**实验8.1 C#聊天程序**

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：

掌握利用C#套接字实现聊天程序。基本过程：

1.根据服务器IP和端口号建立EndPoint对象；

2.建立Socket对象；

3.利用Socket对象的Bind方法绑定EndPoint对象；

4.利用Socket的Listen方法监听；

5.与客户端建立连接并用Socket的Accept创建新的Socket对象并用新对象的Receive和Send方法接收消息和返回消息

6.最后关闭Socket连接；

控制台聊天程序实验步骤：

（1）新建**控制台程序**ServerSocket，在main方法中添加

string host = "127.0.0.1";//定义了一个服务器主机号

int port = 1000;//端口号

IPAddress ip = IPAddress.Parse(host);//获取服务器Ip

IPEndPoint endPoint = new IPEndPoint(ip, port);//定义EndPoint对象 文本

描述已自动生成

Console.ReadLine();

（2）在同一个解决方案下添加**控制台程序**clientSocket，在main方法中添加

string host = "127.0.0.1";//定义了一个服务器主机号

int port = 1000;//端口号 文本

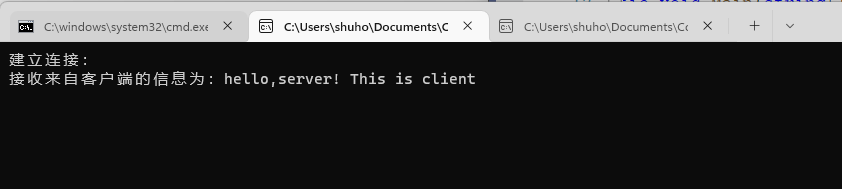
描述已自动生成

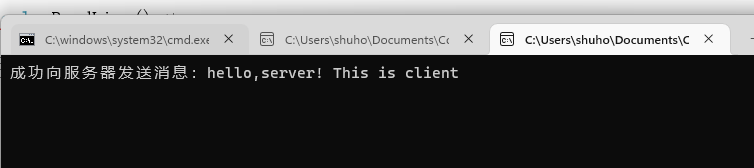
Console.ReadLine();

（3）点击启动按钮，默认启动ServerSocket，在工程名ClientSocket上右键，选择“调试”——“启动新实例”，启动ClientSocket。

提示：退出客户端ClientSocket的控制台窗口后，下次启动ClientSocket之前要解决方案上右键——清理解决方案。

程序运行截图：





二、桌面版聊天程序实验步骤：

（1）服务器端：创建一个**WinForm应用程序**chatServer，在界面上添加如下控件，并按下图修改除IP标签和端口标签之外的各控件的name属性或Text属性。



（2）在 public partial class Form1 : Form中添加全局变量：

List<Socket> ClientProxSocketList = new List<Socket>();

（3）实现btnStart的Click事件

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

{

图形用户界面, 文本, 应用程序

描述已自动生成

}

（4）添加辅助方法：

//接受客户端的连接请求

public void **AcceptClientConnect**(object socket)

{

图形用户界面, 文本

描述已自动生成

}

//接受客户端的消息

public void **ReceiveData**(object socket) {

var proxSocket = socket as Socket;

byte[] data = new byte[1024 \* 1024];

while (true) {

int len= 0;

try

{

len = proxSocket.Receive(data, 0, data.Length, SocketFlags.None);

}

catch(Exception ex) {

AppendTextToTxtLog(string.Format("客户端：{0} 非正常退出", proxSocket.RemoteEndPoint.ToString()));

ClientProxSocketList.Remove(proxSocket);

StopConnect(proxSocket);

return;

}

if (len <= 0) {

//客户端正常退出

AppendTextToTxtLog(string.Format("客户端：{0} 正常退出", proxSocket.RemoteEndPoint.ToString()));

ClientProxSocketList.Remove(proxSocket);

StopConnect(proxSocket);

return;//让方法结束，终结当前接收客户端数据的异步线程

}

//把接收到的数据放到文本框上去

string str= Encoding.Default.GetString(data,0,len);

AppendTextToTxtLog(string.Format("接收到客户端：{0}的消息是：{1}",proxSocket.RemoteEndPoint.ToString(), str));

}

}

//客户端正常或者非常退出时断开连接

private void **StopConnect**(Socket proxSocket)

{

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

}

//往日志的文本框上追加数据

public void **AppendTextToTxtLog**(string txt) {

if (txtLog.InvokeRequired)

{

txtLog.BeginInvoke(new Action<string>(s => {

this.txtLog.Text = string.Format("{0}\r\n{1}", txtLog.Text, s);

}), txt);

}

else {

this.txtLog.Text = string.Format("{0}\r\n{1}", txtLog.Text, txt);

}

}

//发送消息按钮的Click事件

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

{

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

}

（5）客户端：在同一解决方案下添加一个名为ChatClient 的WinForm工程，删除默认的WinFrm窗体，添加一个名为Client的窗体，将服务器端的界面上的所有控件都复制的到Client窗体界面上，按下图修改窗体的text属性、“连接”按钮的name属性和Text属性，其他控件的属性不变



（6）在public partial class client : Form类中创建一个全局变量：

public Socket ClientSocket;

（7）实现btnConnect按钮的Click事件

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

{

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

}

（8）实现btnSendMsg按钮的Click事件

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

（9）添加辅助方法：

public void **ReceiveData**(object socket)

{

var proxSocket = socket as Socket;

byte[] data = new byte[1024 \* 1024];

while (true)

{

int len = 0;

try

{

len = proxSocket.Receive(data, 0, data.Length, SocketFlags.None);

}

catch

{

AppendTextToTxtLog(string.Format("服务器端：{0} 非正常退出", proxSocket.RemoteEndPoint.ToString()));

StopConnect();//关闭连接

return;

}

if (len <= 0)

{

//客户端正常退出

AppendTextToTxtLog(string.Format("服务器端：{0} 正常退出", proxSocket.RemoteEndPoint.ToString()));

StopConnect();//关闭连接

return;//让方法结束，终结当前接收客户端数据的异步线程

}

//把接收到的数据放到文本框上去

string str = Encoding.Default.GetString(data, 0, len);

AppendTextToTxtLog(string.Format("接收到客户端：{0}的消息是：{1}", proxSocket.RemoteEndPoint.ToString(), str));

}

}

private void **StopConnect**() {

try

{

if (ClientSocket.Connected)

{

ClientSocket.Shutdown(SocketShutdown.Both);

ClientSocket.Close(100);

}

}

catch (Exception ex)

{

}

}

//往日志文本框上追加数据

public void **AppendTextToTxtLog**(string txt)

{

if (txtLog.InvokeRequired)

{

txtLog.BeginInvoke(new Action<string>(s => {

this.txtLog.Text = string.Format("{0}\r\n{1}", txtLog.Text , s);

}), txt);

}

else

{

this.txtLog.Text = string.Format("{0}\r\n{1}", txt, txtLog.Text);

}

}

程序运行截图：

提示：点击启动按钮，默认启动chatServer，在工程名chatClient上右键，选择“调试”——“启动新实例”，启动chatClient。分别在两个窗体中向对方发送消息。



