

运用 Winsock 控件实现 C/S 网络通信

Using Winsock Control to realize C/S network communication

唐永红 龚安 (中国石油大学 山东 东营 257061)

摘要: Winsock 控件用于 windows 网络编程, 为访问 TCP 和 UDP 网络服务提供了方便途径。本文介绍了 VB 中 Winsock 控件的属性、方法、事件, 例举了在 VB 中用 Winsock 控件实现网络通信的一种简易方法。

关键词: Winsock 控件 TCP UDP 网络通信

1 引言

随着数据通信技术的发展特别是计算机技术的发展, 计算机网络的应用越来越广泛, 编写网络通信软件的工具和方法有很多, 而使用 Winsock 是其中一种简单有效的方法, 由于 Winsock 控件易于使用, 因而用处颇为广泛。本文介绍了基于 VB 环境运用 Winsock 控件实现服务器/客户机模式的一种简易网络编程方法。

2 Winsock 简介

TCP/IP 协议是 Internet 网最重要的协议, VB 提供了 Winsock 控件用在 TCP/IP 的基础上进行网络通信, 可以方便快捷地与远程计算机建立连接。Winsock 包含用户数据文报协议 UDP 和传输控制协议 TCP。TCP 协议是一个面向连接、全双工、可靠的、按序传送的传输层协议, 允许创建和维护与远程计算机的连接, 连接的计算机可彼此进行数据传输。如果创建客户应用程序, 就必须知道服务器计算机名或者 IP 地址 (RemoteHost 属性), 还要知道进行“侦听”的端口 (RemotePort 属性), 然后调用 Connect 方法。如果创建服务器应用程序, 就应设置一个收听端口 (LocalPort 属性) 并调用 Listen 方法。当客户计算机需要连接时就会发生 Connection - Request 事件, 为了完成连接, 可调用 ConnectionRequest 事件内的 Accept 方法。建立连接后, 任何一方计算机都可以收发数据。为了发送数据, 可调用 SendData 方法, 当接收数据时会发生 DataArrival 事件, 调用 DataArrival 事件内的 GetData 方法就可获取数据。一个 TCP 连接需要客户机和服务器都要有一个 IP 地址和端口, 所标识的端口一定不能再让其它应

用程序使用。许多端口已经定义为标准端口, 因此应用程序不能使用 (例如, Web 浏览器用端口号 80 连接 Web 服务器)。一个最简单的方法是使用 1000 以上的端口号, 因为这些端口号大多数未被占用。

UDP 协议是一个无连接、不可靠的协议, 跟 TCP 的操作不同, 计算机并不建立连接。另外 UDP 应用程序可以是客户机, 也可以是服务器, 为了传输数据, 首先要设置客户计算机的 LocalPort 属性, 然后, 服务器计算机只需将 RemoteHost 设置为客户计算机的 Internet 地址, 并将 RemotePort 属性设置为跟客户计算机的 LocalPort 属性相同的端口, 并调用 SendData 方法来着手发送信息。于是, 客户计算机使用 DataArrival 事件内的 GetData 方法来获取已发送的信息。由于 UDP 协议没有建立连接, 通信效果高, 但可靠性不如 TCP 协议高, 使用 UDP 发送消息, 有时会出现收不到消息的情况。因此在少量数据传送、可靠性要求不高的应用环境可选用 UDP 协议。

3 Winsock 控件的属性、方法、事件

Winsock 控件放到窗体上后就可以把它看成对象, 它具有三要素: 属性、事件、方法。Winsock 控件具有许多属性、方法、事件, 下面介绍其常见的属性、方法和事件。

3.1 Winsock 控件的属性

LocalHostName	本地机器名
LocalIP	本地机器的 IP 地址
LocalPort	本地机器用于通信的 TCP 端口
RemoteHost	远程机器名
RemoteHostIP	远程机器的 IP 地址

RemotePort	远程机器用于通信的 TCP 端口
Protocol	Winsock 控件所使用的通信协议——或者是 TCP, 或者是 UDP
State	连接的当前状态即控件的状态(只读)

State 特性存储了 TCP 连接的当前状态,由于 State 是只读的,因此不能对该连接状态置值,但却可以用各种方式使用状态值。当选用 Winsock 控件作相关动作时,必须经常地读取 State 属性,以了解当前控件的动作过程。

3.2 Winsock 控件的方法

一个连接的生存周期包括以下几步:

- (1) 服务器应用程序监听一个特殊的窗口
- (2) 客户程序请求和服务器程序建立连接
- (3) 服务器程序接受连接
- (4) 数据在客户程序和服务器程序之间传送
- (5) 客户程序或服务器程序关闭连接

方法	含义
Accept	接受一个到来的连接
Close	关闭一个 TCP 连接
Connect	向远程机器发出连接请求
GetData	获得当前数据块
Listen	创建并监听一个套接字
PeekData	不从输入队列删除数据,获得当前数据块的一个拷贝
SendData	把数据传送给远程机器。

Winsock 控件的几个主要方法: Listen : 将其设置为聆听模式,才有可能接收另一方的连接请求; SendData : 将数据传送给远程计算机,一般传送字符串数据时,只要将字符串当成自变量传送即可,如果是二进制数据的话,传输数据就必须以字节数组的方式储存; GetData : 截取目前的数据块,将其存储在参数所定的变量中; Accept 仅适用于 TCP 服务器程序,在处理 Connection Request 事件时用这个方法接受新的连接,执行此方法后,二者的连接才算成功,才能传输数据; Close : 将终止双方的连接,而且对方也会引发 close 事件。

3.3 Winsock 控件的事件

Close	当远程机器关闭连接时触发
Connect	表明连接已可以进行数据传送

ConnectionRequest	当远程机器请求连接时产生
DataArrival	当新数据到达接收缓冲区时发出
Error	表明发生了一个后台错误
SendComplete	表明数据发送完毕
SendProgress	表明数据传送进度

Winsock 控件的几个主要事件: ConnectionRequest : 当远程计算机向服务器请求连接时,将引发服务器端计算机事件,激活事件之后 RemoteHost IP 和 RemotePort 属性会储存有关客户端的信息; Close : 当远程计算机关闭连接时会引发此事件; Connect : 发生于连接动作完成时; DataArrival : 当有新数据到达时,引发此事件。

4 C/S 模式

C/S 又称 Client/Server 或客户/服务器模式,是网络计算理论中的重大技术创新,相对传统的主机集中模式和文件服务模式而言,C/S 结构具有减少网络流量,提高事物处理速度等优点。目前两层或多层 C/S 结构技术在数据库应用领域得到了广泛应用。

5 WinSock 控件的应用实例

以上介绍了 WinSock 的几个主要属性、方法、事件,下面通过一个简单客户机服务器程序,进一步讨论 WinSock 的编程方法。程序采用的是面向连接的 TCP 协议。

5.1 创建服务器应用程序

- (1) 创建新的 Standard. EXE 工程
- (2) 将窗体的标题改为“服务器”
- (3) 在窗体上放置一个 winsock 控件 (WinSocks-erver), 在窗体的 load 事件中, 设置 winsock 控件的本地端口号, 并调用 listen 方法开始侦听

```
Winsockserver.localport = 2002
```

```
Winsockserver.listen
```

- (4) 在 winsock 控件的 connectionrequest 事件中, 首先写入如下语句:

```
If Winsockserver.State < > sckClosed Then Winsockserver.Close
```

然后调用 winsock 控件的 Accept 方法接受客户机的连接

```
Winsockserver.accept requestID
```

(5) 在 winsock 控件的 DataArrival 事件中,使用 winsock 控件的 GetData 方法捕获发送端发来的消息,然后再利用 winsock 控件的 SendData 方法原封不动地将客户机发来的数据发送回去(作为一种最简单的响应)

```
Dim s as string
Winsockserver. getdata s
Winsockserver. senddata s&" ok"
```

(6) 在窗体的 unload 事件中,调用 winsock 控件的 Close 方法主动关闭连接

```
Winsockserver. close
```

(7) 在 winsock 控件的 Close 事件中,调用 winsock 控件的 Close 方法响应客户机的断开连接请求,并使用 unload 命令关闭窗口体

```
Winsockserver. close
Unload me
```

可在服务器端窗体上增加一个标签(用于显示服务器的当前状态如:等待连接、连接已建立、连接已断开等)以及其它的一些功能。利用控件数组,可将服务器端改进为可同时为多台客户机服务。

5.2 创建客户机应用程序

(1) 工程中创建一个新的窗体,将窗体改名为“客户机”

(2) 在窗体上放置四个文本框:分别用来输入服务器 IP 地址(text1)、服务器端口(text2)、发给服务器命令(text3)、以及从服务器返回的响应(text4)及一些相应的标签(用于提示对应文本框的作用)、两个按钮(一个按钮用于与服务器建立连接,另一个按钮用于向服务器发送数据)和一个 winsock 控件(Winsockclient)。

(3) 在窗体的 load 事件中,设置 winsock 控件的本地端口号

```
Winsockclient. localport = 2000
```

(4) 在连接按钮的 click 事件中,设置 winsock 控件的远地 IP 地址和远地端口号,并调用 winsock 控件的 connect 方法向服务器发起连接

```
Winsockclient. RemoteHost = text1. text
Winsockclient. RemotePort = val( text2. text)
Winsockclient. connect
```

(5) 在发送按钮的 click 事件中,调用 winsock 控

件的 SendData 方法向服务器发送数据

```
Winsockclient. senddata text3. text
```

(6) 在 winsock 控件的 DataArrival 事件中,使用 winsock 控件的 GetData 方法捕获发送端发来的消息,并将消息显示在响应文本框中(可设置 Enabled 属性为 false,用于禁止用户的输入)

```
Dim s as string
Winsockclient. getdata s
Text4. text = s
```

(7) 在窗体的 unload 事件中,调用 winsock 控件的 Close 方法主动关闭连接

```
Winsockclient. close
```

(8) 在 winsock 控件的 Close 事件中,调用 winsock 控件的 Close 方法响应客户机的断开连接请求,并使用 unload 命令关闭窗口体

```
Winsockclient. close
Unload me
```

启动运行上述程序,一个简单的 C/S 通信就实现了,由于服务器进程总是先于客户进程启动,没有服务器端应用程序的支持,客户端应用程序虽然可以运行但不能实现网络功能。因此先启动服务器后启动客户机。

6 结束语

在 VB 环境下有效利用 Winsock 控件,开发一种基于 TCP 协议的服务器和客户机应用程序,能够较好地实现服务器端与客户端之间数据信息及文件的传输。TCP 和 UDP 协议在使用中各有特点,如果灵活使用,可以得到很好的效果。

参考文献

- 1 韩静,用 Winsock 控件实现点对点通信[J],安徽工业大学学报,2004.1,第 21 卷第 1 期:57-59。
- 2 洪萍,使用 WinSock 控件开发网络程序[J],计算机与现代化,2004(7):95-98。
- 3 高俊光等,基于 VB 的 Winsock 控件的原理与应用,应用科技,2004.3,第 31 卷第 3 期:13-14。
- 4 毕传林等,Winsock 控件在网络通信软件中的应用,九江职业技术学院学报,2004.1:27-28。