**TCP/IP编程综合实验**

**聊**

**天**

**类**

**程**

**序**

**班级：计算机科学与技术（卓越班）**

**组长：卫小雪（41409310123）**

**组员：任书甲（41409310117）**

1. **设计方案**

* **程序说明：**

实现聊天类程序的设计，TCP/IP部分知识为主，编程语言为辅，实现客户端—服务器端的连接，

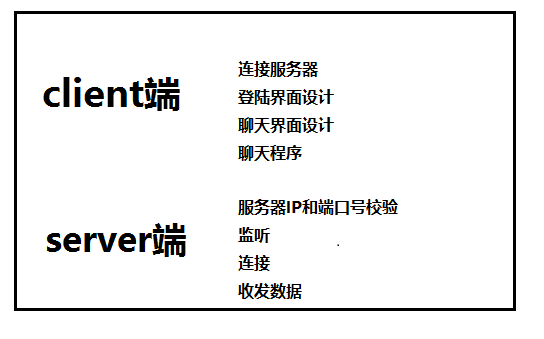
目的是实现聊天功能。 通过C#提供的Socket类来连接客户机和服务器端，并使客户机和服务器之间可以相互通信。

* **系统概述**

实现局域网内的聊天功能，流程为：

**运行服务器->服务器创建成功->让服务器对某一个设定的端口处于监听状态->客服端请求连接->服务器接受请求->建立连接**

* **设计框架思路：**

****

主要分为两部分：Client和Server

重点实现：运行在不同机器上的进程彼此通过套接字发送报文来进行通信，从而实现聊天功能。

**Server端的功能：**

1. 在特定端口进行监听，等待客户端连接
2. 用户可以自己配置监听端口（本程序设置为8888）
3. 向已经连接到服务器端的用户发送系统消息
4. 统计在线人数
5. 当在线进行交互时，服务器可以看到所有的交互。
6. 当关闭服务器时，断开所有用户连接

**Client端功能**

1. 连接服务器端
2. 用户可以自己配置要连接到服务器端的IP地址和端口号
3. 用户可以设置连接后现实的用户名
4. 当服务器开启，客户端可以随时登陆和注销
5. 用户可以发送消息进行聊天
6. 用户可以选择聊天模式：私聊和广播。
7. 模式为私聊时可以选择在线的人进行私聊。
8. 每两个人私聊都不会影响其他人。
9. 用户可以给别人发送一个屏幕抖动。
10. 用户可以保存自己的聊天记录，也可以删除聊天记录。
11. 可以把页面缩小到托盘。

* **系统运行**

语言：基于C#

工具：Vs运行环境

操作系统：安装了windows系统的计算机

* **需求功能分析**

运用socket编程以及C#编程，实现聊天系统。

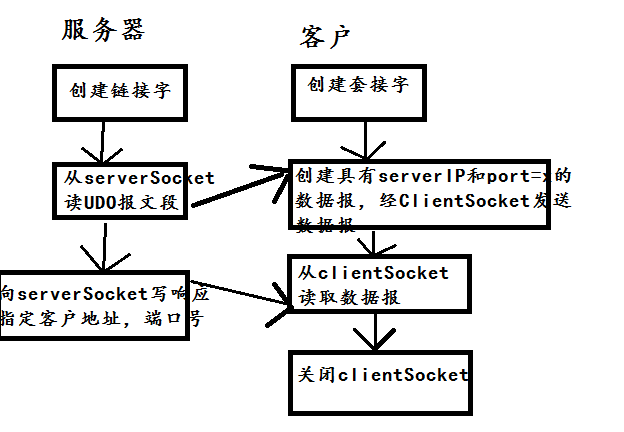
C#编程：此部分不需要重点讨论，框架为Client和Sever两大部分组成。

需要实现：启动服务器，登陆进入，寻找聊天对象进行聊天（发送消息与接收消息），关闭服务器。

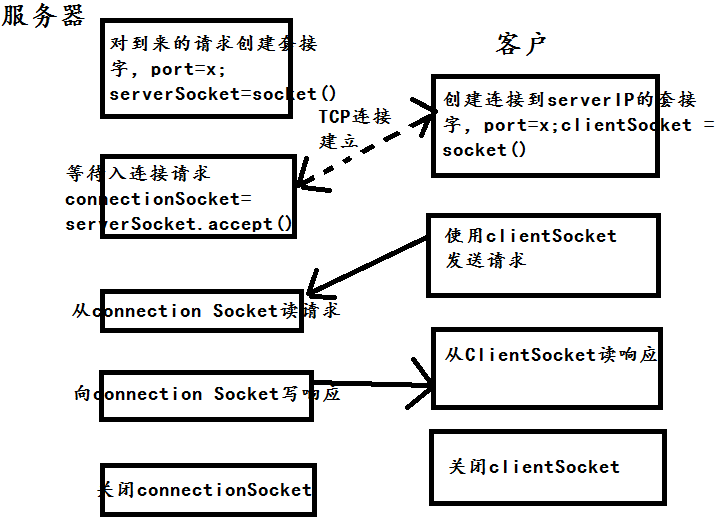
**Socket编程：**

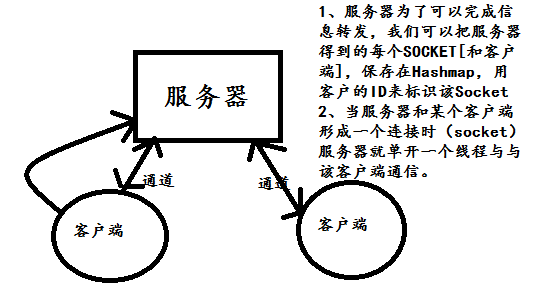
TCP套接字编程主要分为TCP和UDP两种，TCP是面向连接的，并且为两个端系统之间的数据流提供可靠的字节流通道。UDP是无连接的，从一个端服务向另一个端系统发送独立的数据分组，不对交付提供任何保证。

**使用UDP的客户——服务器应用程序**

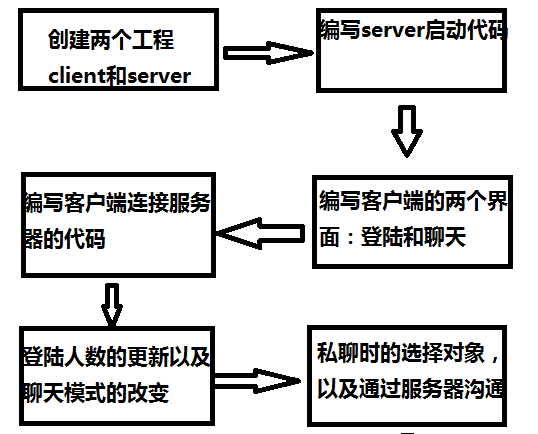


**使用TCP的客户——服务器应用程序**





**界面具体实现设计流程**

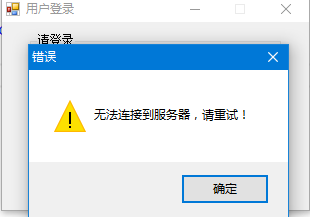




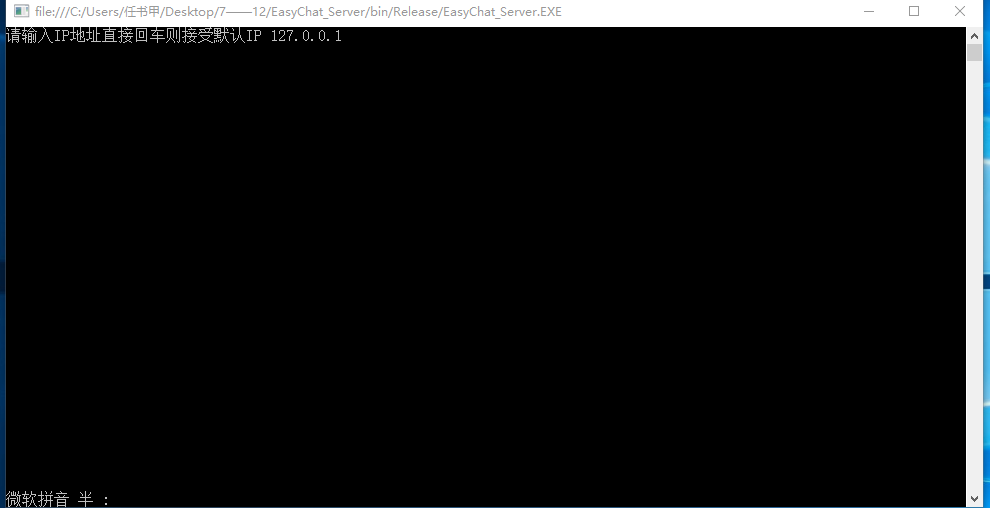
1. **运行界面截图**

启动服务器界面：

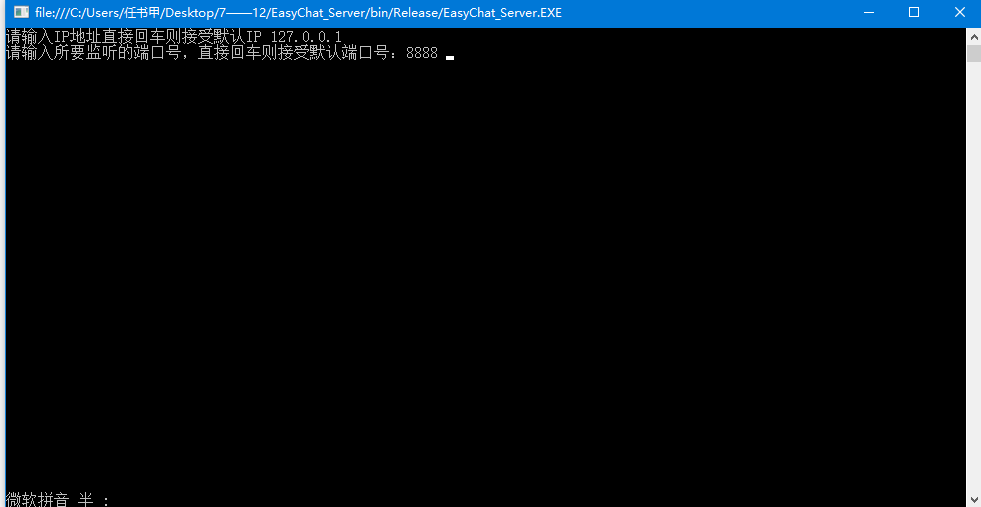
若启动无服务器：



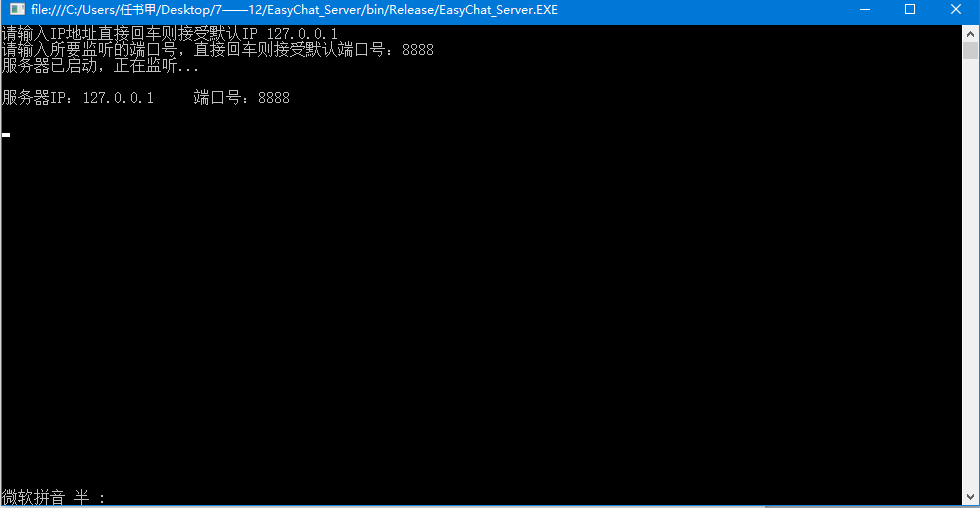
输入IP地址：默认值为127.0.0.1



输入端口号，默认值为8888

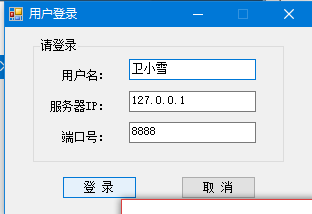


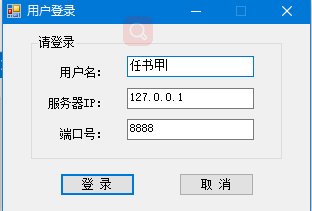
连接成功：



启动客户端界面：

客户端登陆界面：

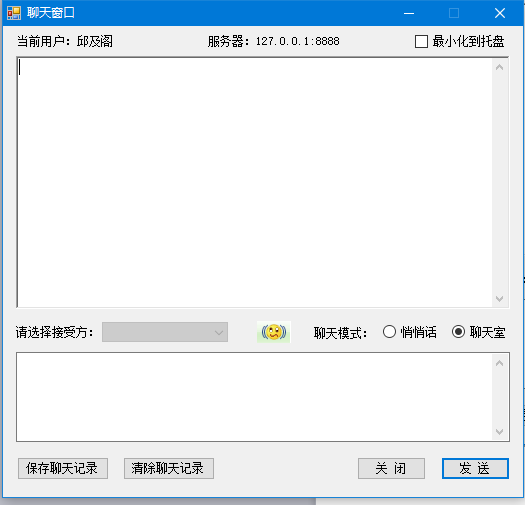
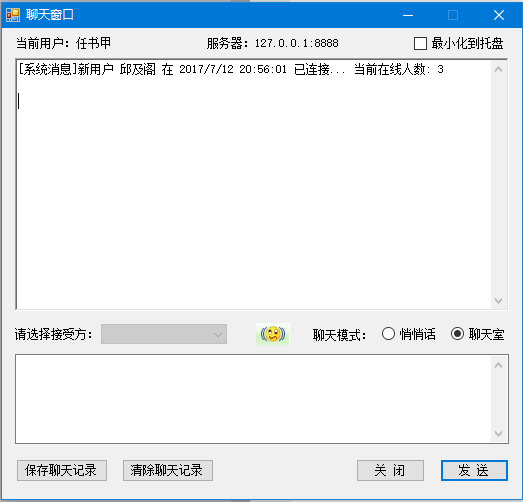




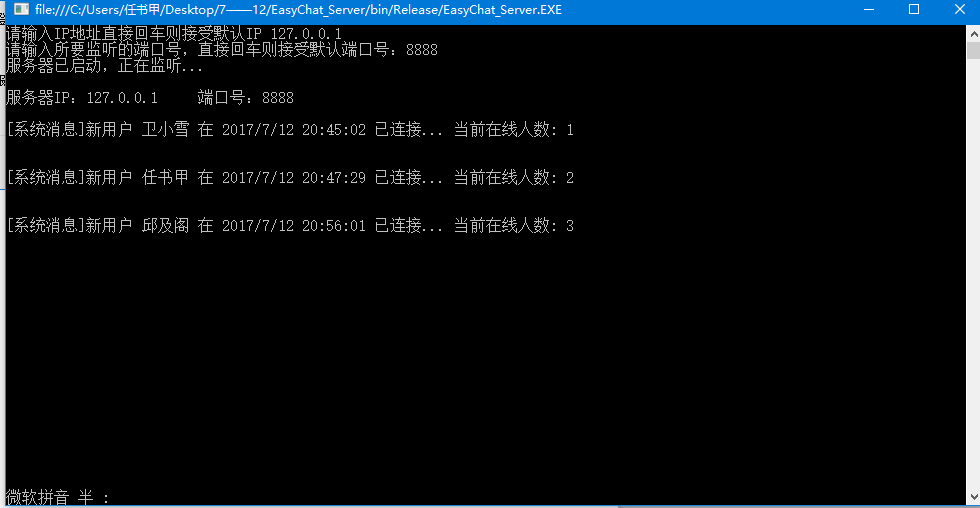


聊天窗口：



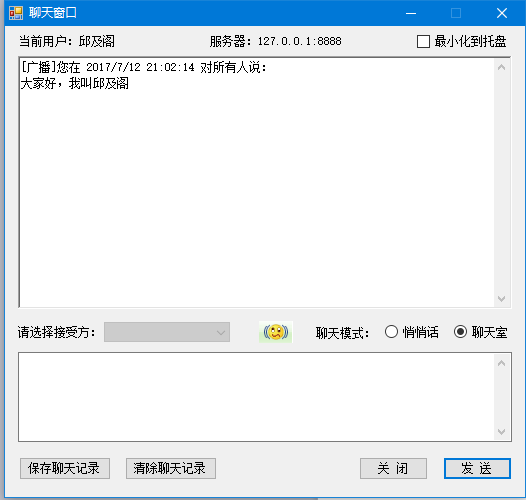


服务器界面：

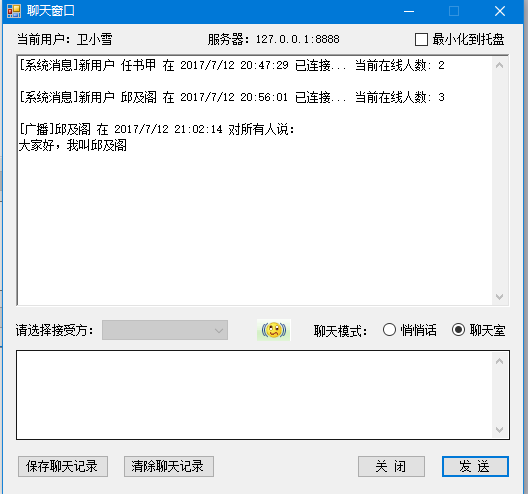


聊天过程：

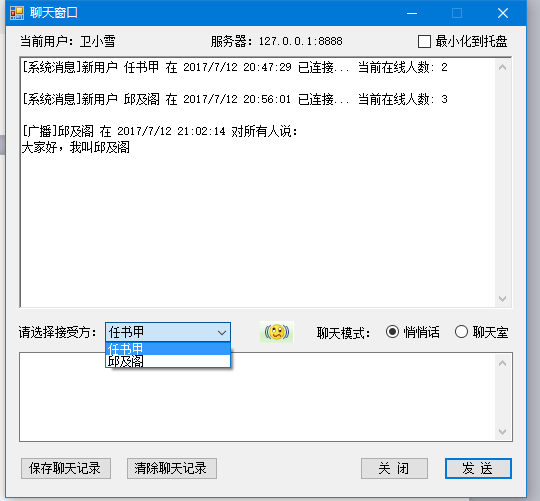
群内聊天（聊天室）：

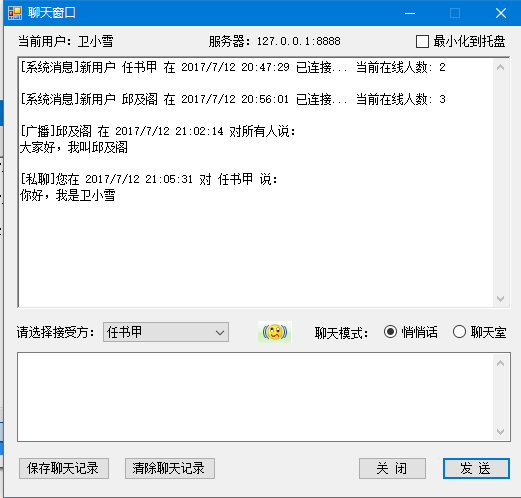


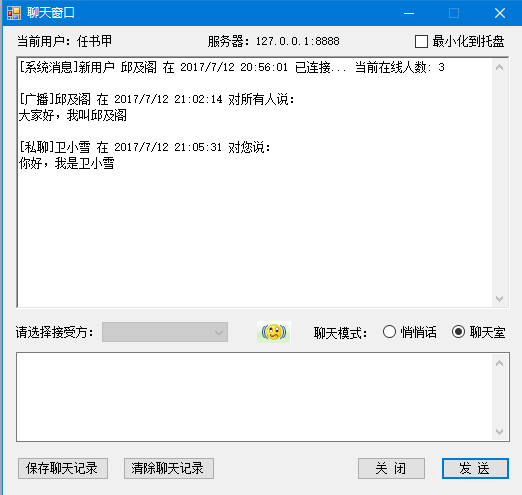




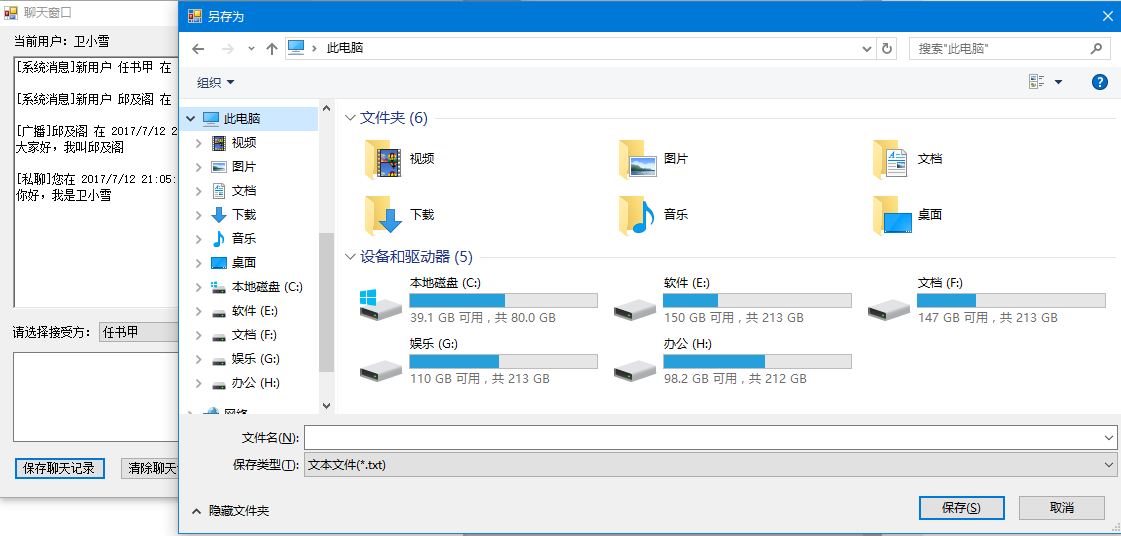
单人聊天（悄悄话）：



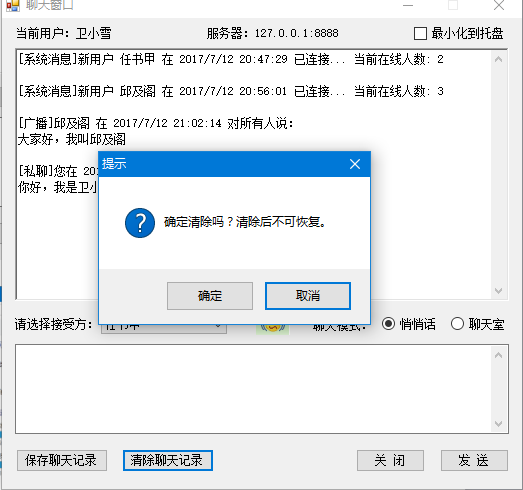




保存聊天记录：



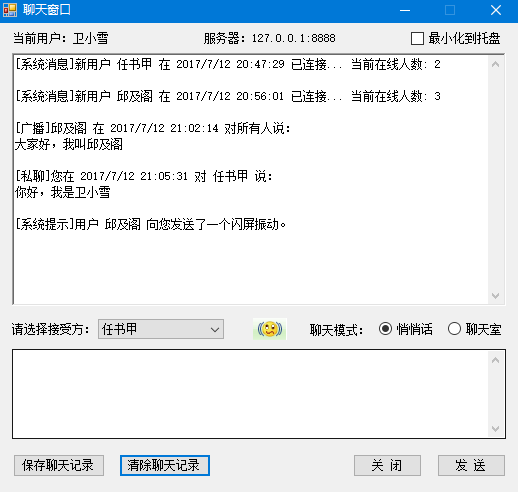
清除聊天记录：

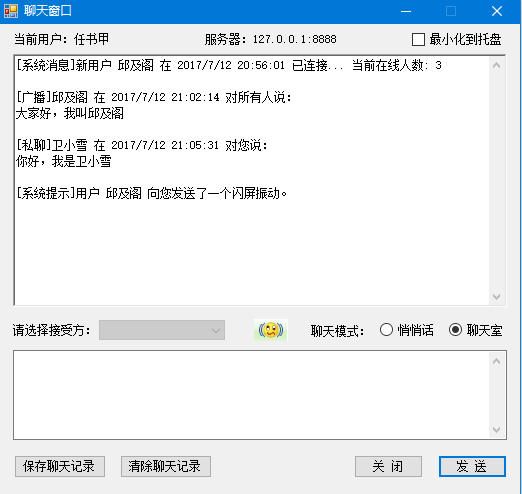


震动窗口：

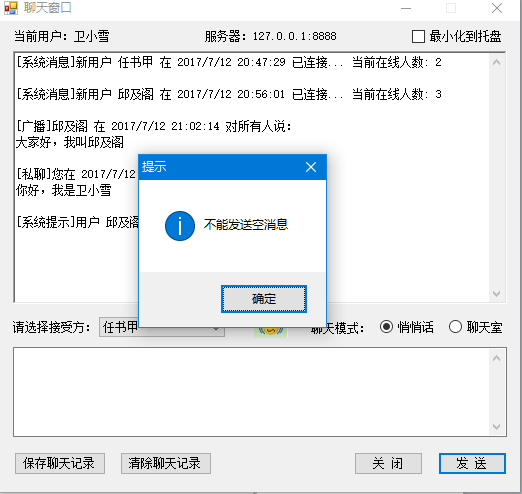




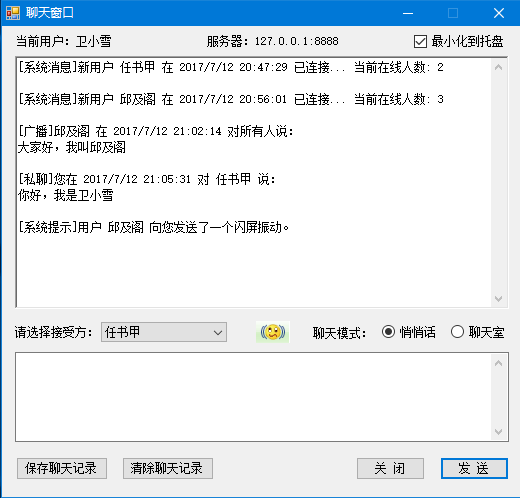




不能发送空消息：

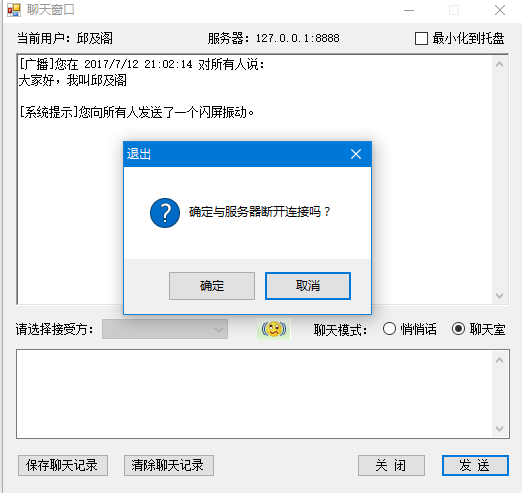


最小化到托盘：

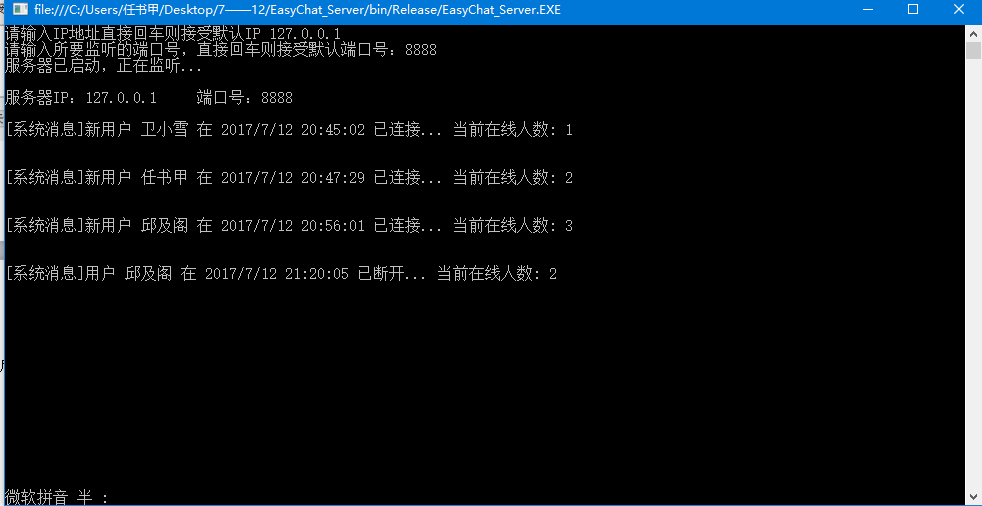


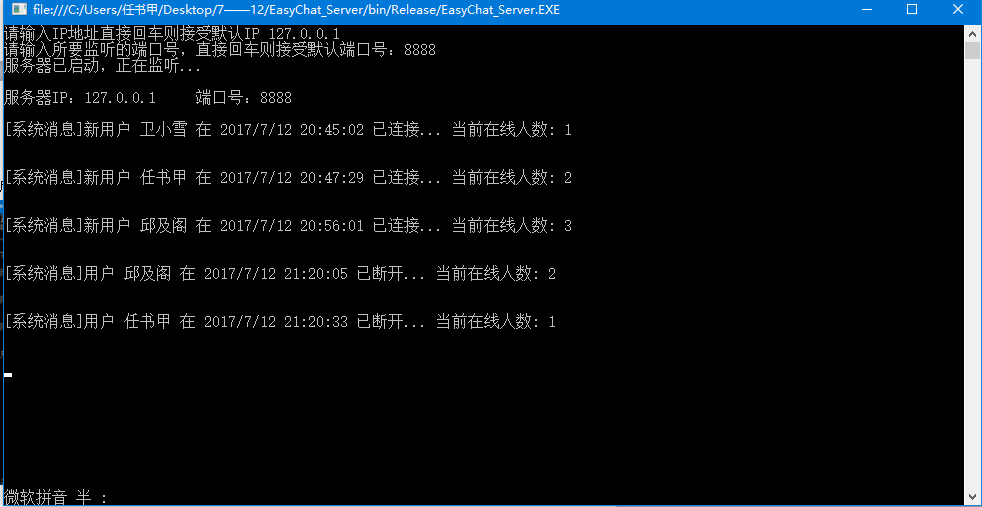


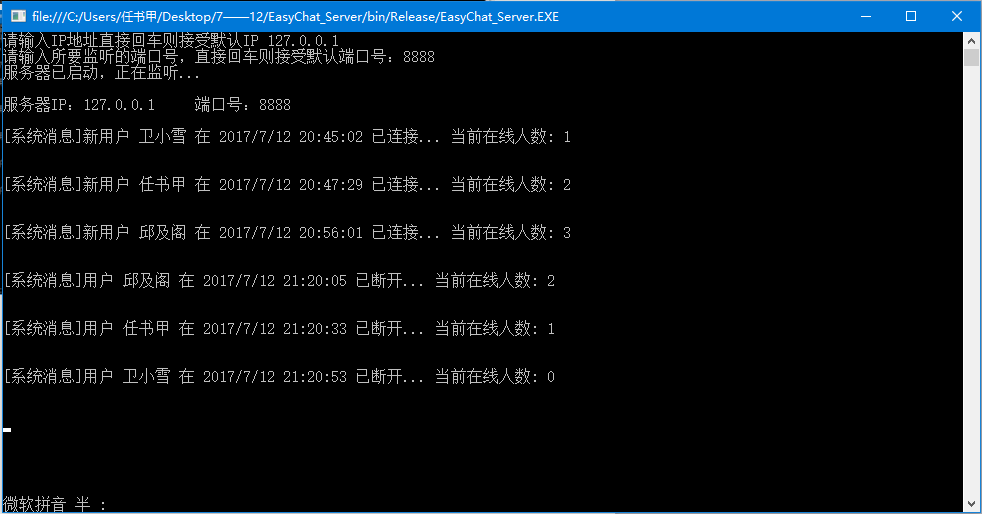
关闭窗口：



关闭窗口后，服务器界面：







1. **源代码中Socket编程部分的详细注释**

服务器端： /// <summary>

/// 服务器程序使用的端口，默认为8888

/// 默认ip是本地环回网

/// </summary>

private IPAddress \_ip = IPAddress.Parse("127.0.0.1");

private int \_port = 8888;

// private IPAddress \_ip = IPAddress.Parse("127.0.0.1");

/// <summary>

/// 配置监听端口号

/// </summary>

public void GetConfig()

{

string portParam;

Console.Write("请输入IP地址直接回车则接受默认IP 127.0.0.1 ");

portParam = Console.ReadLine();

if (portParam != string.Empty)

{

if (!IPAddress.TryParse(portParam, out \_ip))

{

\_ip = IPAddress.Parse("127.0.0.1");

Console.WriteLine("IP不合法,默认IP127.0.0.1被接受!");

}

}

Console.Write("请输入所要监听的端口号，直接回车则接受默认端口号：8888 ");

portParam = Console.ReadLine();

if (portParam != string.Empty)

{

if (!int.TryParse(portParam, out \_port) || \_port < 1023 || \_port > 65535)

{

\_port = 8888;

Console.WriteLine("端口号不合法,默认端口号被接受!");

}

}

}

客户端：

private void login\_btn\_Click(object sender, EventArgs e)

{

//验证数据合法性

if (!ValidateInfo())

{

return;

}

int port = int.Parse(svrport\_tb.Text);

//向服务器发出连接请求

TCPConnection conn = new TCPConnection(\_ipAddr, port);

TcpClient \_tcpc = conn.Connect();

if (\_tcpc == null)

{

MessageBox.Show("无法连接到服务器，请重试！",

"错误",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

}

else

{

NetworkStream netstream = \_tcpc.GetStream();

//向服务器发送用户名以确认身份

netstream.Write(Encoding.Unicode.GetBytes(user\_tb.Text), 0, Encoding.Unicode.GetBytes(user\_tb.Text).Length);

//得到登录结果

byte[] buffer = new byte[50];

netstream.Read(buffer, 0, buffer.Length);

string connResult = Encoding.Unicode.GetString(buffer).TrimEnd('\0');

if (connResult.Equals("cmd::Failed"))

{

MessageBox.Show("您的用户名已经被使用，请尝试其他用户名!",

"提示",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

return;

}

else

{

string svrskt = svrip\_tb.Text + ":" + svrport\_tb.Text;

chat\_frm chatFrm = new chat\_frm(user\_tb.Text, netstream, svrskt);

chatFrm.Owner = this;

this.Hide();

chatFrm.Show();

}

}

}

/// <summary>

/// 初始化登录信息

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void login\_frm\_Load(object sender, EventArgs e)

{

svrip\_tb.Text = "127.0.0.1";//显示在初始位置

svrport\_tb.Text = "8888";//的IP和端口号

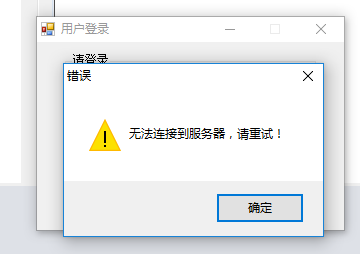
user\_tb.Focus();

}

#endregion

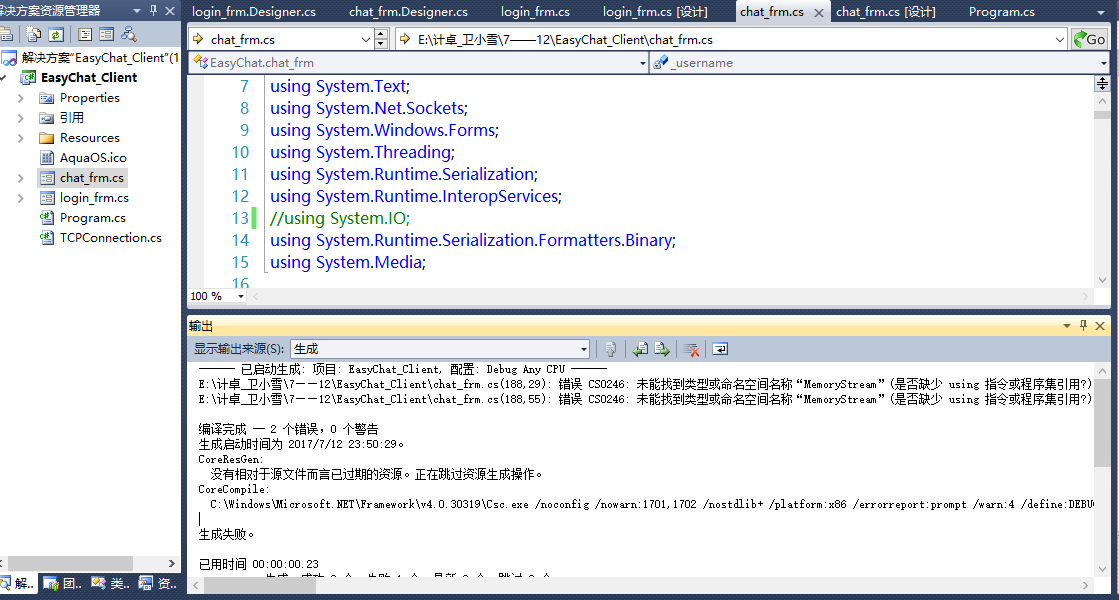
1. **五个以上错误调试的截图，解决方法说明**

* **错误一**



**解决方法：启动服务器**

* **错误二**



**问题描述：程序运行失败**

**问题分析：**

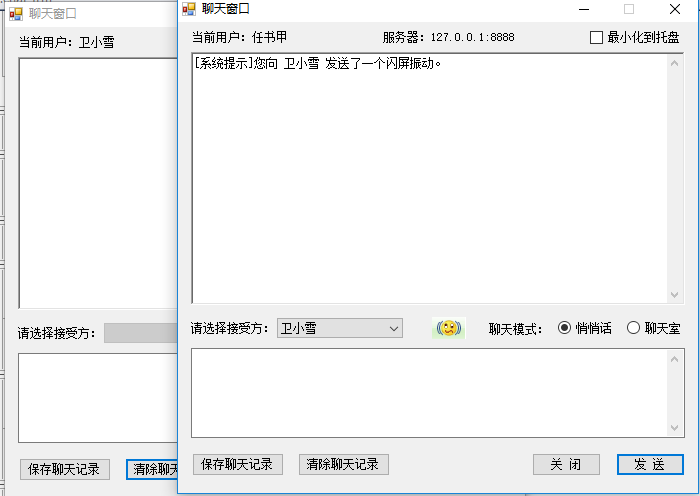
**找不到本地命名空间，少了某个重要的命名空间。**

**修改方法：**

添加 using System.IO;

* **错误三**

**问题描述：在发送屏幕抖动时，对方没有反应。**



**问题分析：**

**1：**看服务器是否连接成功，经过排查，服务器连接是正确无误的

2：看是否是负责判断是否指令的字段\_cmd的存取出现错误。

**解决方法：**

**对\_cmd解释有一点偏差。**

case "12":

{

Nudge();

break;

}

* **错误四**



**错误描述：在运行client界面时会崩溃。**

**解决思路：提示是找不到某个控件，可能是显示的控件的定义函数被误删除了。**

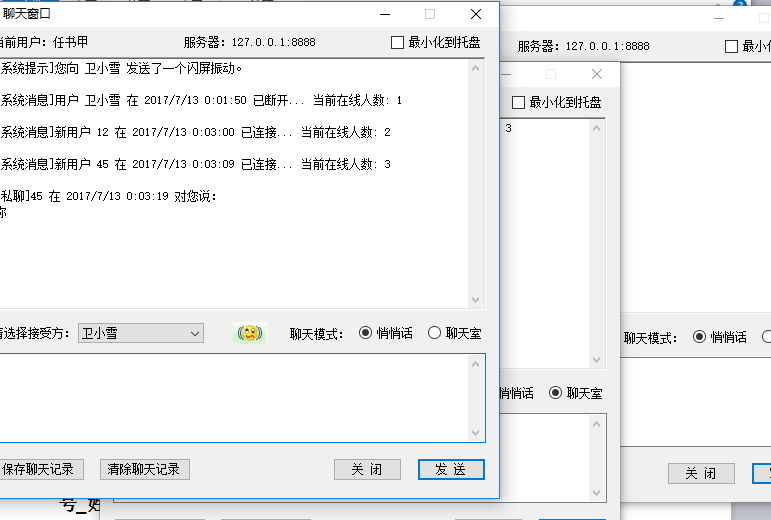
**解决方法:显示的定义出了这个控件的函数，虽然没有函数体。**

private void label2\_Click(object sender, EventArgs e)

{

}

* **错误五**



**错误描述：不能同时和多个用户进行聊天**

**解决思路：**

1. **可以一对一聊天，说明服务器连接是正常的**
2. **可能与线程有关**

**解决方法：**

**每个登陆账号一个线程-------->把通讯任务交给该线程**

1. **源代码**

**（压缩文件电子版，发送至171219976@qq.com，文件名为：组长学号\_姓名）**