

系统分析与设计课程设计实验报告

学号：

姓名：

指导老师：

完成时间：

# 实验五 界面设计与文件存取

## 一、实验目的

(1)练习TopClient和TopListener的用法。

(2)练习NetworkStream的用法。

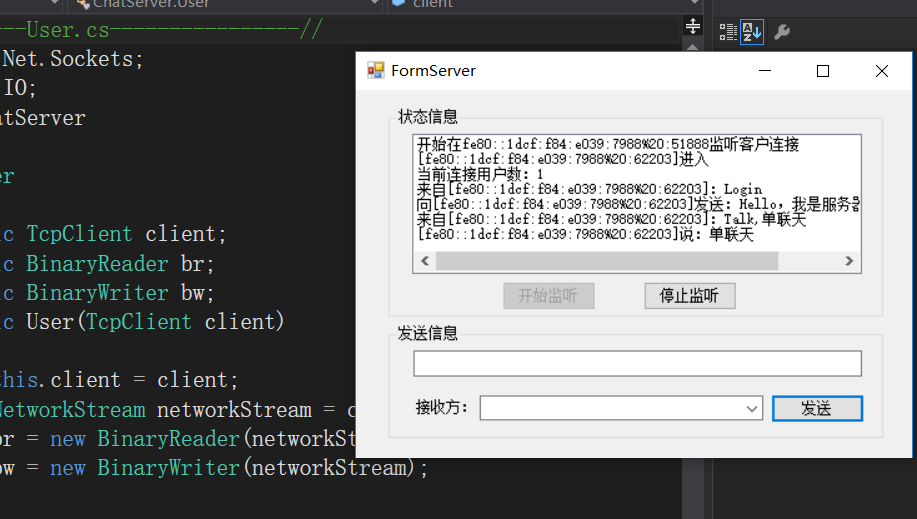
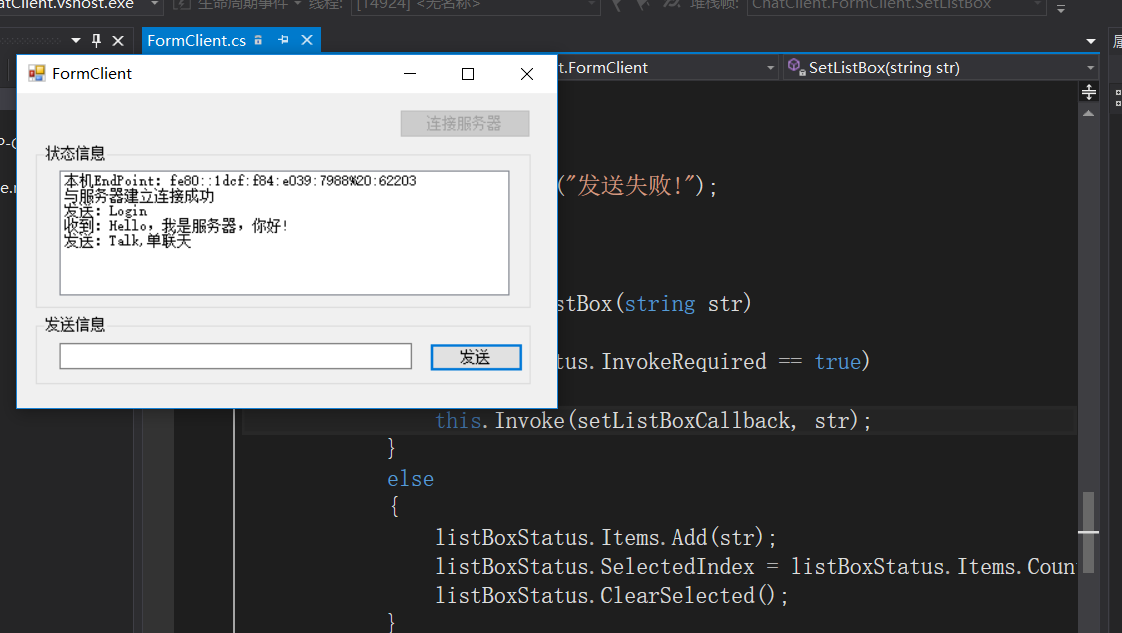
(3)练习BinaryRead和BinaryWriter的用法。

(4)练习线程的创建和使用方法。

(5)练习解决TCP协议消息边界问题的另一种方法。

## 二、实验结果

界面运行截图：



## 三、主要代码

## [Program.cs]

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace MessageAwake

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new FormMain());

}

}

}

## [FormMain.cs]

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Net;

using System.Net.Sockets;

using System.Threading;

namespace MessageAwake

{

public partial class FormMain : Form

{

private System.Windows.Forms.NotifyIcon myNotifyIcon;

private bool isExit = false;

private int formHeight;

private int port = 8001;

private UdpClient udpClient;

public FormMain()

{

InitializeComponent();

formHeight = 0;

this.Height = formHeight;

timer1.Enabled = true;

}

private void Form1\_Load(object sender, EventArgs e)

{

myNotifyIcon = new NotifyIcon(this.components);

myNotifyIcon.Icon = new Icon("demo.ico");

myNotifyIcon.Text = "网络呼叫提醒\n";

myNotifyIcon.ContextMenuStrip = this.contextMenuStrip1;

myNotifyIcon.Visible = true;

myNotifyIcon.DoubleClick += new EventHandler(myNotifyIcon\_DoubleClick);

IPAddress myIP = (IPAddress)Dns.GetHostAddresses(Dns.GetHostName()).GetValue(0);

textBoxRemoteIP.Text = myIP.ToString();

Thread myThread = new Thread(new ThreadStart(ReceiveData));

myThread.Start();

textBoxSendMessage.Focus();

}

void myNotifyIcon\_DoubleClick(object sender, EventArgs e)

{

ShowThisForm();

}

private void ShowThisForm()

{

this.Height = 0;

timer1.Enabled = true;

this.Show();

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (isExit == false)

{

e.Cancel = true;

this.Hide();

}

}

private void 结束程序ToolStripMenuItem\_Click(object sender, EventArgs e)

{

isExit = true;

udpClient.Close();

Application.Exit();

}

private void 呼叫对方ToolStripMenuItem\_Click(object sender, EventArgs e)

{

ShowThisForm();

}

private void ReceiveData()

{

udpClient = new UdpClient(port);

IPEndPoint remote = null;

while (true)

{

try

{

byte[] bytes = udpClient.Receive(ref remote);

string str = Encoding.UTF8.GetString(bytes, 0, bytes.Length);

MessageBox.Show(str, string.Format("收到来自[{0}]的呼叫", remote));

}

catch

{

break;

}

}

}

private void sendData()

{

UdpClient myUdpClient = new UdpClient();

IPAddress remoteIP;

if (IPAddress.TryParse(textBoxRemoteIP.Text, out remoteIP) == false)

{

MessageBox.Show("远程IP格式不正确");

return;

}

IPEndPoint iep = new IPEndPoint(remoteIP, port);

byte[] bytes = System.Text.Encoding.UTF8.GetBytes(textBoxSendMessage.Text);

try

{

myUdpClient.Send(bytes, bytes.Length, iep);

textBoxSendMessage.Clear();

myUdpClient.Close();

textBoxSendMessage.Focus();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "发送失败");

}

finally

{

myUdpClient.Close();

}

}

private void buttonSend\_Click(object sender, EventArgs e)

{

sendData();

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (formHeight < 235)

{

this.Height = formHeight;

formHeight += 5;

}

else

{

timer1.Enabled = false;

formHeight = 0;

}

}

}

}

## 四、问题及解答

(1). 写出你认为有必要解释的关键步骤或代码。

见上述代码。

(2). 写出实验中遇到的问题及解决方法。

运行程序会出现多个用户随机掉线的问题

解决方法：即TcpClient关闭时，也要手工关闭从其获得的NetworkStream（调用Close方法）。

## 五、实验小结

通过此次实验基本学习了线程的创建和适用，以及NetworkStream的用法，学会了用二进制方法来读取客户端适用POST传送方法所传递的数据