

计算机网络开放实验报告

开放式聊天程序

西安交通大学电气工程学院电气 16 班

马怡轩 01041173

指导教师：薛涛

实验日期：2004 年 5 月

报告完成日期：2004 年 5 月 28 日

1 实验目的

编写一个类似 OICQ 的聊天程序（包括客户端和服务端），可供多个用户进行聊天。

2 实验任务

(1) 登录功能。客户端登录到聊天服务器，服务器管理所有登录的客户，并将客户列表发送给各个客户显示。

(2) 客户可以通过服务器转发，实现一对一和多对多聊天。

(3) 实现呼叫功能。当客户端程序连接服务器时，通过服务器搜索所要呼叫的客户，如果检测到此用户且该用户正处于联网状态，则服务器通知此用户的客户端程序响应主叫方客户端程序，然后在主叫方和被叫方建立连接后，双方就可以聊天或进行其它的通信。

(4) 客户端程序应该可以实时显示目前其它用户的状态（例如好友信息上、下线）。

(5) 应该具有易用、美观的图形界面。

3 程序介绍

运行环境：Windows9x/NT/2000/XP

开发环境：Windows XP

Delphi 6.0

程序简介：本程序主要使用的 Delphi 6.0 组件包括：Tform, Tpanel, Tlabel, Ttoolbar, TtoolButton, Tmainmenu, Tstatusbar, Timagelist, Tmemo, Tclientsocket, Tserversocket 和 Ttimer 等。其中主要是利用 Ttoolbar, ToolBar 以及 Timagelist 组件来实现带背景的工具条；利用 Tstatusbar 组件来实现状态条；利用 Tmemo 组件实现文本的输入与显示；利用 Tclientsocket 和 Tserversocket 进行 TCP/IP 的网络编程；利用 ShellExecute 函数实现电子邮件的动态链接。完成的内容包括（1）给应用程序加标题与图标；（2）带图像的菜单制作以及互斥菜单项的制作；（3）带背景图案的工具条的制作，在该工具条按钮上可以同时显示文本和图像；（4）网络通信组件的使用方法与技巧；（5）窗口的分割；（6）电子邮件与主页的动态链接；（7）命令行参数编程方法。

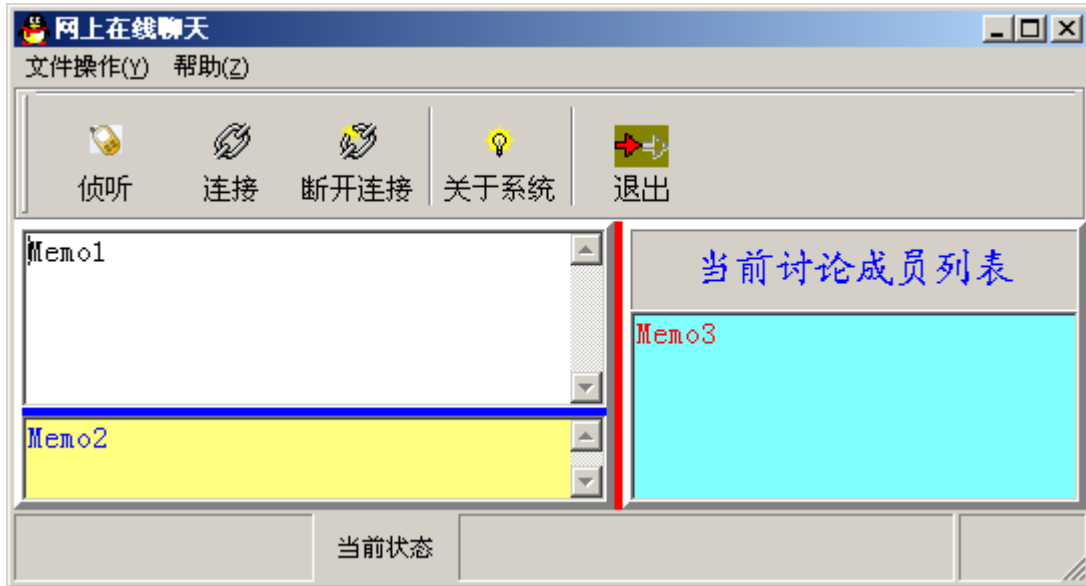
4 制作步骤

- (1) 建立工程及设定相关设置
- (2) 窗口属性的设置
- (3) 向窗口“w_cww”中添加图像列表组件对象
- (4) 向窗口“w_cww”中添加菜单
- (5) 向窗口“w_cww”中添加工具条
- (6) 向窗口“w_cww”中添加状态条
- (7) 分割窗口以及制作各种显示区域
- (8) 向窗口“w_cww”中添加其他组件对象

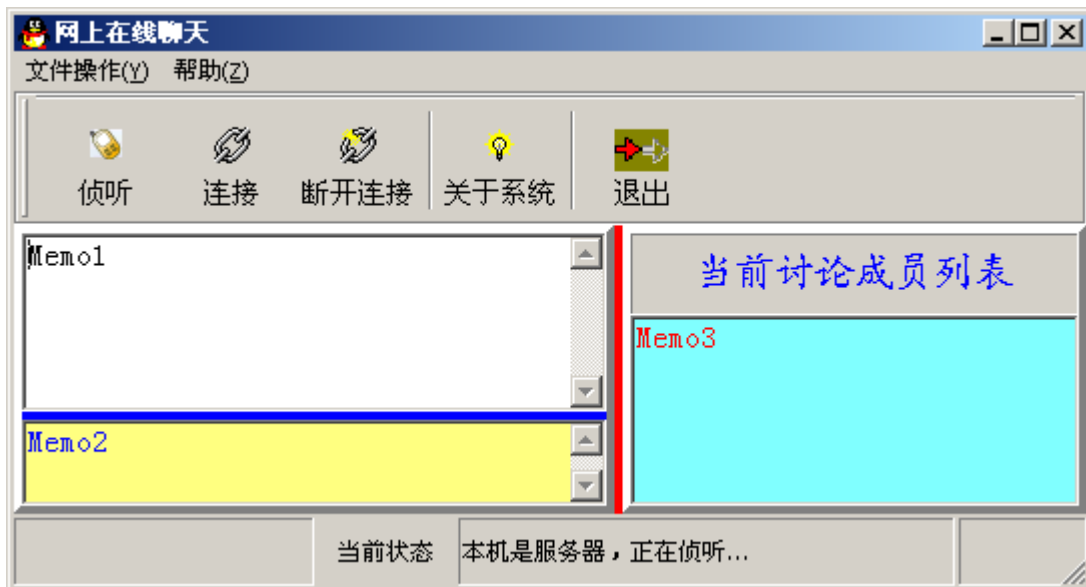
- (9) 自带超级链的系统信息显示窗
 - (10) 给“AboutBox”窗口添加代码
 - (11) 编写“w_cww”窗口的代码
- 以上是程序的基本编写流程

5 程序运行过程

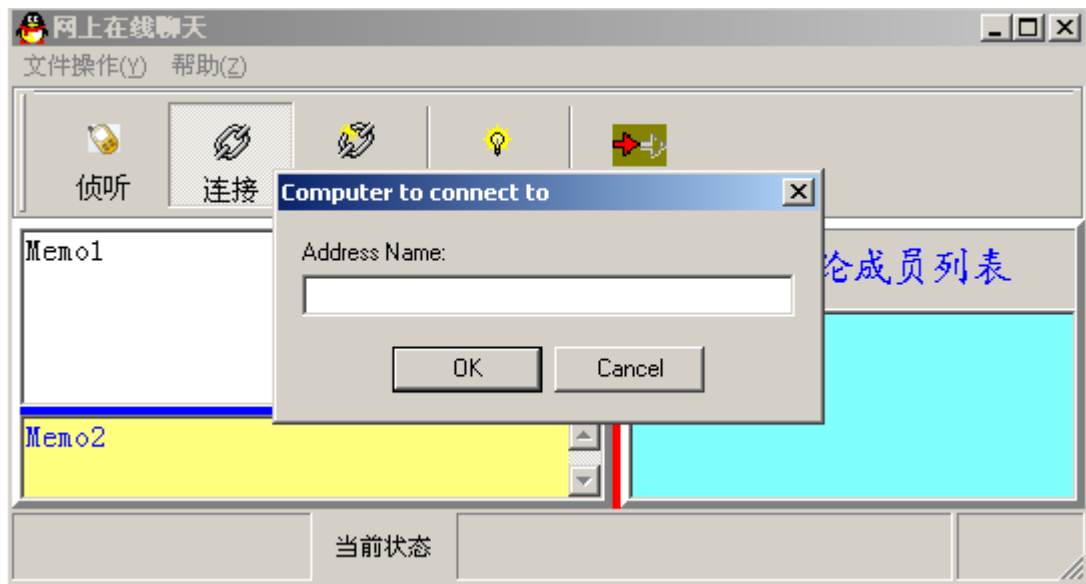
服务器开始运行



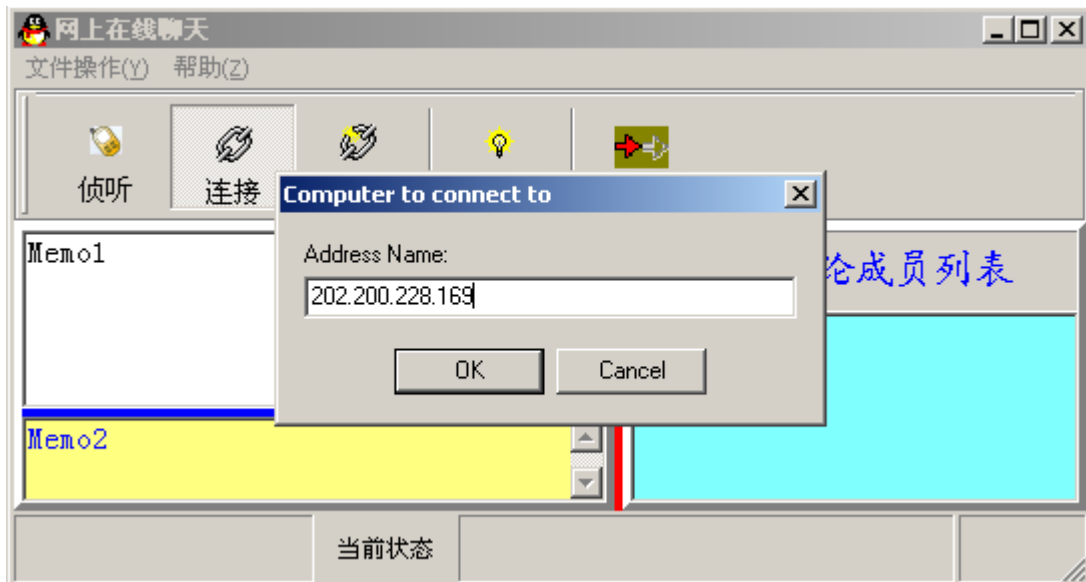
服务器侦听



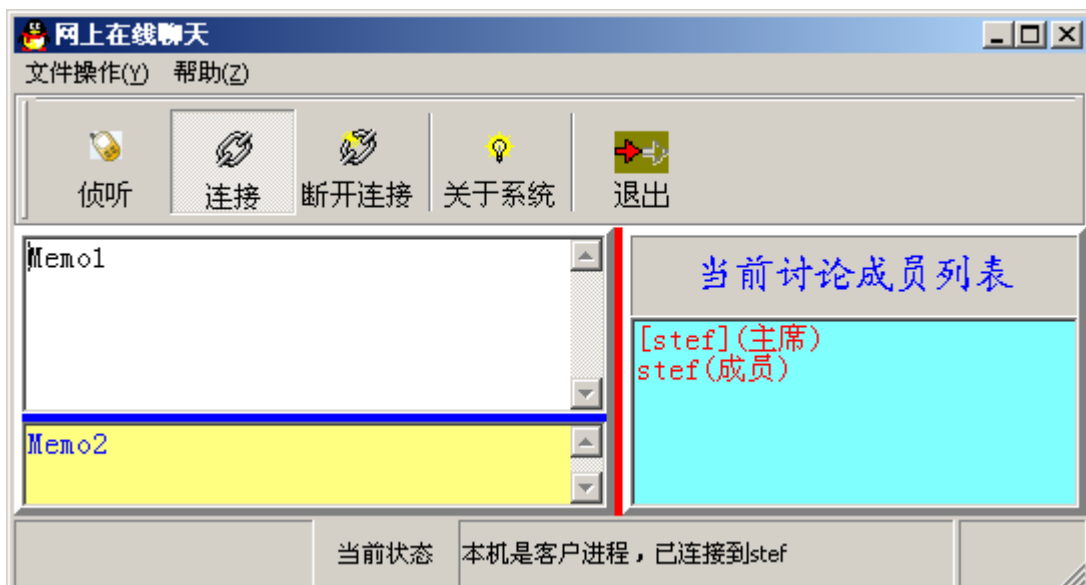
客户端开启并请求连接



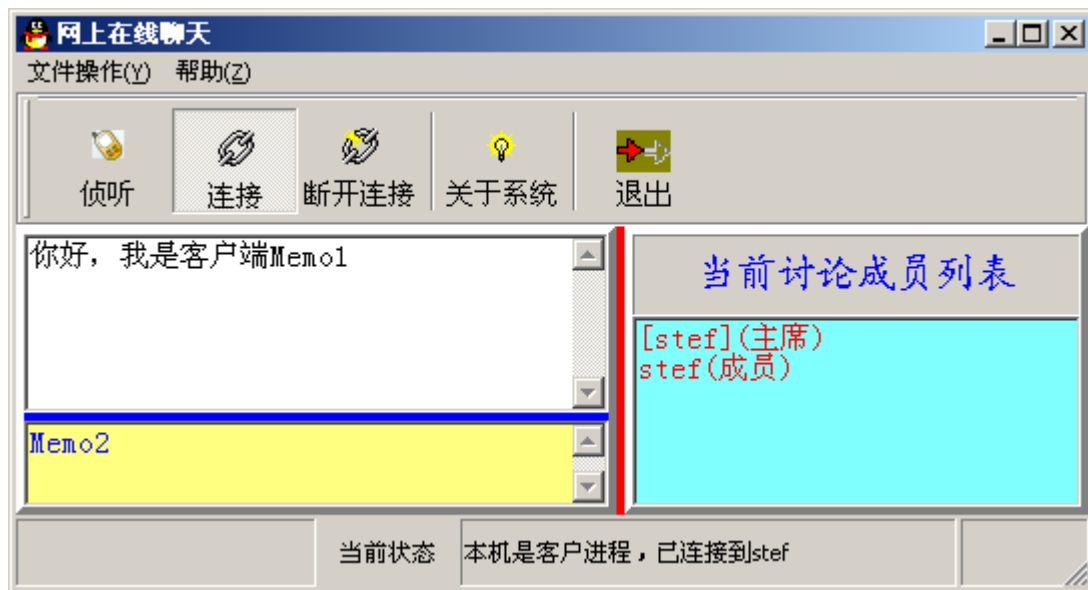
客户端输入服务器端 IP



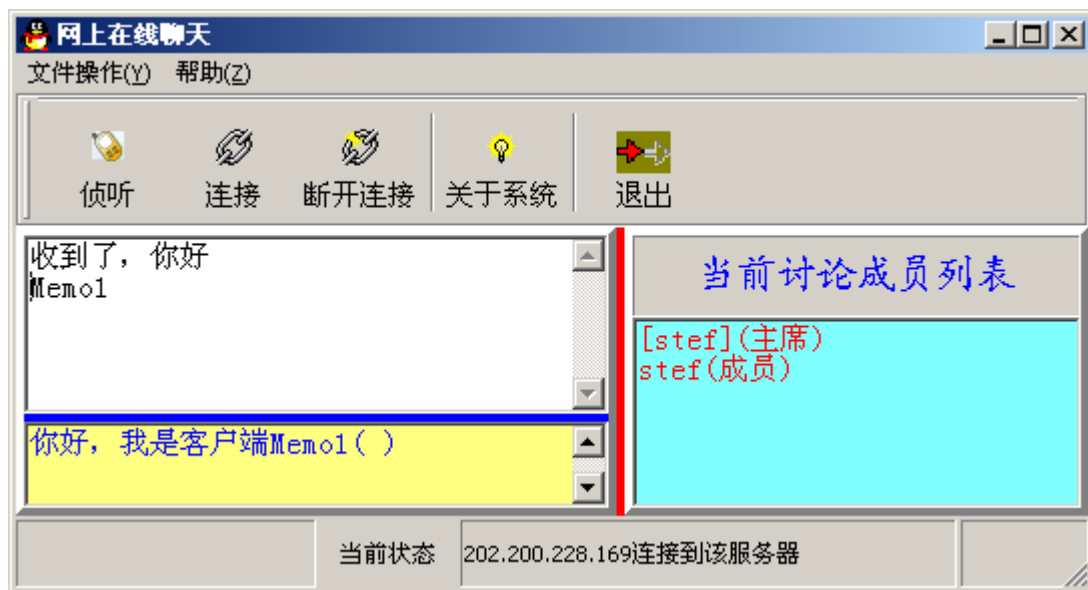
点击 OK 显示连接成功



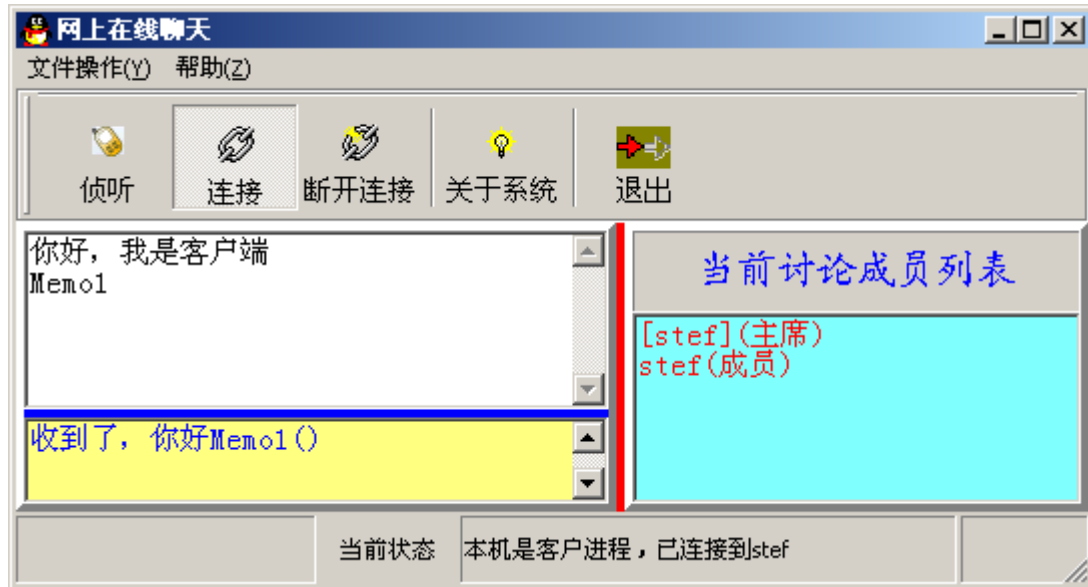
客户端发送消息到服务器端



服务器收到消息并回复



客户端收到服务器的回复消息



以上演示了该程序一对一时的对话过程，显示了该程序对一对一聊天的支持，该程序的界面友好，而且该程序也同时支持多对多的聊天过程，基本上实现了实验任务的需求。但该程序也有不太令人满意的地方，比如它没有很好的实现目前的好友的在线情况，也算是一些遗憾吧。

6 实验总结

通过本次实验我最大的体会是，完成一项工程是十分不容易的，不论工程项目的大或者小。完成一项工程决不是一蹴而就的事情，要经过较为细致的策划，以及不断的实践，并且在实践中不断地进行修改。而且我也体会到了什么叫做计划赶不上变化，往往计划很好的事情也许在实际操作当中会和自己的想象有很大出入。而且，计划当中的工程进度和实际的操作也有很大的出入。

但无论如何，通过这次开放实验的完成，我开发语言的水平有了很大的提高，对 Delphi6.0 更是有了很大的认识，我觉得它真的是一个很好的语言，很值得我再认真的系统的学习，同时我也觉得在这次实验中，通过再对程序不断的调试，运行，我发现问题，解决问题的能力有了很大的提高，对计算机网络的各个层次也有了更深层次的了解。

7 程序代码

```

unit netQQ;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ImgList, ComCtrls, ToolWin, ExtCtrls, ScktComp, Buttons,
  StdCtrls;

type
  Tw_cww = class(TForm)
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    Connect1:TMenuItem;

```

```

Listen1: TMenuItem;
StatusBar1: TStatusBar;
N1: TMenuItem;
Disconnect1: TMenuItem;
  ServerSocket: TServerSocket;
  ClientSocket: TClientSocket;
  ImageList1: TImageList;
  CoolBar1: TCoolBar;
  ToolBar1: TToolBar;
  ToolButton_listen: TToolButton;
  ToolButton_connect: TToolButton;
  ToolButton_disconnect: TToolButton;
  ToolButton7: TToolButton;
  ToolButton_about: TToolButton;
  ToolButton5: TToolButton;
  ToolButton_exit: TToolButton;
  Panel1: TPanel;
Memo1: TMemo;
Memo2: TMemo;
Splitter1: TSplitter;
Splitter2: TSplitter;
Panel2: TPanel;
Panel3: TPanel;
Memo3: TMemo;
Timer1: TTimer;
procedure ToolButton_listenClick(Sender: TObject);
procedure ToolButton_connectClick(Sender: TObject);
procedure ToolButton_disconnectClick(Sender: TObject);
procedure ToolButton_aboutClick(Sender: TObject);
procedure ToolButton_exitClick(Sender: TObject);
procedure StatusBar1DrawPanel(StatusBar: TStatusBar;
  Panel: TStatusPanel; const Rect: TRect);
procedure Memo1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ServerSocketAccept(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ServerSocketClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ServerSocketClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ClientSocketConnect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ClientSocketDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);

```

```

    procedure ClientSocketError(Sender: TObject; Socket: TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
    procedure ClientSocketRead(Sender: TObject; Socket: TCustomWinSocket);
    procedure Timer1Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
//    procedure SpeedButton1Click(Sender: TObject);

private
    { Private declarations }

public
    { Public declarations }
    IsServer:Boolean;
    Server:String;
    Membername:string;
    Procedure refresh_memo3 ;

end;

var
    w_cww: Tw_cww;

implementation
uses unit1;
{$R *.dfm}
    procedure Tw_cww.refresh_memo3;
    var
        i:integer;
    begin
        memo3.Lines.Clear;
        memo3.Lines.Add(membername+' [' +serversocket.Socket.LocalHost+' ](主席)'
        );

        for i:= 0 to serversocket.Socket.ActiveConnections-1 do begin
            memo3.Lines.Add(    serversocket.Socket.Connections[i].remoteHost+' ( 成
            员)');
        end;
    end;
    procedure Tw_cww.ToolButton_listenClick(Sender: TObject);
    begin
        //if IsServer then begin
            //listen1.Enabled:=true;
            //toolbutton_listen.Enabled :=true;

```



```

Listen1.Checked :=not Listen1.Checked;
toolbutton_listen.Down :=not toolbutton_listen.Down;
//ToolButton_connect.Enabled :=false;
//ToolButton_disconnect.Enabled := false;

//connect1.Enabled :=false;
//disconnect1.Enabled :=false;
//end else begin
//Listen1.enabled :=false;
//toolbutton_listen.enabled :=false;

//ToolButton_connect.Enabled :=true;
//ToolButton_disconnect.Enabled := true;

//connect1.Enabled := true;
//disconnect1.Enabled := true;
//end;

if Listen1.Checked then begin

    ClientSocket.Active := False;
    ServerSocket.Active := True;
    StatusBar1.Panels[2].Text := '本机是服务器，正在侦听...';
end else begin
    if ServerSocket.Active then
        ServerSocket.Active :=false;
        StatusBar1.Panels[2].Text := '本机是客户进程';
    end;
end;

procedure Tw_cww.ToolButton_connectClick(Sender: TObject);
begin
    if ClientSocket.Active then ClientSocket.Active := False;
    if InputQuery('Computer to connect to', 'Address Name:', Server) then
        if Length(Server) > 0 then
            with ClientSocket do
                begin
                    Host := Server;
                    Active := True;

                end;
            end;
        end;
end;

```

```

procedure Tw_cww.ToolButton_disconnectClick(Sender: TObject);
begin

    if sender is TmenuItem then begin
        disconnect1.Checked := not disconnect1.Checked;
        toolbutton_disconnect.Down := disconnect1.Checked
    end else
        disconnect1.Checked := toolbutton_disconnect.Down ;

    if disconnect1.Checked then begin

        try
            ClientSocket.Active := False;
            StatusBar1.Panels[2].Text := '本机是客户进程，断开连接!';
        except
            showmessage('断开连接失败!');
        end;
    end;
end;

procedure Tw_cww.ToolButton_aboutClick(Sender: TObject);
begin
    aboutbox.ShowModal ;
end;

procedure Tw_cww.ToolButton_exitClick(Sender: TObject);
begin
    ServerSocket.Close;
    ClientSocket.Close;
    Close;
end;

procedure Tw_cww.StatusBar1DrawPanel(StatusBar: TStatusBar;
    Panel: TStatusPanel; const Rect: TRect);
var
    oldcolor: Tcolor;
    oldstyle: Tbrushstyle;
begin
    if panel.Index = 3 then begin
        with StatusBar1.Canvas do begin
            oldstyle := brush.Style;
            brush.Style := bsClear;
            oldcolor := font.Color;

```

```

font.Color := clYellow;
font.name := '隶书';
font.Height := 20;
ImageList1.Draw(
    StatusBar1.Canvas,
    Rect.Left+5,
    Rect.Top+1,
    7
);
font.Color := clBlue;
TextOut(
    Rect.left + 40,
    Rect.Top+5,
    Panel.Text
);
font.Color := clYellow;
TextOut(
    Rect.left + 42,
    Rect.Top+7,
    Panel.Text
);
font.Color := oldcolor;
brush.Style := oldstyle;
end;
end;
end;

procedure Tw_cww.Memo1KeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
var
    i:integer;
begin
    if Key = VK_Return then

        if IsServer then

            for i := 0 to ServerSocket.Socket.ActiveConnections-1 do begin
                ServerSocket.Socket.Connections[i].SendText(
                    '[讨论]' + Memo1.Lines[Memo1.Lines.Count-1]
                    +' (' + membername + ')')
            )
        end
    else

```

```

        ClientSocket.Socket.SendText (
        '[讨论]' + Memo1.Lines[Memo1.Lines.Count-1]
        + '( ' + membername + ' )'
        );

end;

procedure Tw_cww.ServerSocketAccept (Sender: TObject;
Socket: TCustomWinSocket);
begin
    isservice:=true;
    refresh_memo3 ;
    statusBar1.panels[2].text:=socket.RemoteAddress+' 连接到该服务器';
end;

procedure Tw_cww.ServerSocketClientDisconnect (Sender: TObject;
Socket: TCustomWinSocket);
begin
    refresh_memo3 ;
end;

procedure Tw_cww.ServerSocketClientRead (Sender: TObject;
Socket: TCustomWinSocket);
var
    ss1,ss2: string;
    ll,i:integer;
begin
    ss1 := Socket.ReceiveText ;
    ll :=length(ss1);
    ss2 := copy(ss1,7,ll-6) ;
    Memo2.Lines.Add(ss2);
    for i := 0 to ServerSocket.Socket.ActiveConnections-1 do begin
        if (ServerSocket.Socket.Connections[i]<>socket) then
            ServerSocket.Socket.Connections[i].SendText (
            '[讨论]' +ss2
            )
        end;
    end;
end;

procedure Tw_cww.ClientSocketConnect (Sender: TObject;
Socket: TCustomWinSocket);
begin
    StatusBar1.Panels[2].Text := '本机是客户进程,已连接到'+Socket.RemoteHost;
    memo3.Lines.Add('[' +ClientSocket.Socket.RemoteHost+'](主席)');
end;

```

```

    memo3.Lines.Add(' [' +membername+' ](成员)') ;
end;

procedure Tw_cww.ClientSocketDisconnect(Sender: TObject;
    Socket: TCustomWinSocket);
begin
Memo1.Lines.Clear;
Memo2.Lines.Clear;
Memo3.Lines.Clear;
end;

procedure Tw_cww.ClientSocketError(Sender: TObject;
    Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
    var ErrorCode: Integer);
begin
showmessage(' 连接服务器失败! ');
ErrorCode := 0;
end;

procedure Tw_cww.ClientSocketRead(Sender: TObject;
    Socket: TCustomWinSocket);
var
    cmd, ss1, ss2, ss3: string ;
    ll, k: integer;
begin
    ss1 :=Socket.ReceiveText;
    ll :=length(ss1);
    cmd := copy(ss1, 2, 4) ;
    ss2 := copy(ss1, 7, ll-6);

    if cmd = ' 讨论' then
        Memo2.Lines.Add(ss2)
    else if cmd = ' 刷新' then
        Memo3.Lines.Clear
    else begin
        k :=Pos(' [信息]', ss2);
        while (k>0) do begin
            ss3 := copy( ss2, 1, k-1 ) ;
            ss2 := copy( ss2, k+6, length(ss2)-k-5) ;
            Memo3.Lines.Add( ss3 );
            k := Pos(' [信息]', ss2);
        end;
        Memo3.Lines.Add( ss2 );
    end;
end;

```

```

end;

procedure Tw_cww.Timer1Timer(Sender: TObject);
var
  i,k: integer;
begin
  if IsServer then begin
    refresh_memo3;

    for i := 0 to ServerSocket.Socket.ActiveConnections-1 do
      ServerSocket.Socket.Connections[i].SendText(' [刷新]开始');

    for k :=1 to Memo3.Lines.Count do
      for i := 0 to ServerSocket.Socket.ActiveConnections-1 do
        ServerSocket.Socket.Connections[i].SendText(
          '[信息]' + memo3.Lines[k-1]
        );
      end;
    end;
  end;
end;

procedure Tw_cww.FormCreate(Sender: TObject);

var
  ss: integer ;
  p1,p2:string ;
begin
  ss := paramcount ;
  if ss<3 then
    Application.Terminate
  else begin
    p1 := LowerCase(ParamStr(1)) ;
    p2 := ParamStr(2);
    membername := ParamStr(3) ;

    if p1 ='server' then
      IsServer := true
    else
      IsServer := false ;
      Server := p2 ;
  end;

  caption := caption+' ['+membername+']';
  ToolButton_listenClick(nil);

```

```

end;
end.

unit unit1;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, ExtCtrls, Shellapi;

type
    TAboutBox = class(TForm)
        Panel1: TPanel;
        Image1: TImage;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label5: TLabel;
        Button1: TButton;
        Label4: TLabel;
        procedure Label5Click(Sender: TObject);
        procedure Label5MouseMove(Sender: TObject; Shift: TShiftState; X,
            Y: Integer);
        //procedure Label4MouseMove(Sender: TObject; Shift: TShiftState; X,
            //Y: Integer);
        //procedure Label4Click(Sender: TObject);
        procedure Label4Db1Click(Sender: TObject);
        procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
            Y: Integer);
        procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    AboutBox: TAboutBox;

implementation

```

```
{$R *.dfm}
```

```
procedure TAboutBox.Label5Click(Sender: TObject);  
begin  
ShellExecute(  
handle, nil, pchar('mailto:somebody@someweb.com'),  
nil, nil, sw_shownormal  
);  
end;
```

```
procedure TAboutBox.Label4Db1Click(Sender: TObject);  
begin  
ShellExecute(handle, nil, pchar(Label4.hint), nil, nil, sw_shownormal);  
end;  
//procedure Label4MouseMove(Sender: TObject; Shift: TShiftState; X,  
//      Y: Integer);  
//begin  
//Label4.Font.Color :=clLime;  
//end ;
```

```
procedure TAboutBox.Label5MouseMove(Sender: TObject; Shift: TShiftState; X,  
      Y: Integer);  
begin  
Label5.Font.Color :=clred;  
end;  
procedure TAboutBox.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,  
      Y: Integer);  
begin  
Label5.Font.Color :=clNavy;  
Label4.Font.Color :=clred;  
end;
```

```
procedure TAboutBox.Button1Click(Sender: TObject);  
begin  
close;  
end;  
end.
```