**实验8.2 利用Socket 传送图片**

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！）**

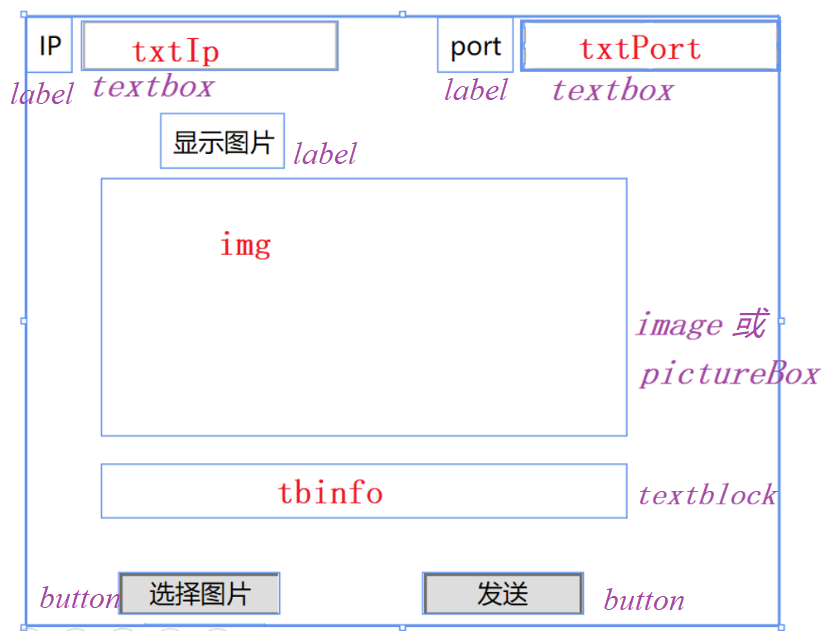
2. 你们没有FTP文件的删除权限，**如果需要提交新版本，在姓名后加序号后提交即可，例如，517300614400张三（1）实验\*.docx**

实验目的：练习Socket传文件

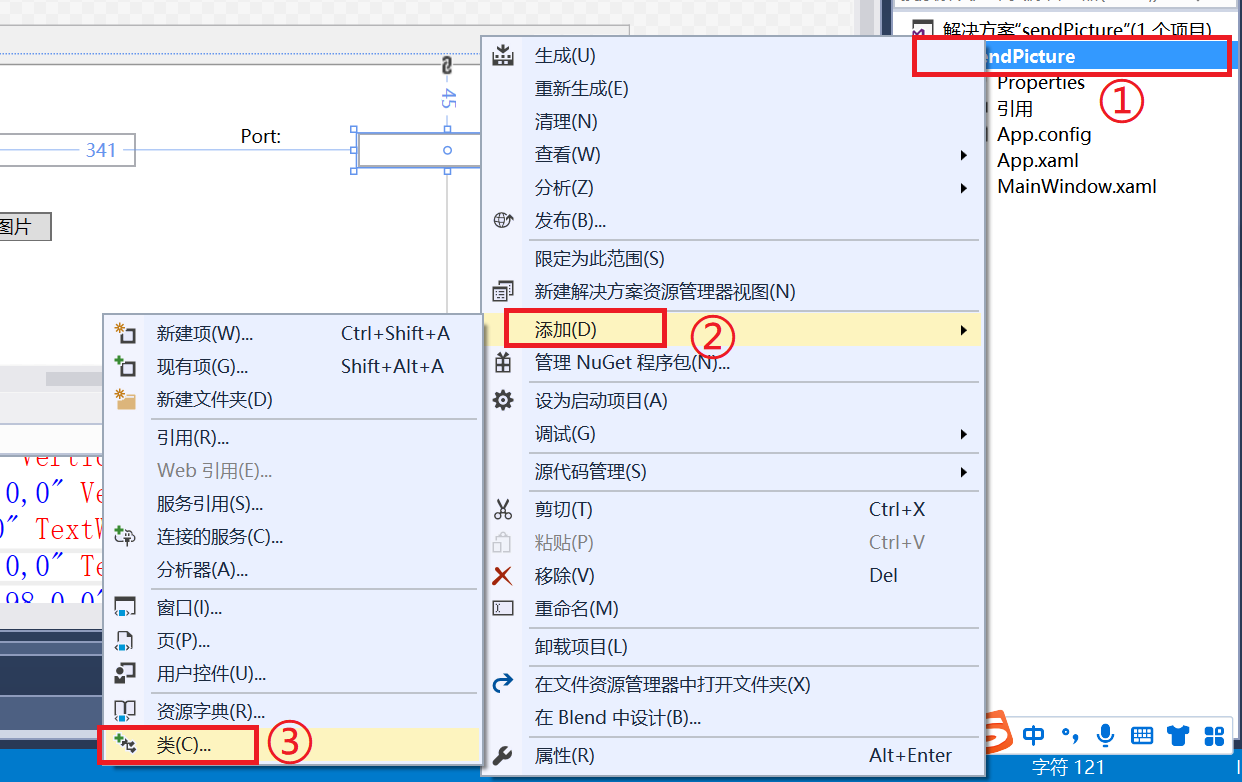
实验步骤参考：

一、新建一个WPF工程，界面上的主要控件的name如下图红字所示：

显示图片的控件根据VS的版本不同可以是image或者pictureBox。



二、将获取IP和端口的方法封装到工具类AddressHelper



public static class AddressHelper

{

/// <summary>

/// 获取本机IPv4地址的集合

/// </summary>

/// <returns></returns>

public static IPAddress[] GetLocalhostIPv4Addresses()

{

String LocalhostName = Dns.GetHostName();

IPHostEntry host = Dns.GetHostEntry(LocalhostName);

List<IPAddress> addresses = new List<IPAddress>();

foreach (IPAddress ip in host.AddressList)

{

if (ip.AddressFamily == AddressFamily.InterNetwork)

addresses.Add(ip);

}

return addresses.ToArray();

}

/// <summary>

/// 以交互方式生成有效的远程主机访问终结点,适用于控制台程序

/// </summary>

/// <returns></returns>

public static IPEndPoint GetRemoteMachineIPEndPoint()

{

IPEndPoint iep = null;

try

{

Console.Write("请输入远程主机的IP地址：");

IPAddress address = IPAddress.Parse(Console.ReadLine());

Console.Write("请输入远程主机打开的端口号：");

int port = Convert.ToInt32(Console.ReadLine());

if (port > 65535 || port < 1024)

throw new Exception("端口号应该为[1024,65535]范围内的整数");

iep = new IPEndPoint(address, port);

}

catch (ArgumentNullException)

{

Console.WriteLine("输入的数据有误！");

}

catch (FormatException)

{

Console.WriteLine("输入的数据有误！");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

return iep;

}

/// <summary>

/// 获取本机当前可用的端口号，此方法是线程安全的

/// </summary>

/// <returns></returns>

public static int GetOneAvailablePortInLocalhost()

{

Mutex mtx = new Mutex(false, "MyNetworkLibrary.AddressHelper.GetOneAvailablePort");

try

{

mtx.WaitOne();

IPEndPoint ep = new IPEndPoint(IPAddress.Any, 0);

using (Socket tempSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp))

{

tempSocket.Bind(ep);

IPEndPoint ipep = tempSocket.LocalEndPoint as IPEndPoint;

return ipep.Port;

}

}

finally

{

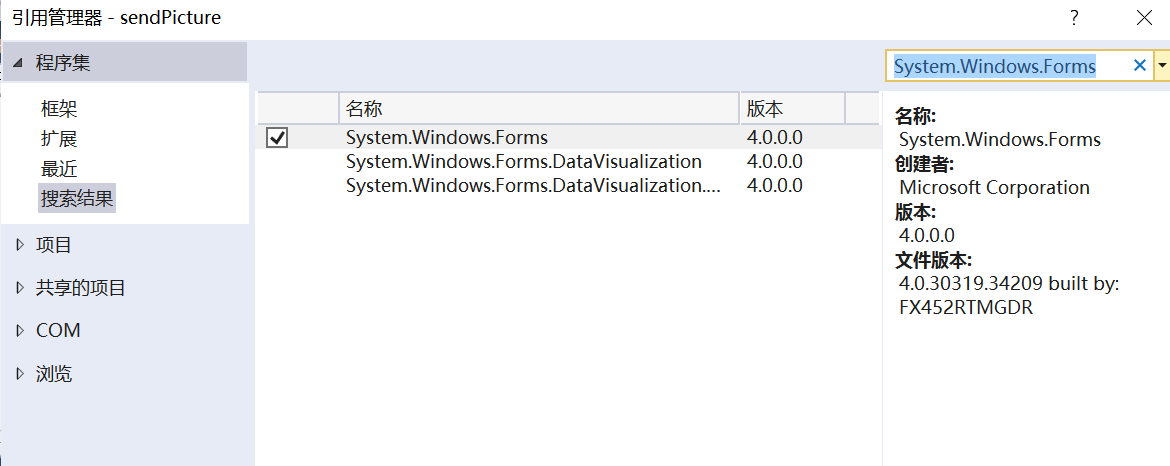
mtx.ReleaseMutex();

}

}

}

由于这句报错，System.Windows.Forms.OpenFileDialog ofd = new System.Windows.Forms.OpenFileDialog();按下图添加对System.Windows.Forms的引用。



三、将发送和接收二进制的socket封装到一个名为SocketHelper的静态工具类中。提示：此类中的几处报错都可以通过引用VS自动提示的内容解决。

public static class SocketHelper

{/// <summary>

/// 接收变长的数据，要求其打头的4个字节代表有效数据的长度

/// </summary>

/// <param name="s"></param>

/// <returns></returns>

public static byte[] ReceiveVarData(Socket s)

{

if (s == null)

throw new ArgumentNullException("s");

int total = 0; //已接收的字节数

int recv;

//接收4个字节，得到“消息长度”

byte[] datasize = new byte[4];

recv = s.Receive(datasize, 0, 4, 0);

int size = BitConverter.ToInt32(datasize, 0);

//按消息长度接收数据

int dataleft = size;

byte[] data = new byte[size];

while (total < size)

{

recv = s.Receive(data, total, dataleft, 0);

if (recv == 0)

{

break;

}

total += recv;

dataleft -= recv;

}

return data;

}

/// <summary>

/// 发送变长的数据，将数据长度附加于数据开头

/// </summary>

/// <param name="s"></param>

/// <param name="data"></param>

/// <returns></returns>

public static int SendVarData(Socket s, byte[] data)

{

int total = 0;

int size = data.Length; //要发送的消息长度

int dataleft = size; //剩余的消息

int sent;

//将消息长度（int类型）的，转为字节数组

byte[] datasize = BitConverter.GetBytes(size);

//将消息长度发送出去

sent = s.Send(datasize);

//发送消息剩余的部分

while (total < size)

{

sent = s.Send(data, total, dataleft, SocketFlags.None);

total += sent;

dataleft -= sent;

}

return total;

}

}

四、完善主窗体类：

public partial class MainWindow : Window

{

Socket server = null;

string picPath = "";

public MainWindow()

{

InitializeComponent();

Init();

}

private void Init()

{

//初始化

var ip = AddressHelper.GetLocalhostIPv4Addresses().First();

var port = AddressHelper.GetOneAvailablePortInLocalhost();

txtIp.Text = ip.ToString();

txtPort.Text = port.ToString();

IPEndPoint iep = new IPEndPoint(ip, port);

server = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

server.Bind(iep);

server.Listen(1);

//启动线程开始接收数据

DoWork();

tbinfo.Text = iep.ToString() + "已启动监听";

}

private Task DoWork()

{

return Task.Run(() =>

{

//创建委托，更新图片控件

Action<byte[]> action = buf =>

{

img.Source = ByteToBitmapImage(buf);

};

while (true)

{

var client = server.Accept();

var data = SocketHelper.ReceiveVarData(client);

this.Dispatcher.BeginInvoke(action, data);

client.Shutdown(SocketShutdown.Both);

client.Close();

}

});

}

//二进制转图片

public BitmapImage ByteToBitmapImage(byte[] buf)

{

BitmapImage bmp = new BitmapImage();

MemoryStream ms = new MemoryStream(buf);

ms.Position = 0;

bmp.BeginInit();

bmp.StreamSource = ms;

bmp.EndInit();

return bmp;

}

//选择图片

private void Button\_Click(object sender, RoutedEventArgs e)

{

System.Windows.Forms.OpenFileDialog ofd = new System.Windows.Forms.OpenFileDialog();

if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

picPath = ofd.FileName;

img.Source = new BitmapImage(new Uri(picPath));

}

}

//取远程终端的IPEndPoint

private IPEndPoint getRemIEP()

{

return new IPEndPoint(IPAddress.Parse(txtIp.Text), int.Parse(txtPort.Text));

}

//发送

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

var newSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

using (newSocket)

{

IPEndPoint remIep = getRemIEP();

newSocket.Connect(remIep);

var buf = File.ReadAllBytes(picPath);

SocketHelper.SendVarData(newSocket, buf);

}

}

}

在xmal文件中为选择图片按钮添加click事件处理函数：Click="Button\_Click\_1"，

为发送按钮添加click事件处理函数：Click="Button\_Click"

五、启动过程：

（1）第一次正常启动一个窗体，第二次在工程名上右键——调试——启动新实例。

（2）**将两个窗体上自动分配的端口对调！**

（3）在连接**无线网**的笔记本上启动窗体后如果不能正常传图片，要将两个窗体上自动获取的IP都改为127.0.0.1

程序运行截图：

