**实验8.4 完善聊天应用程序**

实验相关材料位于：<ftp://192.168.40.100/>下的“**--- 梁其洋**”目录下（可以通过搜索名字找到）

作业提交到：[ftp://192.168.40.14/梁其洋/网络编程\*\*班/实验\*](ftp://192.168.40.14/梁其洋/网络编程**班/实验*)

要求：1. 提交时**将本文档重新命名**，文档命名规则：学号+姓名+实验\*，例如，**517300614400 张三 实验\*.docx （一定要学号在前姓名在后，方便自动排序！）**

2. 你们没有FTP文件的删除权限，**如果需要提交新版本，在姓名后加序号后提交即可，例如，517300614400张三（1）实验\*.docx**

实验目的：

收到消息时，实现任务栏托盘里图标闪烁，单击图标即可打开信息浏览。

实验内容：

本此实验在实现网络数据传输功能时用到了TcpListener类 TcpClient类和 NetworkStream类，在控制托盘图标的闪动时用到Timer组件。用到的关键技术如下：

**（1）TCPListener类**

Tcplistener是对socket tcp协议封装的接口，更便于开发**，**用来从Tcp网络客户端侦听连接，常用方法及说明如下表



**（2）**TcpClient类

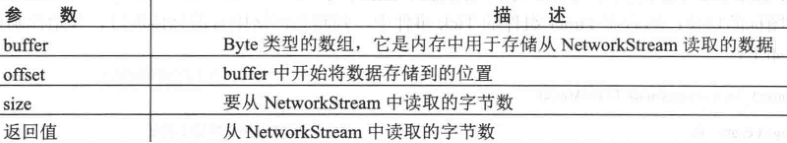
为Tcp网络服务提供客户端连接.

public TcpClient(string hostname,int port)　　//远程主机的**DNS**名、端口号

（3）NetworkStream类

提供在阻塞模式下通过Stream套接字发送和接受数据的方法。其Read方法用于从NetworkStream类读取数据

**public override int Read(byte[] buffer,int offset,int size)**



（4）托盘图标闪动的原理

托盘图标闪动主要是在Timer组件的Tick事件中实现的，在该事件中，如果网络中有数据传输，则通过notifyIcon1控件的Icon属性更改托盘图标。需要注意的是，首先准备两个图标文件，其中一个图标文件是透明的，这样两个图标文件相互切换时，就出现图标闪动的效果。

实验步骤参考：

[**https://www.cnblogs.com/wifi/articles/2471712.html**](https://www.cnblogs.com/wifi/articles/2471712.html) **--------------------------------------url1**

[**https://blog.csdn.net/CRSUN/article/details/246552**](https://blog.csdn.net/CRSUN/article/details/246552) **--------------------------------------url2**

**注意：**

**①需要自己找两张图标文件，其中一张透明（或者颜色、形状不同）。**

**免费图标：<http://www.icosky.com/icon/ichat-2-1-icon/>**

**②本次实验采用url1提供的完整代码，加载图标的方式为：**

**notifyIcon1.Icon = Properties.Resources.\_1;**

**第二种图标加载方式为**url2中采用的（更常用）**：**

**Icon m\_Icon1 = new Icon("Icon1.ico");//导入图标文件**

③**运行程序前在窗体设计界面设置一下**notifyIcon1控件的Icon属性。

④采用第二个URL里引用图标文件的方式时需要在图标文件的属性里设置：每次更新时复制。

程序运行截图：

namespace DynamicTaskStock

{

partial class Frm\_Main

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

#endregion

/// <summary>

/// 设计器支持所需的方法 - 不要

/// 使用代码编辑器修改此方法的内容。

/// </summary>

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container();

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Frm\_Main));

this.notifyIcon1 = new System.Windows.Forms.NotifyIcon(this.components);

this.timer1 = new System.Windows.Forms.Timer(this.components);

this.txtAdd = new System.Windows.Forms.TextBox();

this.button1 = new System.Windows.Forms.Button();

this.label1 = new System.Windows.Forms.Label();

this.groupBox1 = new System.Windows.Forms.GroupBox();

this.button2 = new System.Windows.Forms.Button();

this.groupBox1.SuspendLayout();

this.SuspendLayout();

//

// notifyIcon1

//

this.notifyIcon1.Icon = ((System.Drawing.Icon)(resources.GetObject("notifyIcon1.Icon")));

this.notifyIcon1.Visible = true;

//

// timer1

//

this.timer1.Enabled = true;

this.timer1.Interval = 500;

this.timer1.Tick += new System.EventHandler(this.timer1\_Tick);

//

// txtAdd

//

this.txtAdd.Location = new System.Drawing.Point(137, 25);

this.txtAdd.Name = "txtAdd";

this.txtAdd.Size = new System.Drawing.Size(193, 21);

this.txtAdd.TabIndex = 0;

//

// button1

//

this.button1.Location = new System.Drawing.Point(174, 52);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(75, 23);

this.button1.TabIndex = 1;

this.button1.Text = "发送消息";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(18, 28);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(113, 12);

this.label1.TabIndex = 2;

this.label1.Text = "输入对方主机地址：";

//

// groupBox1

//

this.groupBox1.Controls.Add(this.button2);

this.groupBox1.Controls.Add(this.button1);

this.groupBox1.Controls.Add(this.label1);

this.groupBox1.Controls.Add(this.txtAdd);

this.groupBox1.Location = new System.Drawing.Point(8, 7);

this.groupBox1.Name = "groupBox1";

this.groupBox1.Size = new System.Drawing.Size(347, 88);

this.groupBox1.TabIndex = 3;

this.groupBox1.TabStop = false;

this.groupBox1.Text = "发送消息";

//

// button2

//

this.button2.Location = new System.Drawing.Point(255, 52);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(75, 23);

this.button2.TabIndex = 3;

this.button2.Text = "停止闪动";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// Frm\_Main

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(363, 100);

this.Controls.Add(this.groupBox1);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;

this.MaximizeBox = false;

this.MinimizeBox = false;

this.Name = "Frm\_Main";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

this.Text = "实现动态系统托盘图标";

this.Load += new System.EventHandler(this.Frm\_Main\_Load);

this.FormClosed += new System.Windows.Forms.FormClosedEventHandler(this.Frm\_Main\_FormClosed);

this.groupBox1.ResumeLayout(false);

this.groupBox1.PerformLayout();

this.ResumeLayout(false);

}

private System.Windows.Forms.NotifyIcon notifyIcon1;

private System.Windows.Forms.Timer timer1;

private System.Windows.Forms.TextBox txtAdd;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.GroupBox groupBox1;

private System.Windows.Forms.Button button2;

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Net.Sockets;

using System.Threading;

using System.IO;

using System.Net;

namespace DynamicTaskStock

{

public partial class Frm\_Main : Form

{

public Frm\_Main()

{

InitializeComponent();

}

private Thread td; //创建一个线程

private TcpListener tcpListener; //实例化侦听对象

string message = ""; //记录传输的内容

private void StartListen()

{

tcpListener = new TcpListener(888); //创建TcpListener实例

tcpListener.Start(); //开始监听

while (true)

{

TcpClient tclient = tcpListener.AcceptTcpClient(); //接受连接请求

NetworkStream nstream = tclient.GetStream(); //获取数据流

byte[] mbyte = new byte[1024]; //建立缓存

int i = nstream.Read(mbyte, 0, mbyte.Length); //将数据流写入缓存

message = Encoding.Default.GetString(mbyte, 0, i); //获取传输的内容

}

}

private void Frm\_Main\_Load(object sender, EventArgs e)

{

td = new Thread(new ThreadStart(this.StartListen)); //通过线程调用StartListen方法

td.Start(); //开始运行线程

}

private void Frm\_Main\_FormClosed(object sender, FormClosedEventArgs e)

{

if (this.tcpListener != null)

{

tcpListener.Stop(); //停止侦听对象

}

if (td != null)

{

if (td.ThreadState == ThreadState.Running) //判断线程是否运行

{

td.Abort(); //终止线程

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

IPAddress[] ip = Dns.GetHostAddresses(Dns.GetHostName()); //获取本机地址

string message = "你好兄弟"; //传输的内容

TcpClient client = new TcpClient(txtAdd.Text, 888); //创建TcpClient实例

NetworkStream netstream = client.GetStream(); //创建NetworkStream实例

StreamWriter wstream = new StreamWriter(netstream, Encoding.Default); //创建StreamWriter实例

wstream.Write(message); //将信息写入流

wstream.Flush();

wstream.Close(); //关闭流

client.Close(); //关闭TcpClient对象

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

bool k = true; //一个标记，用于控制图标闪动

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

{

if (message.Length > 0) //如果网络中传输了数据

{

if (k) //k为true时

{

notifyIcon1.Icon = Properties.Resources.\_1; //托盘图标为1

k = false; //设k为false

}

else //k为false时

{

notifyIcon1.Icon = Properties.Resources.\_2; //图盘图标为2，透明的图标

k = true; //k为true

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

message = ""; //清空传输内容

notifyIcon1.Icon = Properties.Resources.\_1; //初始化显示的图标

}

}

}

