事件中添加如下代码：ChatListenClick(nil)//Chat 菜单下 Listen 命令的单击事件函数。

该函数的具体代码如下：

```
ChatListen.Checked:= not ChatListen.Checked;//如果服务器已处于监听状态，就在 Listen 命令前显示"√"符号；反之取消 Listen 命令前的"√"符号
if ChatListen.Checked then//如果服务器不在监听状态，就让服务器进入监听状态
begin
  ClientSocket1.Active:= False;
  ServerSocket1.Active:= True;
  StatusBar1.Panels[0].Text:= 'Listening...';//处于监听状态，在状态栏显示"Listening…"
end
else if ServerSocket1.Active then//如果服务器已处于监听状态，就退出监听状态
begin
  ServerSocket1.Active:= False;
  ClientSocket1.Active:= True;
  StatusBar1.Panels[0].Text:= '';//服务器退出监听状态，清除状态条显示信息
end;
```

2) 服务器进入监听状态后，客户需要提出连接请求，该功能由 Chat 菜单下 Connect 命令来实现，对应的事件为 ChatConnectClick，其函数代码如下：

```
if ClientSocket1.Active then//如果客户与服务器已连接，就断开连接
    ClientSocket1.Active:= False;
    if InputQuery('Computer connect to', 'Address Name:',Server)//输入要连接的服务器地址
    then if Length(server) >0 then//如果输入的服务器地址不为空
        with ClientSocket1 do
        try  sHost:= server; Active:= True; end;//建立连接
```

3) 当客户提出连接请求的时候，在服务器端和客户端都将触发 OnConnect 事件，客户端通过这个事件在状态栏上显示"Connected to:"后跟服务器的主机名，代码如下：

```
StatusBar1.Panels[0].Text:= 'Connected to:'+Socket.RemoteHost
```

4) 当服务器监听到客户的连接请求后，就把 Memo2 清空，准备开始交谈，在 ServerSocket 的 OnClientConnect 事件中添加如下代码：Memo2.Lines.Clear

5) 当服务器接受了客户的连接请求，将触发 OnAccept 事件，在 ServerSocket 的 OnAccept 的事件中添加如下代码：

```
IsServer:= True;//本程序是服务器端
StatusBar1.Panels[0].Text:= 'Connect to:'+Socket.RemoteAddress;//状态条中显示客户端地址
```

6) 服务器接受了连接请求后，客户就可以在 Memo1 中键入字符开始交谈，在 Memo1 的 OnKeyDown 事件中添加如下代码：

```
if Key = VK_Return then//首先判断是否按下了 Enter 键，按下该键既可以使文本换行，也可以把光标所在行作为文本发出去
if IsServer then//如果 IsServer 为 True，就从 TServerSocket 的 Socket 属性返回服务器端的 Socket 对象，再由该对象的 Connections 数组的第一个元素得到 TServerClientWinSocket 对象，最后调用这个对象的 SendText()把光标所在行的文本发出去
    ServerSocket1.Socket.Connections[0].SendText(Memo1.Lines[Memo1.Lines.count-1])
    else ClientSocket1.Socket.SendText(Memo1.Lines[Memo1.Lines.Count-1]);//如果 IsServer 为 False，就从 TClientSocket 的 Socket 属性得到客户端的 Socket 对象，再调用这个对象的 SendText()把光标所在行的文本发出去
```

7) 当客户端收到文本后，就把收到的文本添加在 Memo2 后面，在 ClientSocket 的 OnRead 事件中添加如

下语句:Memo2. Lines. Add(Socket. ReceiveText)

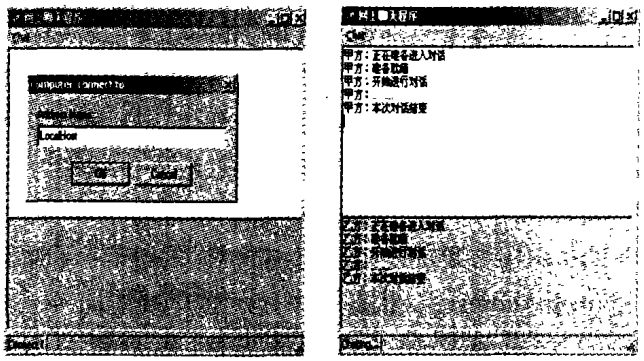
8) 当服务器端收到文本后也将其添加在 Memo2 的后面,在 ServerSocket 的 OnClientRead 事件中添加如下语句:Memo2. Lines. Add(Socket. ReceiveText)

9) 此外,在客户端还增加了处理错误的代码,在 ClientSocket 的 OnError 事件中添加如下代码:Memo2. Lines. Add('Error connectiong to:'+ Server);//显示出错的服务器端地址

ErrorCode: =0;//表明错误已被控制,系统不必再进行其他处理

至此已经可以利用这个程序在网上进行交谈,在编译这个程序后可以把执行文件分别放在两台具有固定 IP 地址的计算机上运行;也可以把这个程序在单机上调试,单击 Chat 菜单下的 Connect 命令后在"Address Name"框内键入"LocalHost",即可与本机进行连接。

程序运行结果如图 1 所示。



程序运行图(1): 建立连接前 程序运行图(2): 建立连接后

图 1 程序运行结果

4 结束语

基于 WinSock 文件传输实现起来较为灵活,只需配置 TCP/IP 协议即可;它既可以直接传送文本信息,也可以以流的形式传送任意格式的文件;此外,客户端和服务器端的应用程序还可以通过这种方法进行相互通信。如果传送的文件较大,可将其拆分成几个流分别进行传送,每次给出相应的格式和信息,接收方再将其合并到一个文件[6]。其中内存缓冲的大小必须选取适当,超出该范围可进行循环处理。这样读者可根据自己的需要在上述方法的基础上灵活地加入多线程及其他方面的处理,就可以成为功能强大的 Internet 通信和文件传送应用程序。

参考文献:

[1] Steve Teixeira Xavier Pacheco. Delphi 4.0 开发大全[M]. 北京:人民邮电出版社, 1999.

[2] 徐新华. COM CORBA 与 Internet 编程[M]. 北京:人民邮电出版社, 2000.

[3] 乔林. Delphi5 程序设计 - Internet 应用实务篇[M]. 北京:中国铁道出版社,2000.

[4] 黄国盛,梁平原. 通过 WinSock 实现 TCP/IP 网络通信[J]. 吉林大学学报,2002,23(2):66 - 69.

[5] 张树兵,庞勇. WinSock 网络通信程序的开发[J]. 华北工学院学报,2002,23(2):147 - 151.

[6] 秦绪佳等. Delphi 5.0 高级类参考详解[M]. 北京:清华大学出版社, 1999.

File Intertransmission Based on WinSock Component

MA Yong-qin, LI Zheng-wen, WANG Liu-yan

(Institute of Information Engineering, Chengdu University of Technology, Chengdu 610059, China)

Abstract: WinSock is a group of API, which is used to transmit data and exchange information on Internet, the Socket Application programme impoldered by users may communicate with other Socket Application programes by Internet. Delphi provides Socket components, which sustains the TCP/IP protocol and makes the Internet programming acquiring much more flexibility. This paper probes that how to realize transmission of files in any format under nonblocking mode between clients and servers by means of Socket components of Delphi.

Key words: WinSock; Socket; TCP/IP; Internet programming; file transmission

(编辑 成孝义)