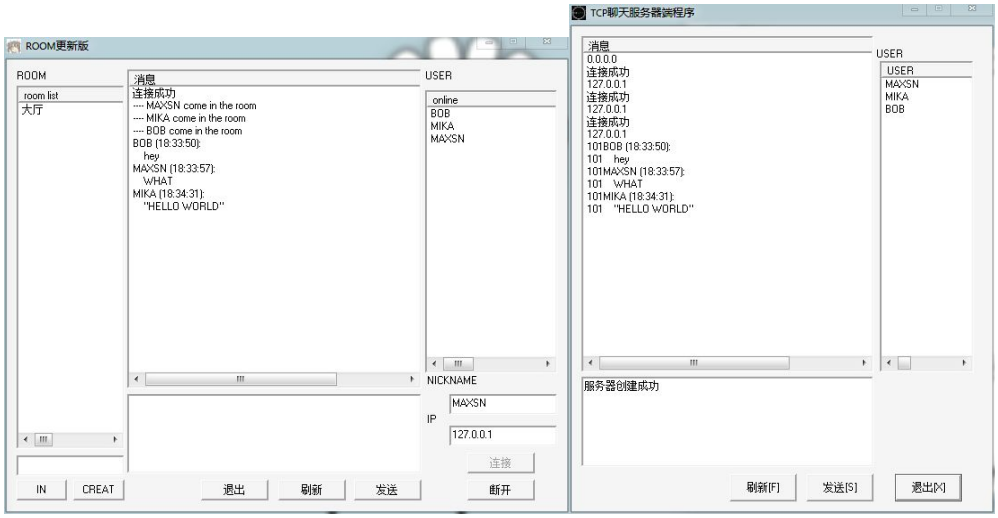
# ROOMCHAT（version.3.0）

## 1.界面：

这是一个基于 TCP/IP 的聊天室软件，整个软件分为服务端部分和客户端部分。界面如下：




## 2.使用步骤：

1.首先在服务器开启服务器端

2.客户打开客户端后在客户端填写自己的 NICKNAME

3.点击连接（默认为作者本地 IP,）

4.一开始默认进入大厅，并自动发出进入房间的信息。

4.用户可在编辑区编辑自己想说的话

5.按发送或者直接 ENTER 即可发送出去

6.在左边的编辑框可以编辑自己想建的房间名称

7.点击 CREAT 建立

8.或点击其他房间之后点击 IN 按钮进入其他房间

9.若想换一个服务器频道可点击断开，按照自己的意愿更换到其他开启了服务端的 IP 点连接即可。

10.完成了对话则点击退出，可退出对话框。

## 3.主要思路：

信息的传输主要是基于 socket 类，令服务端使用多线程监听来自多个客户端的请求，AC 之后以此被基于信息的传递，建立一个套接字的数组来记录所有连通的客户端。

### 3.1 关于显示聊天信息

当用户发送信息之后全部交给服务器来处理，接收到一个客户的信息之后将由服务器处理它，当确认该信息的前缀为‘1’时，将作为聊天信息发送给所有的用户，包括他自己，当客户接受到信息之后也会根据信息的前缀来决定显示到哪里去，去掉前缀之后将显示到聊天框当中去。

### 3.2 关于信息的前缀处理

在这个聊天的信息处理中，使用前三位数字来作为信息的辅助信息。第一位用来分类每一种信息的用处以此来在服务器端和客户端分类处理这些信息。第二三位是每个用户的 clitag 表示他们所在的房间的 tag,通过这个 tag 使得每个人只能可见改 tag 来的信息和 useronline 实现 room 的功能。下面是第一位的分类作用列表：

| 第一位前缀 | 在服务端中代表的意义 | 在客户端中的意义 |
|---|---|---|
| 1 | 从客户端到服务端的信息，表示正常的聊天信息，应该被发送到各个用户的聊天框内容中去 | 这意味着要显示到聊天客户端的所有信息。只要发送到 List 控件中即可。 |
| 2 | 从客户端到服务端的信息，该信息表示有新的用户连接到这个服务端，将用户的昵称加入记录数组中去，并刷新 USER 列表，并发送到所有的客户让他们更新列表 | |
| 3 | 从客户端到服务端的信息，该信息表示有用户退出了，将他的昵称在 | |

| | | |
|---|---|---|
| | 昵称数组中清除，并更新所有用户 USER 列表 | |
| 4 | 从客户端到服务端的信息，该信息帮助处理显示刚进入的用户的进入信息。 | |
| 9 | 从客户端到服务端的信息，创建一个房间并刷新房间列表，更改房间人数，更改创建者的 clitag 为创建的房间的 tag。 | |
| a | 是关于申请者更换房间的信息，更改申请者的 clitag，重置房间人数，并且刷新所有人的房间列表。 | |
| 0 | 代表各种系统功能（socket 连接问题）的信息 | |
| 5 | | 这是关于服务器发来的用户列表的信息，循环接收它并根据 tag 来显示你所在的房间该显示的用户 |
| 8 | | 这是关于服务器发来的房间列表的信息，循环接收 |
| b | 循环接收更新 ROOM 或 USER 内容完成 | |
| c | | 更改客户第二第三位的 clitag 的内容 |
| 9 | | 创建一个新的房间，并作各种更新内容 |
| a | | 表明有用户要换房间，那么更改他的 clitag 并且做相关的更新 |

### 3.3 关于 USERONLINE&&ROOM 的显示

这一部分的功能也靠的是上一部分的分类，并且全部在服务端处理。服务端接受所有的 IN 和 OUT 的信息。并储存在一个用户列表以及 ROOM 列表当中去。并根据其他的 INOUT 的信息来更新这些数组，一旦更新就发送到所有的用户那里是他们也更新。

### 3.4 关于 ROOM 的 CREAT 和 IN

上面已经谈到，关于 CREAT 的处理方式其实和用户的昵称的处理方式是一样的，根据用户的 CREAT 来动态更新。多了一点是需要将他的 clitag 更改。来造成房间的效果。因为每个用户只会看到跟他同 TAG 的用户的 Onlie 显示和信息。

而关于 IN 的部分，通过控件的选中的行号来处理，返获得房间 tag 然后更新每个房间的人数，以及该申请者的 tag，然后 fresh 他的对话框，和所有人的 useronline 即可。

## 4. 目前的源代码：

### 4.1 CLI 部分：

```
class CCSocketDlg : public CDialog
{
// Construction
public:
    CCSocketDlg(CWnd* pParent = NULL);   // standard constructor
    ~CCSocketDlg();
// Dialog Data
    //{{AFX_DATA(CCSocketDlg)
    enum { IDD = IDD_CSOCKET_DIALOG };
    CButton    m_button;
    CListCtrl m_list;
    CEdit  m_edit;
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CCSocketDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);  //    DDX/DDV
support
```

```
    //}}AFX_VIRTUAL

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    //{{AFX_MSG(CCSocketDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    virtual void OnOK();
    afx_msg void OnButton1();
    afx_msg void OnFresh();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
public:
    SOCKET sock,msgsock[50],clisock;
    char clinickname[50][100];//记录客户的名字
    char roomshow[50][100];//记录房间
    int onlinecount[50];//记录房间人数
    int addlen;
    sockaddr_in serv;
    int count;
    int countuser;
    int getcount();
    void sendtoall(char*);

    CListCtrl m_user;
    void freshuser(void);
};
UINT thread(LPVOID);

//初始化对话框
BOOL CCSocketDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);
```

```
    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING,            IDM_ABOUTBOX,
strAboutMenu);
        }
    }

    // Set the icon for this dialog.  The framework does this
automatically
    //  when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);        // Set big icon
    SetIcon(m_hIcon, FALSE);       // Set small icon

    // TODO: Add extra initialization here

    count=0;
    countuser=0;
    m_list.InsertColumn(0,"消息");
    m_list.SetColumnWidth(0,435);
    m_edit.SetLimitText(99);
    m_user.InsertColumn(0,"USER");
    m_user.SetColumnWidth(0,435);
    //m_edit.SetLimitText(99);
    for (int i=0;i<50;i++)
        msgsock[i]=NULL;
    char clinickname[50][100]={0};
    memset(roomshow,0,sizeof(roomshow));
    memset(onlinecount,0,sizeof(onlinecount));
    //char t[100]={0};
    //itoa(onlinecount[1],t,10);
                    //MessageBox("大厅人数");
                    //MessageBox(t);
    CString start="大厅";
    for(int i=0;i<strlen(start);i++) roomshow[1][i]=start[i];
    roomshow[1][strlen(start)]='\0';
    //MessageBox(roomshow[1]);
    //设定地址
```

```
    serv.sin_addr.s_addr=htonl(INADDR_ANY);//绑定
    serv.sin_family=AF_INET;
    serv.sin_port=5000;//htons(5000);
    addlen=sizeof(serv);
    m_button.EnableWindow(FALSE);
    //创建socket
    sock=socket(AF_INET,SOCK_STREAM,0);
    //绑定
    if (bind(sock,(sockaddr*)&serv,addlen))
    {
        m_edit.SetWindowText("绑定错误");
    }else
    {
    m_list.InsertItem(count++,inet_ntoa(serv.sin_addr));
    m_edit.SetWindowText("服务器创建成功");
    //开始侦听
    listen(sock,5);
    //调用线程
    AfxBeginThread(&thread,0);

    }


    return TRUE;  // return TRUE  unless you set the focus to a control
}

void CCSocketDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code
below
//  to draw the icon.  For MFC applications using the document/view
model,
//  this is automatically done for you by the framework.
```

```
void CCSocketDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the
user drags
//  the minimized window.
HCURSOR CCSocketDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

void CCSocketDlg::OnOK()
{

    // CDialog::OnOK();
}


//服务器线程
UINT thread(LPVOID p)
{
```

```
    char buff[100]={0};
    char temp[100]={0};
    CString clitag="01";
    CSize size;
    size.cx=0;
    size.cy=30;
    int i=0,j=0;
    int s=1,msgcount,loop=1,flag=0,s2=1,roomcount;
    CCSocketDlg *dlg=(CCSocketDlg*)AfxGetApp()->GetMainWnd();
    //获得客户端数量
    msgcount=dlg->getcount();
    if (msgcount==-1)
        loop=0;
    if(loop)
    {
        s=1;

    dlg->msgsock[msgcount]=accept(dlg->sock,(sockaddr*)&(dlg->serv)
,&(dlg->addlen));
        if (dlg->msgsock[msgcount]==INVALID_SOCKET)
        {
            dlg->m_edit.SetWindowText("Error accept");
        }
        else
        {
            //启动线程
            AfxBeginThread(thread,0);
            dlg->SetForegroundWindow();
            dlg->m_list.InsertItem(dlg->count++,"连接成功");

    dlg->m_list.InsertItem(dlg->count++,inet_ntoa(dlg->serv.sin_ad
dr));
            dlg->m_list.Scroll(size);
            dlg->m_button.EnableWindow(TRUE);
            //====显示在room
            CString nickname,nickshow;

            while(s!=SOCKET_ERROR)
            {
                //循环接收数据
                s=recv(dlg->msgsock[msgcount],buff,100,0);
                //dlg->SetForegroundWindow();
                if (s!=SOCKET_ERROR)
                {
```

```
                    //======正常信息通道
                    if(buff[0]=='1'){
                        //dlg->MessageBox("SEND   THE   FIREST   LOGIN
MESSAGE");
                        dlg->sendtoall(buff);
                        dlg->m_list.InsertItem(dlg->count++,buff);
                        dlg->m_list.Scroll(size);

                    }
                    //======系统消息通道
                    if(buff[0]=='0'){
                    }
                    //======ROOM USER IN
                    if(buff[0]=='2'){
                        memset(temp,0,sizeof(temp));
                        for(int j=3;j<strlen(buff);j++)
                            temp[j-3]=buff[j];//============去掉标注信息
                        for(int j=0;j<strlen(buff)-3;j++)
                            dlg->clinickname[msgcount][j+1]=temp[j];
                        dlg->clinickname[msgcount][0]=strlen(buff)-3;
//不知原因无法存储的很正常。于是把长度另储存。
                        //将tag放在末端

    dlg->clinickname[msgcount][strlen(buff)-3+1]=clitag[0];

    dlg->clinickname[msgcount][strlen(buff)-3+2]=clitag[1];

    //=====room=======================================
                        for(int        i=dlg->m_user.GetItemCount()-1;
i>=0;i--)
                                dlg->m_user.DeleteItem(i);
                        dlg->countuser=0;
                        for(int i=0;i<=49;i++){
                            memset(temp,0,sizeof(temp));
                            if(dlg->clinickname[i][0]>=1){
                                for(int
j=0;j<dlg->clinickname[i][0];j++)
                                    temp[j]=dlg->clinickname[i][j+1];

    dlg->m_user.InsertItem(dlg->countuser++,temp);
                                dlg->m_user.Scroll(size);
                                char gai[100]="5";
                                char clitagtemp[10]={0};
                                strcat(gai,clitag);
```

```
    gai[1]=dlg->clinickname[i][dlg->clinickname[i][0]+1];

    gai[2]=dlg->clinickname[i][dlg->clinickname[i][0]+2];
                            strcat(gai,temp);
                            dlg->sendtoall(gai);
                            //dlg->MessageBox("发送跟新的代码");
                            //dlg->MessageBox(gai);
                        }
                    }
                    dlg->sendtoall("b");
                    //更新大厅人数
                    dlg->onlinecount[1]+=1;
                    //char t[100]={0};
                    //i//toa(dlg->onlinecount[1],t,10);
                    //dlg->MessageBox("大厅人数");
                    //dlg->MessageBox(t);
                    //给新来的列表
                    roomcount=0;
                    //dlg->MessageBox("刷新ROOM列表");
                    for(int i=1;i<=49;i++){
                        if(dlg->onlinecount[i]>=1||i==1){
                            memset(temp,0,sizeof(temp));
                            for(int
j=0;j<strlen(dlg->roomshow[i]);j++)
                                temp[j]=dlg->roomshow[i][j];
                            //itoa(i,t,10);
                            //dlg->MessageBox(temp);
                            //dlg->MessageBox(dlg->roomshow[i]);
                            char gai[100]="8";
                            strcat(gai,clitag);
                            strcat(gai,temp);
                            dlg->sendtoall(gai);
                        }
                    }
                    dlg->sendtoall("b");
                }
                //======ROOM USER OUT
                if(buff[0]=='3'){

    //memset(,0,sizeof(dlg->clinickname[msgcount]));
                    break;
                    dlg->clinickname[msgcount][0]=0;
                    //dlg->MessageBox("clear");
```

```
                }
                //======ROOM USER FIRST IN
                if(buff[0]=='4'){
                    //显示在对话框
                    char tagtemp[2]={0};
                    memset(temp,0,sizeof(temp));
                    for(int j=3;j<strlen(buff);j++)
                        temp[j-3]=buff[j];//============去掉标注信息
                    for(int i=1;i<=2;i++) tagtemp[i-1]=buff[i];
                    tagtemp[2]='\0';
                    //dlg->MessageBox(tagtemp);
                    clitag=tagtemp;


    dlg->clinickname[msgcount][dlg->clinickname[msgcount][0]+1]=cl
itag[0];

    dlg->clinickname[msgcount][dlg->clinickname[msgcount][0]+2]=cl
itag[1];

                    nickname=temp;
                    nickshow="4"+clitag+"---- "+nickname+" come in
the room";
                    //dlg->MessageBox(nickshow);
                    dlg->m_list.Scroll(size);
                    for(int                    i=0;nickshow[i];i++)
buff[i]=nickshow[i];
                    int len=strlen(nickshow);buff[len]='\0';
                    dlg->sendtoall(buff);
                }
                if(buff[0]=='9'){//创建一个房间&&刷新房间列表
                    memset(temp,0,sizeof(temp));

    //memset(dlg->onlinecount,0,sizeof(dlg->onlinecount));
                    roomcount=1;
                    for(int j=3;j<strlen(buff);j++)
                        temp[j-3]=buff[j];//============去掉标注信息
                    //dlg->MessageBox(buff);
                    //dlg->MessageBox(temp);
                    //找出可以放新房间的位置
                    do{
                        roomcount++;
                    }while(dlg->onlinecount[roomcount]!=0);
                    //dlg->MessageBox(buff);
```

```
                    for(int j=0;j<strlen(buff)-3;j++)
                        dlg->roomshow[roomcount][j]=temp[j];
                    char t[10]={0};
                    //itoa(strlen(buff)-3,t,10);
                    //dlg->MessageBox(t);
                    //itoa(roomcount,t,10);
                    //dlg->MessageBox(t);
                    //itoa(dlg->onlinecount[1],t,10);
                    //dlg->MessageBox(t);
                    //dlg->roomshow[roomcount][0]=t[0];

    dlg->roomshow[roomcount][strlen(buff)-3]='\0';
                    //dlg->MessageBox(buff);
                    //dlg->MessageBox(temp);
                    //dlg->MessageBox(dlg->roomshow[roomcount]);
                    //dlg->roomshow[roomcount][0]=strlen(buff)-3;
                    dlg->onlinecount[roomcount]=1;//目前创建人一人
                    if(roomcount<=9){
                        char roomidtemp[2]={0};
                        itoa(roomcount,roomidtemp,10);
                        memset(buff,0,sizeof(buff));
                        buff[0]='0';
                        buff[1]=roomidtemp[0];
                    }else{
                        memset(buff,0,sizeof(buff));
                        itoa(roomcount,buff,10);
                    }
                    i=atoi(clitag);//获得原来的房间
                    dlg->onlinecount[i]-=1;//原来的房间人数-1
                    clitag=buff;
                    //更改储存的tag

    dlg->clinickname[msgcount][dlg->clinickname[msgcount][0]+1]=cl
itag[0];

    dlg->clinickname[msgcount][dlg->clinickname[msgcount][0]+2]=cl
itag[1];
                    char gai[100]="c";
                    strcat(gai,clitag);
                    //strcat(gai,temp);
                    //dlg->sendtoall(gai);
                    //dlg->MessageBox(clitag);
                    send(dlg->msgsock[msgcount],gai,100,0);// 发 送
clitag
```

```
                    //=====     刷     新     房     间     列     表
=========================================
                    //for(int      i=dlg->m_user.GetItemCount()-1;
i>=0;i--)
                    //     dlg->m_user.DeleteItem(i);
                    //dlg->MessageBox("刷新ROOM列表");
                    roomcount=0;
                    for(int i=1;i<=49;i++){
                        if(dlg->onlinecount[i]>=1||i==1){
                            memset(temp,0,sizeof(temp));
                            for(int
j=0;j<strlen(dlg->roomshow[i]);j++)
                                temp[j]=dlg->roomshow[i][j];
                            //itoa(i,t,10);
                            //dlg->MessageBox(temp);
                        //  dlg->MessageBox(dlg->roomshow[i]);
                            char gai[100]="8";
                            strcat(gai,clitag);

    gai[1]=dlg->clinickname[i][dlg->clinickname[i][0]+1];

    gai[2]=dlg->clinickname[i][dlg->clinickname[i][0]+2];
                            strcat(gai,temp);
                            dlg->sendtoall(gai);
                            //dlg->MessageBox("发送跟新的代码");
                            //dlg->MessageBox(gai);
                        }
                    }
                    dlg->sendtoall("b");
                    //为用户刷新新的用户列表
                    for(int        i=dlg->m_user.GetItemCount()-1;
i>=0;i--)
                            dlg->m_user.DeleteItem(i);
                    dlg->countuser=0;
                    for(int i=0;i<=49;i++){
                        memset(temp,0,sizeof(temp));
                        if(dlg->clinickname[i][0]>=1){
                            for(int
j=0;j<dlg->clinickname[i][0];j++)
                                temp[j]=dlg->clinickname[i][j+1];

    dlg->m_user.InsertItem(dlg->countuser++,temp);
                            dlg->m_user.Scroll(size);
                            char gai[100]="5";
```

```
                        strcat(gai,clitag);

    gai[1]=dlg->clinickname[i][dlg->clinickname[i][0]+1];

    gai[2]=dlg->clinickname[i][dlg->clinickname[i][0]+2];
                        strcat(gai,temp);
                        dlg->sendtoall(gai);
                        //dlg->MessageBox("发送跟新的代码");
                        //dlg->MessageBox(gai);
                    }
                }
                dlg->sendtoall("b");
            }
            if(buff[0]=='a'){//IN the room
                //改变clitag、更改房间人数
                char roomid[2]={0};
                int linecount=0;//有效房间计数
                memset(temp,0,sizeof(temp));
                for(int j=1;j<strlen(buff);j++)
                    roomid[j-1]=buff[j];//===========去掉标注信
息
                linecount=atoi(roomid);
                i=0;
                linecount++;//因为零行没被记录
                //itoa(linecount,buff,10);
                //dlg->MessageBox(buff);
                while(linecount){

    if(dlg->onlinecount[i++]||i++==1)linecount--;//根据提供的客户端行
号找房间。
                }
                i--;
                //itoa(i,buff,10);
                //dlg->MessageBox("找到的tag");
                //dlg->MessageBox(buff);
                if(i<=9){
                    char roomidtemp[2]={0};
                    itoa(i,roomidtemp,10);
                    memset(buff,0,sizeof(buff));
                    buff[0]='0';
                    buff[1]=roomidtemp[0];
                }else{
                    memset(buff,0,sizeof(buff));
                    itoa(i,buff,10);
```

```
                    }
                    i=atoi(clitag);//获得原来的房间
                    dlg->onlinecount[i]-=1;
                    //dlg->MessageBox("原房间的TAG");
                    //dlg->MessageBox(clitag);
                    clitag=buff;

   dlg->clinickname[msgcount][dlg->clinickname[msgcount][0]+1]=cl
itag[0];

   dlg->clinickname[msgcount][dlg->clinickname[msgcount][0]+2]=cl
itag[1];
                    itoa(dlg->onlinecount[i],buff,10);
                    //dlg->MessageBox("原房间的人数");
                    //dlg->MessageBox(buff);
                    //dlg->MessageBox("现在的房间TAG");
                    //dlg->MessageBox(clitag);
                    //dlg->MessageBox("现在的房间人数");
                    i=atoi(clitag);
                    dlg->onlinecount[i]+=1;
                    itoa(dlg->onlinecount[i],buff,10);
                    //dlg->MessageBox(buff);

                    char gai[100]="c";
                    strcat(gai,clitag);
                    send(dlg->msgsock[msgcount],gai,100,0);// 发 送
clitag
                    i=atoi(clitag);
                    //dlg->onlinecount[i]+=1;
                    //刷新列表
                    roomcount=0;
                    //dlg->MessageBox("刷新ROOM列表");
                    for(int i=1;i<=49;i++){
                        if(dlg->onlinecount[i]>=1||i==1){
                            memset(temp,0,sizeof(temp));
                            for(int
j=0;j<strlen(dlg->roomshow[i]);j++)
                                temp[j]=dlg->roomshow[i][j];
                            //itoa(i,t,10);
                            //dlg->MessageBox(temp);
                            //dlg->MessageBox(dlg->roomshow[i]);
                            char gai[100]="8";
                            strcat(gai,clitag);
```

```
    gai[1]=dlg->clinickname[i][dlg->clinickname[i][0]+1];

    gai[2]=dlg->clinickname[i][dlg->clinickname[i][0]+2];
                            strcat(gai,temp);
                            dlg->sendtoall(gai);
                        }
                    }
                    dlg->sendtoall("b");
                    //为用户刷新新的用户列表
                    for(int         i=dlg->m_user.GetItemCount()-1;
i>=0;i--)
                            dlg->m_user.DeleteItem(i);
                    dlg->countuser=0;
                    for(int i=0;i<=49;i++){
                        memset(temp,0,sizeof(temp));
                        if(dlg->clinickname[i][0]>=1){
                            for(int
j=0;j<dlg->clinickname[i][0];j++)
                                temp[j]=dlg->clinickname[i][j+1];

    dlg->m_user.InsertItem(dlg->countuser++,temp);
                            dlg->m_user.Scroll(size);
                            char gai[100]="5";
                            strcat(gai,clitag);

    gai[1]=dlg->clinickname[i][dlg->clinickname[i][0]+1];

    gai[2]=dlg->clinickname[i][dlg->clinickname[i][0]+2];
                            strcat(gai,temp);
                            dlg->sendtoall(gai);
                            //dlg->MessageBox("发送跟新的代码");
                            //dlg->MessageBox(gai);
                        }
                    }
                    dlg->sendtoall("b");
                }
            }

            /*//dlg->SetForegroundWindow();
            if (online!=SOCKET_ERROR)
            {
                dlg->m_user.InsertItem(1,buff);
                dlg->m_user.Scroll(size);
                //dlg->sendtoall(dlg->msgsock[msgcount],buff);
```

```
            }*/
        }

        dlg->clinickname[msgcount][0]=0;
        //dlg->MessageBox("clear");
        for(int     i=dlg->m_user.GetItemCount()-1;     i>=0;i--)
dlg->m_user.DeleteItem(i);
        dlg->countuser=0;
        for(int i=0;i<=49;i++){
            memset(temp,0,sizeof(temp));
            if(dlg->clinickname[i][0]>=1){
                for(int j=0;j<dlg->clinickname[i][0];j++)
                    temp[j]=dlg->clinickname[i][j+1];
                //dlg->MessageBox("更新用户列表");
                //char t[10]={0};
                //itoa(i,t,10);
                //dlg->MessageBox(t);
                //dlg->MessageBox(temp);
                dlg->m_user.InsertItem(dlg->countuser++,temp);
                dlg->m_user.Scroll(size);
                char gai[100]="5";
                strcat(gai,clitag);

    gai[1]=dlg->clinickname[i][dlg->clinickname[i][0]+1];

    gai[2]=dlg->clinickname[i][dlg->clinickname[i][0]+2];
                strcat(gai,temp);
                //dlg->MessageBox(gai);
                dlg->sendtoall(gai);
            }
        }
        dlg->sendtoall("b");
        i=atoi(clitag);
        dlg->onlinecount[i]-=1;
        //退出后刷新房间列表
        //dlg->MessageBox("刷新ROOM列表");
        roomcount=0;
                  for(int i=1;i<=49;i++){
                      if(dlg->onlinecount[i]>=1||i==1){
                          memset(temp,0,sizeof(temp));
                          for(int
j=0;j<strlen(dlg->roomshow[i]);j++)
                              temp[j]=dlg->roomshow[i][j];
                          //itoa(i,t,10);
```

```
                            //dlg->MessageBox(temp);
                            //dlg->MessageBox(dlg->roomshow[i]);
                            char gai[100]="8";
                            strcat(gai,clitag);
                            strcat(gai,temp);
                            dlg->sendtoall(gai);
                        }
                    }
                    dlg->sendtoall("b");

            send(dlg->msgsock[msgcount],"601Disconnected",100,0);
            dlg->m_list.InsertItem(dlg->count++,"Disconnected");
            dlg->m_list.Scroll(size);
            dlg->msgsock[msgcount]=NULL;
            for (int i=0;i<50;i++)
                if (dlg->msgsock[i]!=NULL)
                    flag=1;
            if (flag!=1)
                dlg->m_button.EnableWindow(FALSE);
            closesocket(dlg->msgsock[msgcount]);
        }
    }
    //终止线程
    AfxEndThread(0);
    return 0;
}


//发送数据
void CCSocketDlg::OnButton1()
{
    char buff[100];
    m_edit.GetWindowText(buff,99);
    m_edit.SetWindowText("");
    m_list.InsertItem(count++,buff);
    CSize size;
    size.cx=0;
    size.cy=30;
    m_list.Scroll(size);
    char gai[100]="700";

    //循环向所有客户发送信息
    for (int i=0;i<50;i++)
    {
```

```
        if (msgsock[i]!=NULL){
            strcat(gai,buff);
            send(msgsock[i],buff,100,0);
        }
    }
}


CCSocketDlg::~CCSocketDlg()
{
    for (int i=0;i<50;i++)
        if (msgsock[i]!=NULL)
            send(msgsock[i],"600Disconnected",100,0);
}

//获得还没有使用的socket数组号
int CCSocketDlg::getcount()
{
    for (int i=0;i<50;i++)
    {
        if (msgsock[i]==NULL)
            return i;
    }
    return -1;

}

//向所有客户发送数据
void CCSocketDlg::sendtoall(char *buff)
{
    for (int i=0;i<50;i++)
    {
        if (msgsock[i]!=NULL) {
            char t[10]={0};
            send(msgsock[i],buff,100,0);
        }
    }
}

void CCSocketDlg::OnFresh()
{
    // TODO: Add your control notification handler code here
    m_list.DeleteAllItems();
}
```

```
void CCSocketDlg::freshuser(void)
{
    for(int i=m_user.GetItemCount()-1; i>=0;i--)
        m_user.DeleteItem(i);
    countuser=0;
    int okcount=0;
    char temp[100]={0},buff[100]={0};
    for (int i=0;i<50;i++)
    {
        if (msgsock[i]!=NULL) {
            memset(temp,0,sizeof(temp));
            char t[10]={0};
            itoa(i,t,10);
            MessageBox("对第几号");
            MessageBox(t);
            send(msgsock[i],"f",100,0);
            recv(msgsock[i],buff,100,0);
            MessageBox("得到的回复是");
            MessageBox(buff);
            for(int j=3;j<strlen(buff);j++)
                temp[j-3]=buff[j];
            m_user.InsertItem(countuser++,temp);
            //m_user.Scroll(size);
            sendtoall(buff);
            //okcount++;
        }
    }
    sendtoall("b");
}
```

### 4.2 CLI 部分：

```
class CCSocketcliDlg : public CDialog
{
// Construction
public:
    CCSocketcliDlg(CWnd* pParent = NULL);   // standard constructor
    ~CCSocketcliDlg();
// Dialog Data
    //{{AFX_DATA(CCSocketcliDlg)
    enum { IDD = IDD_CSOCKETCLI_DIALOG };
    CButton    m_disconnect;
```

```
    CButton   m_connect;
    CEdit  m_edit2;
    CListCtrl m_list;
    CButton   m_button1;
    CEdit  m_edit;
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CCSocketcliDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);  //    DDX/DDV
support
    //}}AFX_VIRTUAL

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    //{{AFX_MSG(CCSocketcliDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    virtual void OnOK();
    afx_msg void OnButton1();
    afx_msg void OnButton2();
    afx_msg void OnDisconnect();
    afx_msg void OnFresh();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
public:
    CString msg;
    void GetTime();
    CString strTime;
    CString str;
    void Getlocalip();
    SOCKET clisock;
    sockaddr_in cli;
    int count,ee;
    CString clitag;//用来控制房间
    afx_msg   void   OnLvnItemchangedList2(NMHDR   *pNMHDR,   LRESULT
*pResult);
    CEdit m_edit3;
```

```
    CEdit m_edit4;
    //void InitConsoleWindow(void);
    CListCtrl m_user;
    CListCtrl m_room;
    CButton m_buttoncreat;
    CButton m_buttonin;
    afx_msg void OnClickedCreat();
    afx_msg void OnClickedIn();
};
UINT thread(LPVOID);

BOOL CCSocketcliDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING,            IDM_ABOUTBOX,
strAboutMenu);
        }
    }

    // Set the icon for this dialog.  The framework does this
automatically
    //  when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);        // Set big icon
    SetIcon(m_hIcon, FALSE);       // Set small icon

    // TODO: Add extra initialization here
    int roomcount=0;
    clitag="01";
    m_edit.SetLimitText(99);
```

```
    m_list.InsertColumn(0,"消息");
    m_user.InsertColumn(0,"online");
    m_room.InsertColumn(0,"room list");
    m_user.SetColumnWidth(0,435);
    m_list.SetColumnWidth(0,435);
    m_room.SetColumnWidth(0,435);
    m_room.InsertItem(roomcount++,"大厅");
    m_button1.EnableWindow(FALSE);
    m_disconnect.EnableWindow(FALSE);
    //MessageBox("in the thread");
    m_edit2.SetWindowText("211.80.63.162");
    count=0;
    return TRUE;  // return TRUE  unless you set the focus to a control
}

//发送信息
void CCSocketcliDlg::OnButton1()
{
    CCSocketcliDlg::Getlocalip();
    CCSocketcliDlg::GetTime();


    char buff[100]={0};
    CSize size;
    size.cx=0;
    size.cy=30;
    //获得发送信息
    m_edit.GetWindowText(buff,99);

    CString a;
    a=buff[0];
    for(int i=1;i<100;i++)
    {
        a=a+buff[i];
    } //把buff中的值传给a

    m_edit.SetWindowText("");
    CString nickname="";
    if(a!=""){
        GetDlgItem(IDC_EDIT3)->GetWindowText(nickname); //获取昵称
        msg = nickname+" ("+strTime+"):";
        //m_list.InsertItem(count++,msg);
        send(clisock,"1"+clitag+msg,100,0);
        msg = "    "+a;
```

```
        //m_list.InsertItem(count++,msg);
        send(clisock,"1"+clitag+msg,100,0);
        //m_list.Scroll(size);
        UpdateData(FALSE);
    }
}


//线程
UINT thread(LPVOID v)
{
    char buff[100];
    char array[5][16]=
    {"192.168.186.71",
     "192.168.186.72",
     "192.168.186.73",
     "192.168.186.74",
     "192.168.186.75",
    };
    CSize size;
    size.cx=0;
    size.cy=30;
    int s=1,addcount=0;
    CCSocketcliDlg                              *dlg=(CCSocketcliDlg*)
AfxGetApp()->GetMainWnd();
    dlg->m_connect.EnableWindow(FALSE);
    dlg->m_disconnect.EnableWindow(TRUE);
    while(connect(dlg->clisock,(sockaddr*)&(dlg->cli),sizeof(dlg->
cli)) && dlg->ee!=0)//若连接失败&&想要连接时进入。
    {
        //dlg->MessageBox("come in");
        if (addcount==5)
            addcount=0;
        dlg->cli.sin_addr.s_addr=inet_addr(array[addcount++]);
    }
        if (dlg->ee==1)  dlg->m_list.InsertItem(dlg->count++,"连接成
功");
        //========显示连接成功===========
        CString nickname="";
        dlg->GetDlgItem(IDC_EDIT3)->GetWindowText(nickname); //获取
昵称
        dlg->msg="---- "+nickname+" come in the room";

        CString temp;
```

```
        temp="4"+(dlg->clitag+nickname);
        char te[100]={0};
        for(int i=0;temp[i];i++)te[i]=temp[i];
        //dlg->MessageBox(te);

        send(dlg->clisock,"4"+(dlg->clitag+nickname),100,0);//用于在对话框中通知所有用户

        dlg->msg=nickname;
        int usercount=0;
        //dlg->m_user.InsertItem(usercount++,dlg->msg);
        send(dlg->clisock,"2"+dlg->clitag+nickname,100,0);//用于处理useronline in
        dlg->m_button1.EnableWindow(TRUE);
        dlg->m_edit.SetWindowText("");
        //dlg->SetForegroundWindow();

    //循环获得数据
    while(s!=SOCKET_ERROR && dlg->ee!=0)
    {
        //调用recv函数接收数据
        s=recv(dlg->clisock,buff,100,0);
        //dlg->MessageBox("第一次信息");
        //dlg->MessageBox(buff);
        //dlg->SetForegroundWindow();
        //从服务端获得在线列表
        if(buff[0]=='5'){
            //dlg->MessageBox(buff);
            for(int i=dlg->m_user.GetItemCount()-1; i>=0;i--)
                    dlg->m_user.DeleteItem(i);
            //dlg->MessageBox("清除完毕");
            //dlg->MessageBox(buff);
            int usercount=0;
            while(buff[0]=='5'&&s!=SOCKET_ERROR){
                char temp[100]={0};
                for(int j=3;j<strlen(buff);j++)
                        temp[j-3]=buff[j];
                if(buff[1]==dlg->clitag[0]&&buff[2]==dlg->clitag[1])
                    dlg->m_user.InsertItem(usercount++,temp);
                //dlg->MessageBox(buff);
                //dlg->m_user.Scroll(size);
                s=recv(dlg->clisock,buff,100,0);
                if(buff[0]=='b'||buff[0]=='f')break;
            }
```

```
        }
        if(buff[0]=='8'){
            //dlg->MessageBox(buff);
            for(int i=dlg->m_room.GetItemCount()-1; i>=0;i--)
                   dlg->m_room.DeleteItem(i);
            int roomcount=0;
            while(buff[0]=='8'&&s!=SOCKET_ERROR){
                char temp[100]={0};
                for(int j=3;j<strlen(buff);j++)
                       temp[j-3]=buff[j];
                dlg->m_room.InsertItem(roomcount++,temp);
                //dlg->MessageBox(buff);
                //dlg->m_user.Scroll(size);
                s=recv(dlg->clisock,buff,100,0);
                if(buff[0]=='b'||buff[0]=='f')break;
            }
        }
        if(buff[0]=='c'){
            char temp[100]={0};
            for(int j=1;j<strlen(buff);j++)
                temp[j-1]=buff[j];
            dlg->clitag=temp;
            //dlg->MessageBox("换TAG");
            //dlg->MessageBox(dlg->clitag);
            //更改之后应当重新显示进入
            dlg->GetDlgItem(IDC_EDIT3)->GetWindowText(nickname);  //
获取昵称
            dlg->msg="---- "+nickname+" come in the room";
            send(dlg->clisock,"4"+(dlg->clitag+nickname),100,0);//用
于在对话框中通知所有用户
            continue;
        }
        if(buff[0]=='f'){
            dlg->msg="5"+dlg->clitag+nickname;
            send(dlg->clisock,dlg->msg,100,0);
            dlg->MessageBox("发送刷新信息");
            dlg->MessageBox(dlg->msg);
            continue;
        }
        if(buff[0]!='5'&&buff[0]!='8'&&buff[0]!='b'){
            char temp[100]={0};
            for(int j=3;j<strlen(buff);j++)
                temp[j-3]=buff[j];
            //&&buff[1]==dlg->clitag[0]&&buff[2]==dlg->clitag[1]
```

```
            if                    (s!=SOCKET_ERROR                    &&
dlg->ee!=0&&buff[1]==dlg->clitag[0]&&buff[2]==dlg->clitag[1])
                dlg->m_list.InsertItem(dlg->count++,temp);
            //dlg->MessageBox(buff);
        }
        dlg->m_list.Scroll(size);

    }
    //发送断开命令
    //send(dlg->clisock,"001Disconnected",100,0);
    dlg->m_button1.EnableWindow(FALSE);
    dlg->m_connect.EnableWindow(TRUE);
    dlg->m_disconnect.EnableWindow(FALSE);
    closesocket(dlg->clisock);
    AfxEndThread(0);
    return 0;
}
CCSocketcliDlg::~CCSocketcliDlg()
{
    send(clisock,"0"+clitag+"Disconnected",100,0);
}

//连接服务器
void CCSocketcliDlg::OnButton2()
{
    char ipaddress[35];
    m_edit2.GetWindowText(ipaddress,30);
    cli.sin_addr.s_addr=inet_addr(ipaddress);

    cli.sin_family=AF_INET;
    cli.sin_port=5000;//htons(5000);
    //创建socket
    clisock=socket(AF_INET,SOCK_STREAM,0);
    ee=1;
    printf("in the thread");
    //启动线程
    AfxBeginThread(thread,0);



}

void CCSocketcliDlg::OnDisconnect()
{
```

```
    //send(clisock,"301"+msg,100,0);//用于处理useronline断开
    //ee=0;
    closesocket(clisock);
}


void CCSocketcliDlg::Getlocalip()
{
   char szHostName[128];
   if (gethostname(szHostName,128)==0)
   {
    struct hostent * pHost;
    pHost = gethostbyname(szHostName);

    str=inet_ntoa(*(struct in_addr*)pHost->h_addr_list[0]);
   }
}

void CCSocketcliDlg::GetTime()
{
    //获取系统时间
    CTime tm;
    tm=CTime::GetCurrentTime();
    strTime=tm.Format("%X");//现在时间是%Y年%m月%d日
}

void CCSocketcliDlg::OnFresh()
{
    // TODO: Add your control notification handler code here
     m_list.DeleteAllItems();
}

/*
void CCSocketcliDlg::InitConsoleWindow(void)
{
    int nCrt = 0;
    FILE* fp;
    AllocConsole();
    nCrt  =  _open_osfhandle((long)GetStdHandle(STD_OUTPUT_HANDLE),
_O_TEXT);
    fp = _fdopen(nCrt, "w");
    *stdout = *fp;
    setvbuf(stdout, NULL, _IONBF, 0);
}*/
```

```
void CCSocketcliDlg::OnClickedCreat()
{
    // TODO: 在此添加控件通知处理程序代码
    char buff[100]={0};
    CString roomname;
    GetDlgItem(IDC_EDIT4)->GetWindowText(roomname); //获取昵称
    //m_list.InsertItem(count++,msg);
    //MessageBox("9"+clitag+roomname);
    send(clisock,"9"+clitag+roomname,100,0);
    //recv(clisock,buff,100,0);
    //MessageBox("接受过了");
    //MessageBox(buff);
    //MessageBox("CREAT还没完啊");
    //clitag=buff;
    OnFresh();   //刷新对话跨
}


void CCSocketcliDlg::OnClickedIn()
{
    // TODO: 在此添加控件通知处理程序代码
    CString str; char buff[100]={0};
    char roomid[10]={0};
    int line;//行号
    CString nickname;
    //首先得到点击的位置
    POSITION pos=m_room.GetFirstSelectedItemPosition();
    if(pos==NULL) return;
    //得到行号，通过POSITION转化
    line=(int)m_room.GetNextSelectedItem(pos);
    itoa(line,roomid,10);
    for(int i=strlen(roomid);i>=1;i--) roomid[i]=roomid[i-1];
    roomid[0]='a';
    //MessageBox("IN有效行数");
    //MessageBox(roomid);
    send(clisock,roomid,100,0);//通过逐个确认行号获得roomtag
    //recv(clisock,buff,100,0);//get roomtag
    //clitag=buff;
    OnFresh();   //刷新对话跨
}
```

## 5. 开发结语：

说是开发感受也是整个课程下来的感受。不是电院的学生，通过任选课选上了这门课，本身对算法之类的更加有兴趣，对应用设计开发一窍不通。而且从来没接触过 MFC，并且开课一开始的时候特别厌恶 MFC，以前最讨厌的事情就是看别人的代码，感觉 MFC 里面充斥了别人写好的代码，给好的框架。实际上慢慢的感受到了他的便利。说实在的上课学到的东西真的很少，但是老师提供的东西非常有用，像 PPT 和大量的实例全部的代码，在课后上机的时候才是真正学懂东西的时候。

在做这个案例的时候是基于老师网络编程的那一课所讲的知识，当天只讲了 socket 和服务端和客户端一对一的案例。在研究之后，决定了要做一个聊天室，这个聊天室的整个构想是怀旧性质的，模仿的是 10 年前左右流行的聊天室的功能。（参考了 http://drrr.us/这个网站（翻《无头骑士异闻录》中的聊天室，死宅性质暴露），很早就想自己去作一个这样的，遇到这样的机会再好不过了，可惜时间和技术有限，没有做 SKIN）在 FTP 上找到了聊天室的案例。研究了其中的 socket 和多线程的做法。为整个软件的框架打下了基础。

然后考虑整个网站的构想，最首先是从数据和功能来展开考虑，数据最重要的是两个一个是每个用户的数据，其中包括用户的昵称，IP,目前的房间号。其次是所有房间的数据，包括房间的标签，以前目前有多少人等等。而从功能来考虑最大的三块功能也就是三块显示的内容，右边显示在线用户，左边是现有的房间，中间是对话内容。整体考虑之后，决定要减少对字符串的使用以免在我不熟悉的环境下出不易调试的错误。所以所有的数据和网站一样全都在服务端中，客户端只储存自己当前的昵称和客户房间 TAG。并尽量使用即时内容，减少储存量。所以整个案例基本上靠即时通信时传递的信息来起作用，对所有流通的信息，加上前缀，来执行各类的功能，而其实所有的用户都是在同一房间，只是他们非同个标签不可见，（因为没有时间和技术去做多对话框。）

这些都几乎在一瞬间想好了，构建代码也没花多久。深深的感受到了软件的重要之处除了构想还有DEBUG。而做题的时候只要有了好的算法，debug就可以深信不疑的做下去，应用开发却不是，根本没有算法，只有模拟，不断的debug。最重要的是我不会MFC的debug功能，于是我想用输出的方式也各种失败,所以代码里

最后到处都是我的messagebox调试的时候各种跳框非常的难受。不过经过了长期的调试最终还是大部分的功能都算是没有太大的问题了。

程序发送到很多同学那里一起测试之后也都没有大的问题，遗留了几个小问题，一个是处理速度和网速的问题会导致对话信息出现一些先后的错误，当然大部分时候都正常。二，有的用户网络特殊的，容易断开，并导致整个软件的房间功能出现一些问题。没办反，这个版本就暂时做到这样吧。还有就是偶尔有些电脑并不能作为客户端，不知为何，应该不是端口的问题。在连接的时候发生了超时、失败的问题。

通过这一次的案例，让我对socket，websocket产生了一定的兴趣，深刻又一次的感受到了debug的痛苦。不过还是很感谢有这样一次机会。

王誉锡 5131309040

国际与公共事务学院