

基于 Delphi7 的网络通讯系统的设计与实现

周维柏 李蓉 (华南师范大学增城学院 511363)

摘要:本文先简单介绍了 Delphi 实现即时通讯的原理、与 Window 的关系以及网络接口编程的方法。接着使用 Socket 接口对网络通讯系统进行设计与实现。

关键词:Socket 接口 网络通讯

1 引言

计算机网络是由一系列网络通信协议组成的,其中的核心协议是传输层的 TCP/IP 和 UDP 协议。微软的 Windows Socket 规范(简称 winsock)对 Berkley 的套接字规范进行了扩展,利用标准的 Socket 的方法,可以同任何平台上的 Socket 进行通信;利用其扩展,可以更有效地实现在 Windows 平台上计算机间的通信。在 Delphi 中,其底层的 Socket 也应该是 Windows 的 Socket。Socket 减轻了编写计算机间通信软件的难度,Inprise 在 Delphi 中对 Windows Socket 进行了有效的封装,使得用户可以很方便地编写网络通信程序。本文就在 Delphi 中如何利用 Socket 编写通信程序进行探讨。

2 Delphi 实现即时通讯的原理

用 Delphi 设计即时聊天通讯的工具,主要采用 Client/Server 结构,并用 WinSock(即 Windows Sockets)写,实现服务器监听、客户端发送请求信息。这种方法需要两个 Socket,一个作为 Client,发送请求信息;另一个作为 Server,接收并应答请求。

对 WinSock 实现的 C/S 结构的实时聊天工具,又可采用多种技术方案。本文采用基于 TCP 有连接的方案。这种方案是 C/S 两端均采用 TCP 协议,这样做的好处是网络传输可靠,也能一次传输更多的数据,不必关心发送数据的大小,Socket 会自动将它分割成合适的 TCP 报进行发送,然后会根据接收到的 TCP 报拼装之后提交用户。

对 Delphi 而言,由于 VCL 组件提供了 TtcpServer 和 TtcpClient 控件,使在 Delphi 实现 TCP 协议的通讯变

得更容易。TtcpServer 控件用来做服务器端,指定它的 Port 口之后,运用 Open 方法就可以在 Server 上开始监听该端口,等待 TtcpClient 的连接。TtcpClient 需要指定 Host、Port 等属性,然后运用 Open 方法,就可以向 Server 发送连接请求了。

3 Delphi7 Studio 的 winsock 编程原理

3.1 Delphi7 的 Socket 组件与原理

在 Delphi7 Studio 中,提供了 Socket 组件,以便创建使用 TCP/IP 和相关协议,与其他计算机系统进行通讯的应用程序。由于 Delphi7 已经对 WinSock 进行了封装,因此可以使用其 Socket 组件通过网络连接进行读写操作。

Delphi7 将 Socket 连接分为 Client 连接、Listening 连接和 Sever 连接三种。每个 Socket 连接绑定了特定 IP(或主机)以及该主机的某个端口号。

建立 Socket 连接后,可以设置响应该连接的特定事件的代码,以便通过该连接实现数据的读写操作。

3.2 TtcpServer 组件和 TtcpClient 组件

TtcpServer 组件和 TtcpClient 组件是经 Delphi7 封装的 Windows Sockets,它们均继承 Socket 组件,并定义在 Delphi7 的 Socket 单元中。

3.3 TudpSocket 组件

使用 TudpSocket 组件,可以创建基于 UDP 的应用。增加一个 TudpSocket 组件到某窗体或是 Data module(数据模板)中,可以使应用程序变为一个 UDP/IP 的 Client 和 Server。TudpSocket 指定一个期望的连接到 UDP/IP 服务器,当该连接打开时管理该连接。或者当该连接断开时终止该连接。TudpSocket 也可以用于监听 UDP/IP 连接的需求,并在接到请求时建

立 UDP/IP 连接。

4 网络通讯系统的设计与实现

4.1 系统需求

功能如下:

网络聊天功能、局域网内用户列举和搜索功能、列表的管理功能、发送 Email 功能、个性化浏览器功能、文本类型文件打开和聊天数据保存功能、用户信息记录和信箱信息记录功能。

4.2 系统结构

系统结构如图 1。

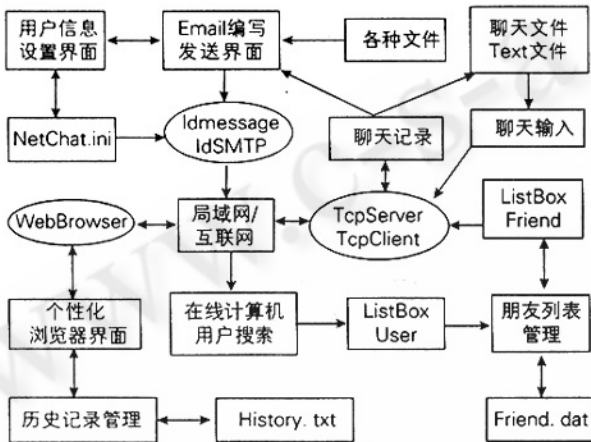


图 1 网络通讯管理系统的总统结构

4.3 系统的实现

4.3.1 Ttcpserver 和 TtcpClient 实现即时通讯程序

(1) 窗体的设计。该程序为对等的聊天程序,每个用户计算机既是 Client,又是 Server。为此在窗体中添加了 TtcpServer 和 TtcpClient 组件各一个,其中 TtcpClient1 作为 Client,用于发送请求;TtcpServer1 作为 Server,接收并应答请求。

先设置 TtcpServer1 和 TtcpClient1 的 Text 码。设 TtcpClient1 的端口 RemotePort = '5858'。TtcpServer1 的 LocalPort = '5858'。接着设置 TtcpServer1 的 BlockMode 属性为 bmThreadBlocking,设置 TtcpClient1 的 Block-Mode 属性为 bmBlocking。

另外在窗体中放入两个 Memo 控件,其中 Memo-Send 用于书写发送数据,MemoChat 用于书写发出和

收到的数据。并增加一个用于填写聊天对象的 Edit 控件 EditChat 和一个发送数据 Button 控件 BTNSend。

(2) TclientDataThread 的编制。为了提高应用程序的效率,编写一个 TclientDataThread 自定义线程类,用于完成客户端数据的发送工作。先进行类声明,接着 Create 构造函数。部分代码如下:

```

Constructor TclientDataThread. Create ( createSuspended: Boolean ); //Create 构造函数
Begin
    Inherited Create ( CreateSuspended );
    FreeOnTerminate := True; //终止
    ListBuffer := TStringList. Create;
End;
Procedure TclientDataThread. Terminate; //Terminate 终止过程
Begin
    ListBuffer. Free; //释放
    Inherited;
End;
Procedure TclientDataThread. execute; //Execute 执行过程
Begin
    Synchronize ( synchAddDataToControl ); //同步
End;
Procedure TclientDataThread. synchAddDataToControl; //同步增加数据过程
Begin
    TargetList. AddStrings ( ListBuffer );
End;
(3) TcpServer 的 Accept 过程编制。当一个 Client Socket 连接被 TcpServer 接受时将产生一个 OnAccept 事件。对于面向连接的 Socket 服务,应在该事件出现时对 Client Socket 的连接进行处理。TcpServer1 的 Accept 过程代码如下:
procedure TFormMain. TcpServer1Accept ( Sender: TObject; ClientSocket: TCustomIplClient );
var
    S: string;
    DataThread: TclientDataThread; //声明一个自定义的线程类
  
```

```

Begin
    DataThread := TClientDataThread.Create(True);
// 创建线程
    DataThread.TargetList := MemoChat.Lines; // 设置 TargetList 到显示区域
    DataThread.ListBuff.Add(ClientSocket.LookUpHostName(ClientSocket.RemoteHost) + '来自' + ClientSocket.RemoteHost + '说:'); // 装入线程 ListBuff 客户端主机名 + 客户端远端主机 IP
    S := ClientSock.ReceiveIn; // 接收数据并赋给 s
    While s < > '' do
    Begin
        DataThread.ListBuffer.Add(s);
        S := ClientSocket.ReceiveIn;
    End;
    DataThread.Resume; // 刷新线程
End;

```

(4) 发送数据消息的代码。要使用 TCP 组件进行聊天,需要进行初始化工作,即设置其相应属性,并打开 TcpServer1 和 TcpClient1。为使 TcpClient1 组件正常工作,需设置其 RemoteHost,对于不同的主机聊天这个值不同,在发送信息前要重新设置。发送数据消息的部分代码如下:

```

procedure TFormMain.ButtonSendClick(Sender: TObject);
var
    I: integer;
    Str: String;
Begin
if trim(MemoSend.Lines.Text) < > '' then // 消息不能为空
    begin
        if FindComputer(trim(EditChat.Text)) then // 对方计算机在线
            begin // 设置对方主机,调用'localhost'
                TcpClient1.RemoteHost := trim(EditChat.Text);
                TcpClient1.RemotePort := trim(EditPort.Text); // 设置对方端口号
            Try
                if TcpClient1.Connect then // 连接
                    Begin

```

```

For I := 0 to MemoSend.Lines.Count - 1 do
    Begin
        Str := TcpServer1.LocalHost + '说' + TcpClient1.RemoteHost + '说'
            + #13#10 + MemoSend.Lines[I];
        MemoChat.Lines.Add(Str); // 增加数据发送信息
    End;
    For I := 0 to MemoSend.Lines.Count - 1 do
        TcpClient1.SendIn(MemoSend.Lines[I]); // 发送数据信息
    Finally
        TcpClient1.Disconnect; // 断开
        MemoSend.Clear; // 内容清除
    End;
End
Else
Begin
    Application.MessageBox(pchar(''), Pchar(''), Mb_Ok + MB_ICONERROR);
    ...

```

4.3.2 局域网内用户列举和搜索

使用 Windows API 来实现,首先列举工作组,然后按工作组搜索全部的计算机,包括其主机和 IP 地址。

根据计算机主机名得到 IP 地址是通过建立 Socket 连接,并由 GetHostByName 函数得到计算机的地址,通过 P_JrPlnAddr(phe.h_addr_list) 转换为 Ip 地址。

列举局域网内的工作组是通过 WNetOpenEnum 函数和 WNetEnumResource 函数,枚举计算机的资源,分析并得到所有局域网内的工作组信息。WNetOpenEnum 函数的原型是 WNetOpenEnum(Handle hEnum, Lpdword lpcCount, lpVoid lpBuffer, Lpdword lpBufferSize)。

列举工作组内的全部在线计算机也是通过 WNetOpenEnum 函数和 WNetEnumResource 函数,枚举工作组内的计算机资源,分析并得到计算机的名称。

列举局域网内全部在线计算机。通过调用上述函数,实现查询局域网内全部在线计算机的代码。

判断计算机是否在线是通过 Socket 连接方法来实现的。部分代码如下:

```

....
Result: = False;
Wsastartup(2, wsaData);
HostEnt: = Gethostbyname ( pchar ( Computer-
Name) );
If hostEnt = nil then Result: = false
Else Result: = True;
WsaCleanup;
....

```

4.3.3 发送 SMTP 邮件程序

界面如图 2。运用 VCL 组件中的 Indy 组件,关键是采用了 IdMessage 控件和 IdSMTP 控件。

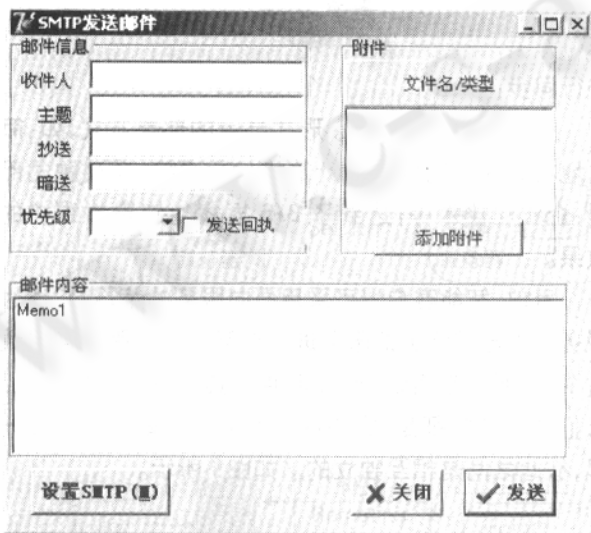


图 2

4.3.4 个性化浏览器功能

最简单的方法是直接运用 WebBrowser 控件。

4.3.5 文件类型打开和聊天数据保存功能

直接使用 OpenFileDialog、SaveDialog 控件,并利用 Memo 控件的 LoadFromFile 和 SaveToFile 方法。

参考文献

- 1 本书编委会, 编著 Delphi 编程篇, 电子工业出版社, 2004。
- 2 蒋方帅 编著, Borland Delphi 程序设计, 清华大学出版社, 2002。
- 3 甘志, Delphi 7 应用开发丛书: Delphi 7 高级应用开

发篇, 清华大学出版社, 2003。

- 4 张世明 编, Delphi 软件开发技术, 人民邮电出版社, 2004。
- 5 陈省 编著, Delphi 深度探索(第二版), 电子工业出版社, 2004。