

# Socket编程

2011360\_牟迪

## 一.实验说明

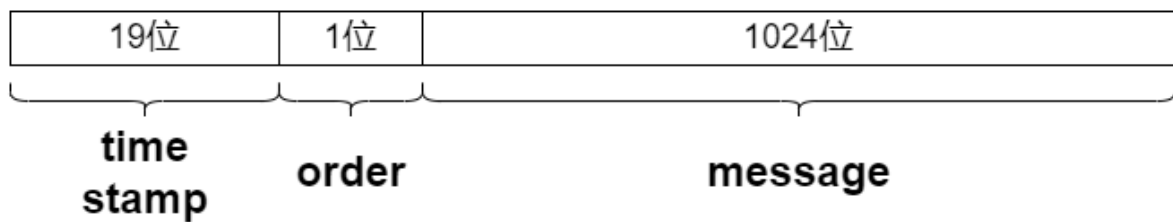
1. 使用流式socket，设计一个两人聊天协议，要去聊天信息带有时间标签，请完整的说明交互信息的**类型，语法，语义，时序**等具体的消息处理方式
2. 对聊天程序进行设计，给出**模块划分**说明，模块的功能和模块的**流程图**
3. 在Windows系统下，利用C/C++对设计的程序进行实现，采用命令行方式，但需要给出使用方法，编写程序时，只能使用基本的socket函数
4. 对实现的程序进行测试
5. 撰写实验报告

## 二.实验过程

### 1.协议设计

在本次利用Socket编写聊天程序的实验中，采用了client-server的模式，同时支持多人聊天室，其中通信消息将由客户端发送给服务器，并通过服务器转发至其余服务端，实现聊天室的功能。同时为客户端与服务器均添加了多线程的控制，通过主线程进行信息的发送，通过额外的消息接收线程来接收消息，并进行相应的处理

对于消息协议的设计，采用了多层的封装，对于**消息类型**的设计，分为**普通消息**(在群聊中的公共消息，发送至服务器端并由服务器进行转发)，**系统消息**(一般为服务器主动向所有用户发送的系统消息，也包含用户进入和离开聊天室时的通知信息)和**私聊消息**(由某一用户发出，通过服务器传达给特定用户)。本次采用字符流进行传输，通过大小为1044字节的字符数组作为发送与接收的缓冲区。信息发送的**语法结构**如下图所示：



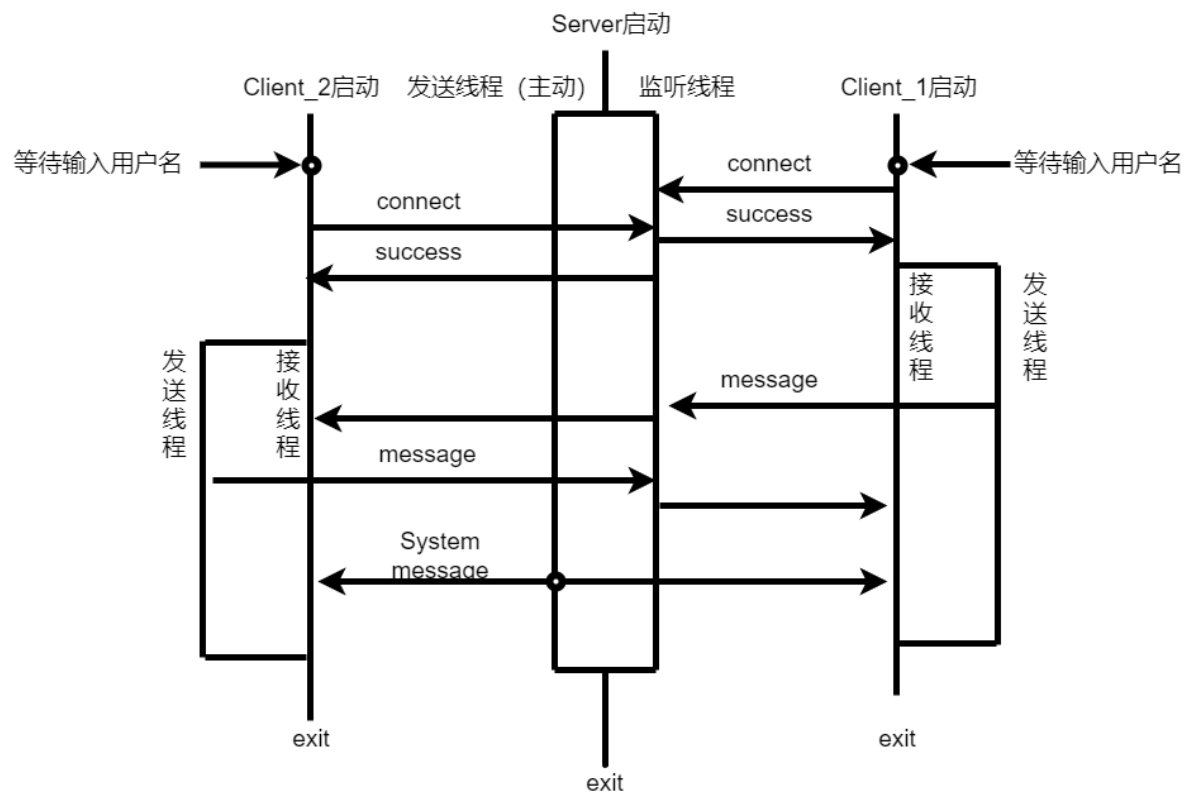
下面对具体的**语义结构**进行说明，

- timestamp：时间戳部分，通过函数getTime()进行当前系统时间的获取，存储在timeRecoed中，传输时拷贝至缓冲区中
- order：标志位，用以区分具体的消息类型。在客户端与服务端随消息进行解析时，依据此位会进行不同的后续动作，其具体含义如下(对应端接收到标志位时的动作)：

order位	Server端	Client端
0	普通消息，进行群发	用户的普通消息
1	新用户的用户名，进行保存并广播	系统消息
2	私聊消息，向对应用户进行转发	其他用户的私聊消息
3	无	准备结束聊天

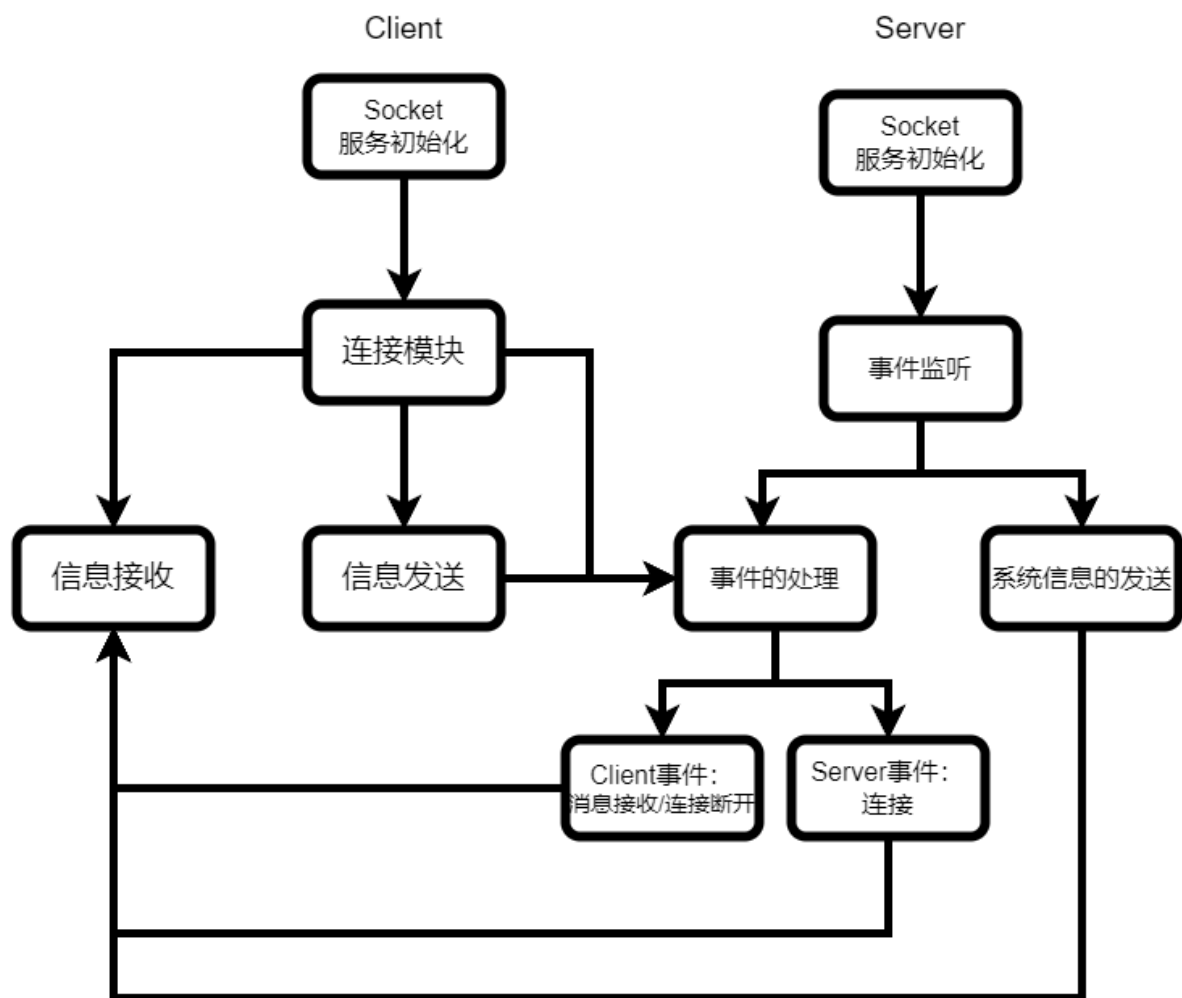
- message：具体的消息内容

聊天程序在运行时的**时序动作**如下图所示：



## 2.程序设计

程序具体的模块执行流程如下图所示：



对重要模块中的代码进行介绍：

以客户端Socket服务初始化模块为例：

#### ▼ Socket服务初始化

```

1 WSADATA wsaData; //异步套接字启动
2 WSStartup(MAKEWORD(2, 2), &wsaData); //声明版本号
3 SOCKET ClientSocket; //建立客户端Socket对象
4 //声明地址类型, 套接字服务类型, 通信协议
5 ClientSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
6 SOCKADDR_IN ServerAddr; //声明服务端的地址
7 ServerAddr.sin_family = AF_INET; //地址类型
8 USHORT uPort = 8888;
9 ServerAddr.sin_port = htons(uPort); //要连接的端口号
10 ServerAddr.sin_addr.S_un.S_addr = inet_addr("127.0.0.1"); //要连接的
    IP地址
  
```

Socket的连接：

#### ▼ 连接模块

```
1 // 由客户端的Socket对象, 向服务器地址进行连接
2 connect(ClientSocket, (SOCKADDR*)&ServerAddr, sizeof(ServerAddr));
```

客户端信息的发送:

#### ▼ 信息发送

```
1 memset(messagebuffer, 0, sizeof(messagebuffer)); //清空消息缓冲
2 memset(sendBuffer, 0, sizeof(sendBuffer)); //清空完整的发送缓冲
3 cin.getline(messagebuffer, sizeof(messagebuffer)); //获取当前要发送的信息
4 string messageToSend = messagebuffer;
5 if (messageToSend.find("over") == 0)
6 {
7     //客户端申请退出
8     cout << "You will exit in 3 sec" << endl;
9     Sleep(3000);
10    return 0;
11 }
12 else if (messageToSend.find(">>") < 1044)
13 {
14     //客户端发送向指定对象的私聊消息
15     getTime(); //获取当前系统时间
16     strcpy(sendBuffer, timeRecord); //时间戳
17     strcat(sendBuffer, "2"); //消息类型
18     strcat(sendBuffer, messagebuffer); //消息内容
19     //通过send函数进行发送
20     send(ClientSocket, sendBuffer, strlen(sendBuffer), 0);
21 }
22 else
23 {
24     //客户端发送普通的群聊消息
25     getTime();
26     strcpy(sendBuffer, timeRecord);
27     strcat(sendBuffer, "0");
28     strcat(sendBuffer, messagebuffer);
29     send(ClientSocket, sendBuffer, strlen(sendBuffer), 0);
30 }
```

客户端信息接收:

#### ▼ 信息接收

```

1  memset(recvBuffer, 0, sizeof(recvBuffer)); //清空消息接收缓冲
2  //通过recv函数进行消息的接收, 同时获得消息长度, 以此判断是否正常
3  int nrecv = recv(cliSock, recvBuffer, sizeof(recvBuffer), 0);
4  if (nrecv > 0)
5  {
6      string receMessage = recvBuffer; //接收到的消息
7      string recetime = receMessage.substr(0, 19); //时间戳
8      char order = receMessage[19]; //消息类型
9      string message = receMessage.substr(20, receMessage.length() -
10     20); //消息内容
11     if (order == '0')
12     {
13         //普通用户消息
14         cout << "[info]# " << recetime << ' ' << message <<
endl;
15     }
16     if (order == '1')
17     {
18         //客户端消息
19         cout << "[server]# " << recetime << ' ' << message <<
endl;
20     }
21     if (order == '2')
22     {
23         //私聊用户消息
24         cout << "[info]# " << recetime << ' ' << message <<
endl;
25     }
26     if (order == '3')
27     {
28         //服务器准备退出
29         cout << "[server]# " << recetime << ' ' << message <<
endl;
30         cout << "You will exit in 3 sec, Welcome to come again" <<
endl;
31     }
32     Sleep(3000);

```

服务器事件处理:

#### ▼ 服务器中事件的监听与处理

```

1  for (int i = 0; i <= total; i++)
2  {
3      //获取产生事件的socket对象标号
4      int index = WSAWaitForMultipleEvents(1, &cliEvent[i], false,
waitTime, 0);

```

```

5     if (index == 0)
6     {
7         WSANETWORKEVENTS networkEvents;
8         WSAEnumNetworkEvents(cliSock[i], cliEvent[i],
    &networkEvents); //查看是什么事件
9         if (networkEvents.lNetworkEvents & FD_ACCEPT)
10        {
11            /*有新用户发起了连接*/
12        }
13        else if (networkEvents.lNetworkEvents & FD_CLOSE)
14        {
15            /*有用户断开了连接*/
16        }
17        else if (networkEvents.lNetworkEvents & FD_READ)
18        {
19            //有用户发来了消息
20            for (int j = 1; j <= total; j++)
21            {
22                //清空接收与发送的缓冲区
23                memset(recvBuffer, 0, sizeof(recvBuffer));
24                memset(sendBuffer, 0, sizeof(sendBuffer));
25                int nrecv = recv(cliSock[j], recvBuffer,
    sizeof(recvBuffer), 0);
26                if (nrecv > 0)
27                {
28                    /*消息的处理, 与客户端相同*/
29                    if (order == '1')
30                    {
31                        //收到了新用户发来的用户名 message
32                        clientName[j] = message;
33                        /*在服务器端打印日志*/
34                        //准备向各个用户分发
35                        strcpy(sendBuffer, recetime.c_str());
36                        strcat(sendBuffer, "1");
37                        string buffertmp = "Welcome " + message +
    " enter the room";
38                        strcat(sendBuffer, buffertmp.c_str());
39                        /*通过for循环向各个客户端发送*/
40                    }
41                    else if (order == '0')
42                    {
43                        //收到了用户消息
44                        /*向各个客户端进行分发*/
45                    }

```

```
46         else if (order == '2')
47         {
48             //用户的私聊消息
49             /*解析要私聊的用户名，并将消息进行转发*/
50         }
51     }
52 }
```

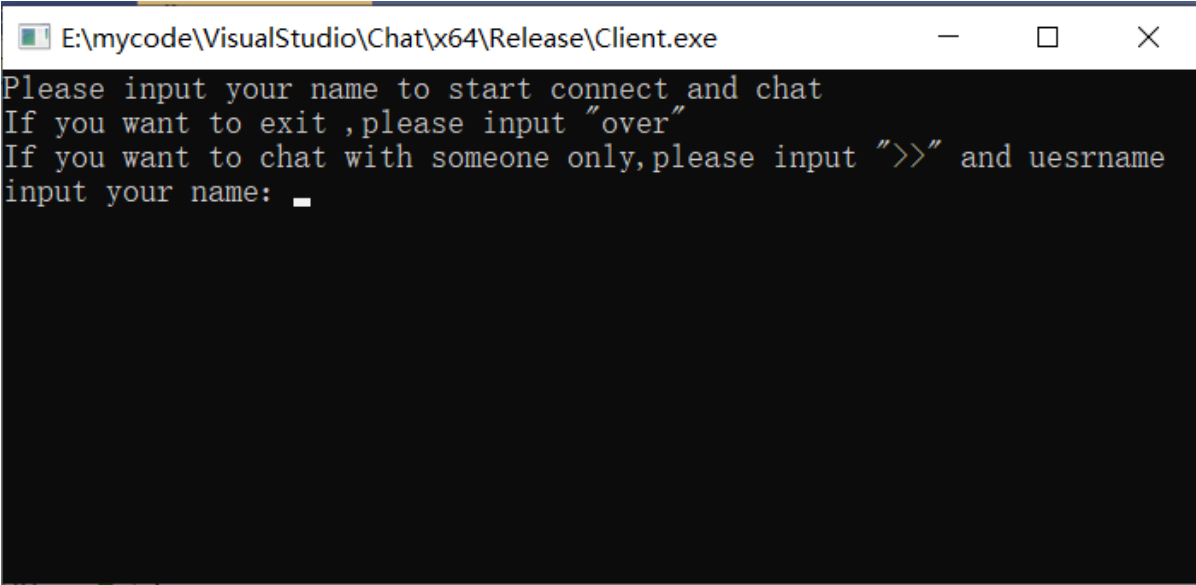
### 3.程序测试}

54 }

服务器启动:



客户端启动：



客户端通过输入用户名启动连接，进入聊天室：



```
E:\mycode\VisualStudio\Chat\x64\Release\Client.exe

Please input your name to start connect and chat
If you want to exit ,please input "over"
If you want to chat with someone only,please input ">>" and uesrname
input your name: KIDSSCC
connect success!
[server]# 2022-10-22 21:31:27 Welcome KIDSSCC enter the room
```

当有多人进入聊天室时，服务器打印日志，并向全体成员广播：

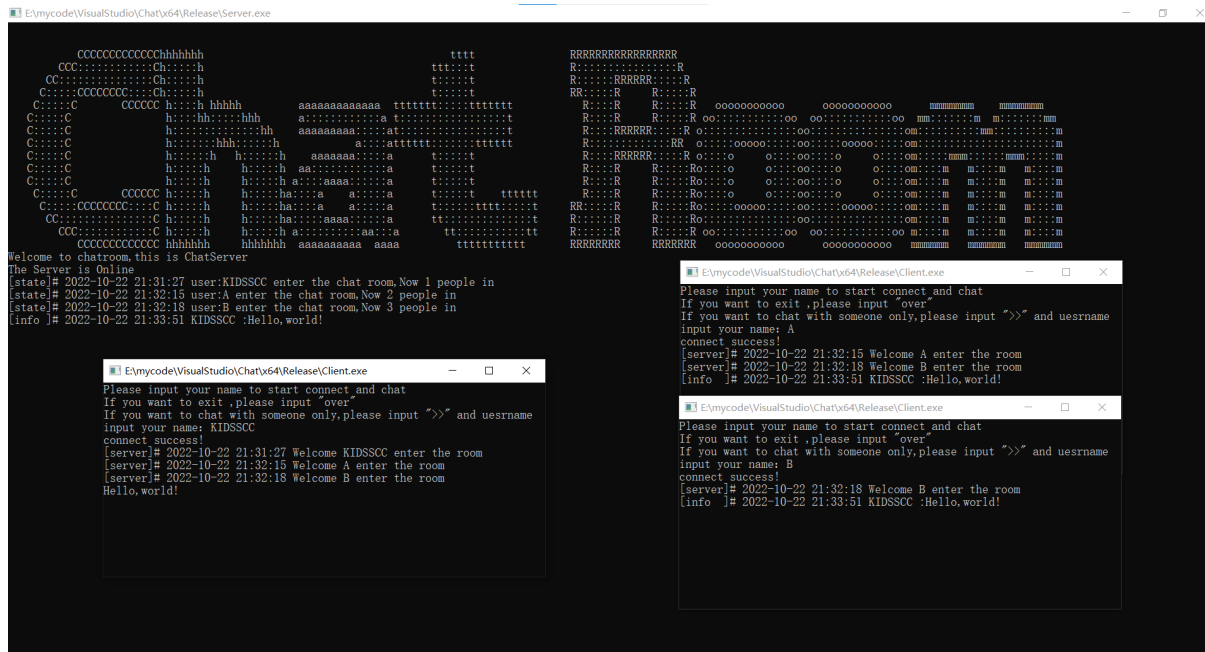
```
E:\mycode\VisualStudio\Chat\x64\Release\Server.exe

Welcome to chatroom,this is ChatServer
The Server is Online
[state]# 2022-10-22 21:31:27 user:KIDSSCC enter the chat room,Now 1 people in
[state]# 2022-10-22 21:32:15 user:A enter the chat room,Now 2 people in
[state]# 2022-10-22 21:32:18 user:B enter the chat room,Now 3 people in
```

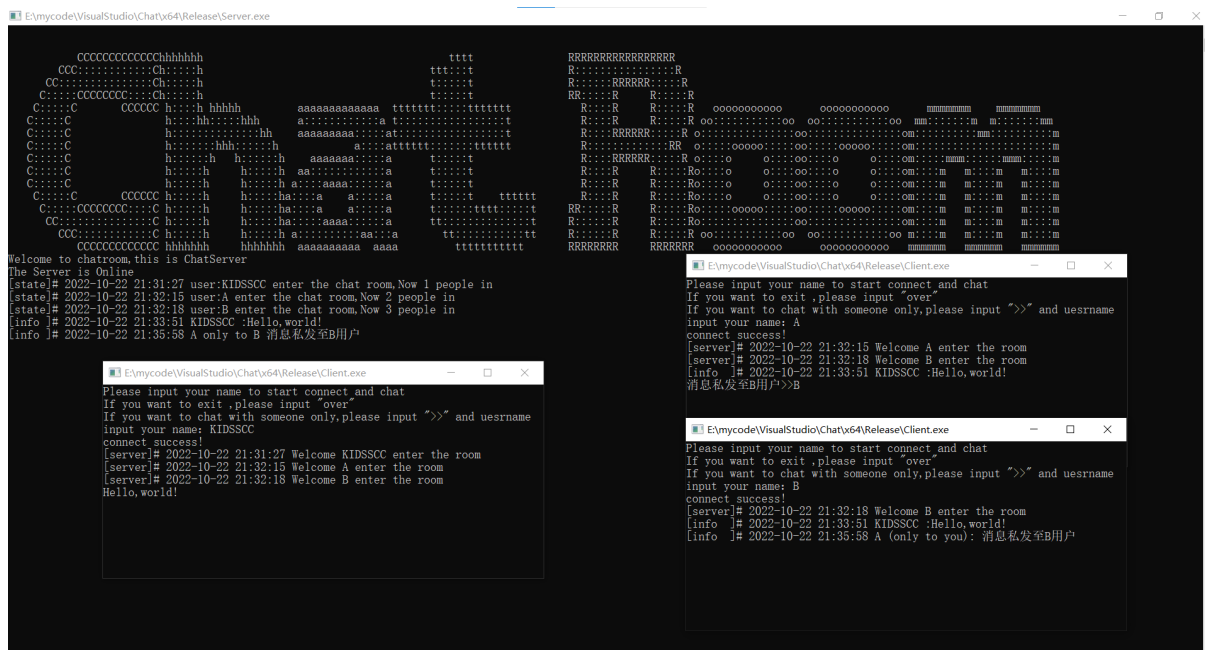
```
E:\mycode\VisualStudio\Chat\x64\Release\Client.exe

Please input your name to start connect and chat
If you want to exit ,please input "over"
If you want to chat with someone only,please input ">>" and uesrname
input your name: KIDSSCC
connect success!
[server]# 2022-10-22 21:31:27 Welcome KIDSSCC enter the room
[server]# 2022-10-22 21:32:15 Welcome A enter the room
[server]# 2022-10-22 21:32:18 Welcome B enter the room
```

由用户发起的普通信息会被服务器广播至所有成员(由用户KIDSSCC发送)



私聊功能，由用户A发起向用户B的私聊，用户KIDSSCC不会收到



有用户离开时，其余用户将收到通知：

