

计算机网络 PJ 聊天系统技术文档

By 13307130173 万清甫

● 系统简介

完成了第四个选题：聊天室。该系统支持多个用户同时连接到一个服务器聊天，在聊天室内的用户能看到其它用户发的所有消息，类似群聊。

● 传输

服务器 IP 为本机 127.0.0.1, 监听 20 端口。服务器接收客户端发来的请求，进行相关处理。

● 协议设计

1. 字段

TYPE | IP:PORT | USERNAME | CONTENT | FLAG

1)TYPE

为一位数字

0: 某个用户进入聊天系统

1: 某个用户发送消息

2: 所有用户收到服务器发送的消息

3: 某个用户离开聊天系统

2)IP:PORT

:前的是某个用户的 IP, :后是连接的端口号

3)USERNAME

为用户名

4)CONTENT

为具体内容（可能是用户发送的消息或者是用户接收的来自服务器的消息）……

5)FLAG

结束标识，为“\r\n”

2. 回复

1)用户发送消息到服务器，该请求的回复内容为

2 IP:PORT USERNAME YEAR/MONTH/DAY HOUR:MINUTE:SECOND MESSAGE FLAG

其中 YEAR MONTH DAY HOUR MINUTE SECOND 表示服务器接收到用户发送到的消息的时间 MESSAGE 为用户发送的聊天信息

如图

```
2015/12/8 23:01:44: .....  
CSC说:  
FDFDSFD
```

2)当用户进入系统，发送如下信息到服务器

0 IP:PORT USERNAME FLAG 进入系统!

服务器的返回信息相同

如图

登陆信息

```
0 127.0.0.1:56250 进入系统!  
0 127.0.0.1:56251 进入系统!
```

3)当用户单击窗体的关闭按钮时，发送如下信息到服务器

3 IP:PORT USERNAME 离开系统! FLAG

服务器的返回信息相同

如图

```
0 127.0.0.1:56269 进入系统!  
3 127.0.0.1:56269 ---- 离开系统!
```

● 服务器接收新的客户连接

```
1 个引用  
private void receiveaccept()  
{  
    while (true)  
    {  
        clientsocket[cntclient] = serverreceivesocket.Accept();  
        clientsocket[cntclient].BeginReceive(information, 0, information.Length, SocketFlags.None, new AsyncCallback(receivecallback), clientsocket[cntclient]);  
        this.Invoke((MethodInvoker)delegate  
        {  
            lock (this.txtinfo) this.txtinfo.Text += "0 "+clientsocket[cntclient].RemoteEndPoint.ToString() + " 进入系统!\r\n";  
        });  
        cntclient++;  
    }  
}
```

● 服务器发送接收到的用户聊天信息到所有在线的用户

```
for (int i = 0; i <= cntclient; i++)  
{  
    if (clientsocket[i].Connected)  
    {  
        clientsocket[i].Send(Encoding.Unicode.GetBytes("2 "+DateTime.Now.ToString()+":\r\n" + System.Text.Encoding.Unicode.GetString(information)+"\r\n"));  
    }  
    receivesocket.BeginReceive(information, 0, information.Length, 0, new AsyncCallback(receivecallback), receivesocket);  
    string s = Encoding.Unicode.GetString(information);  
}
```

其中的 `Datetime.Now.ToString()`将当前的系统时间即服务器接收到聊天信息的时间传输给每个客户端

● 客户端发消息给服务器

```
1 个引用  
private void btnsend_Click(object sender, EventArgs e)  
{  
    if (exitornot == true) return;  
    send = Encoding.Unicode.GetBytes("/*1 "+clientsocket.LocalEndPoint+" /*"+txtnickname.Text + "说: \r\n" + this.txtsend.Text + "\r\n\0");  
    if (clientsocket.Connected)  
    {  
        clientsocket.Send(send);  
    }  
}
```

● 客户端接收服务器回复的消息（某个客户端发送的聊天信息）

```
private void receivecallback(IAsyncResult ar)
{
    try
    {
        int receiveend = clientsocket.EndReceive(ar);
        this.Invoke((MethodInvoker)delegate
        {
            lock (this.txtinfo)
            {
                //if (first == true) { first = false; return; }
                this.txtinfo.Text += (System.Text.Encoding.Unicode.GetString(buffer)).Substring(2) + "\r\n";
            }
        });

        clientsocket.BeginReceive(buffer, 0, buffer.Length, 0, new AsyncCallback(receivecallback), null);
    }
    catch (System.Exception err)
    {
        return;
    }
}
}
```

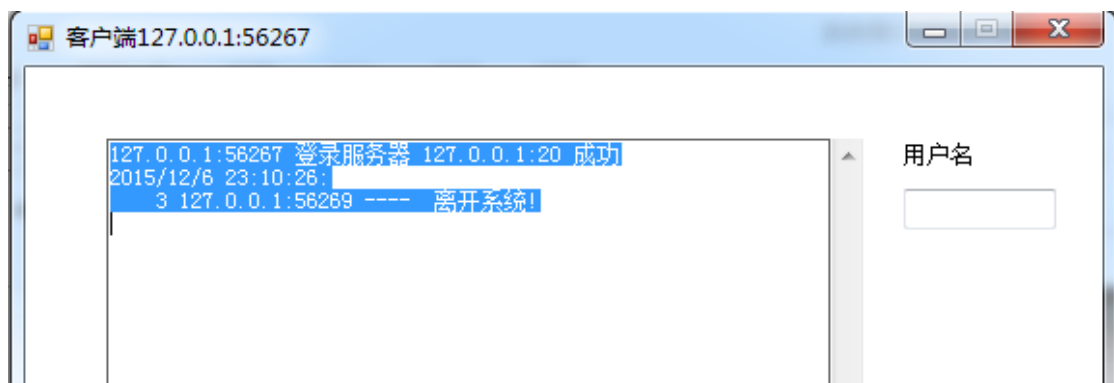
● 客户端离开系统

离开系统要将客户 socket 给 shutdown 关闭连接 同时向服务器发送一条消息表示具体哪个客户端正在离开（TYPE 为 3 的离开消息）

```
1 个引用
private void btnexit_Click(object sender, EventArgs e)
{
    for (int i = 0; i < from.lstcurrent.Items.Count; i++)
    {
        ListViewItem lvi = from.lstcurrent.Items[i];
        if (lvi.Text == clientsocket.LocalEndPoint.ToString())
        {
            from.lstcurrent.Items.RemoveAt(i);
            break;
        }
    }
    if (clientsocket.Connected)
    {
        clientsocket.Send(Encoding.Unicode.GetBytes("3 "+clientsocket.LocalEndPoint + " ---- "+txtnickname.Text+ " 离开系统!\r\n"));
        clientsocket.Shutdown(SocketShutdown.Both);
        clientsocket.Disconnect(false);
    }
    clientsocket.Close();
    exitornot = true;
}
}
```

● 客户端随时更改用户名

客户端可以随时更改昵称



只要在右边的文本框中输入当前昵称就可以修改昵称。

服务器窗口显示当前在线用户

当前在线用户

IP:port	用户名
127.0.0.1:56330	a
127.0.0.1:56331	b
127.0.0.1:56332	c
127.0.0.1:56333	e

某个客户端修改昵称可以随时在服务器界面看到:

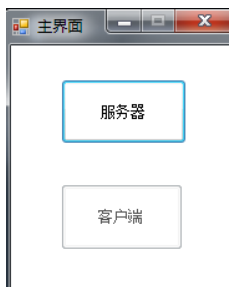
```
1 个引用
private void txtnickname_TextChanged(object sender, EventArgs e)
{
    for (int i=0;i<from.lstcurrent.Items.Count;i++)
    {
        ListViewItem lvi = from.lstcurrent.Items[i];
        if (lvi.Text == clientsocket.LocalEndPoint.ToString()) from.lstcurrent.Items[i].SubItems[1].Text = txtnickname.Text;
    }
}
```

某个客户端离开系统也可以在服务器界面看到:

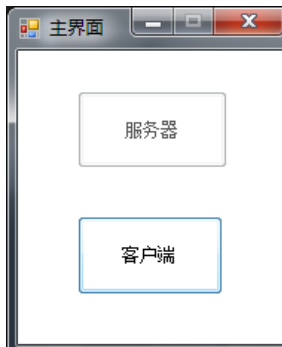
```
1 个引用
private void btnexit_Click(object sender, EventArgs e)
{
    for (int i = 0; i < from.lstcurrent.Items.Count; i++)
    {
        ListViewItem lvi = from.lstcurrent.Items[i];
        if (lvi.Text == clientsocket.LocalEndPoint.ToString())
        {
            from.lstcurrent.Items.RemoveAt(i);
            break;
        }
    }
}
```

● 主界面

开始“客户端”按钮是不可用的

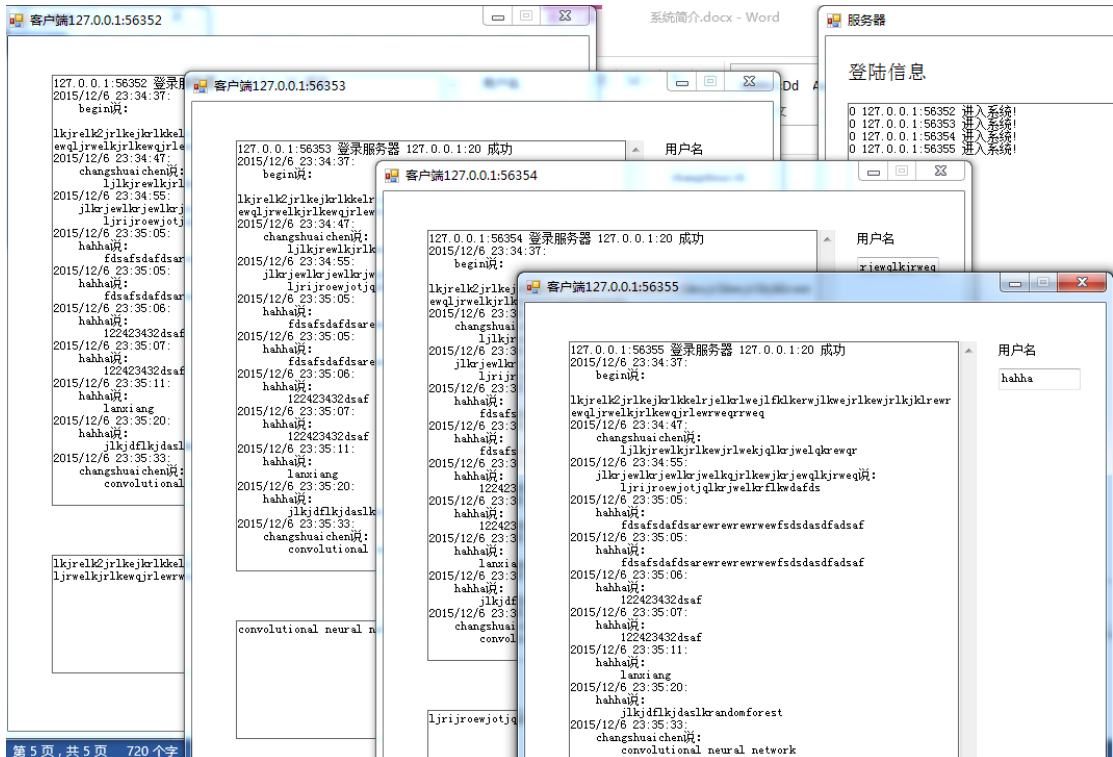


当开启服务器后，“服务器”按钮不可用，“客户端”按钮可用



● 客户端界面

客户端界面显示所有人的群聊信息，信息保证同步。窗体标题为当前的 IP: PORT ip 和端口号



● 服务器界面

为了方便，服务器只显示客户端进入和离开系统，不显示哪个客户发送了消息。

bin	2015/12/6 15:14	文件夹	
obj	2015/12/6 15:14	文件夹	
Properties	2015/12/6 15:13	文件夹	
App.config	2015/12/6 15:13	XML Configurati...	1 KB
chatroom.csproj	2015/12/6 15:34	Visual C# Projec...	5 KB
frmclient.cs	2015/12/6 22:45	Visual C# Sourc...	5 KB
frmclient.Designer.cs	2015/12/6 22:45	Visual C# Sourc...	6 KB
frmclient.resx	2015/12/6 22:45	.NET Managed ...	6 KB
frmmain.cs	2015/12/6 22:44	Visual C# Sourc...	2 KB
frmmain.Designer.cs	2015/12/6 22:44	Visual C# Sourc...	4 KB
frmmain.resx	2015/12/6 22:44	.NET Managed ...	6 KB
frmserver.cs	2015/12/6 22:45	Visual C# Sourc...	5 KB
frmserver.Designer.cs	2015/12/6 22:45	Visual C# Sourc...	9 KB
frmserver.resx	2015/12/6 22:45	.NET Managed ...	6 KB
Program.cs	2015/12/6 16:53	Visual C# Sourc...	1 KB

frmclient.cs 客户端核心代码

frmserver.cs 服务器核心代码

frmclient.Designer.cs 客户端界面代码

frmserver.Designer.cs 服务器界面代码

Program.cs 启动程序代码（默认启动 frmmain）

frmmain.cs 主界面代码

● 安装说明

双击 setup.exe



选择“安装”

即可运行程序

● 实验环境

Windows 7

Microsoft Visual Studio 2013

.Net Framework

实验代码 C#