聊天室TeamChat实验报告

实验环境配置：

- Windows 10 64-bit
- Visual Studio 2017/2019
- masm32汇编开发工具包

文件结构

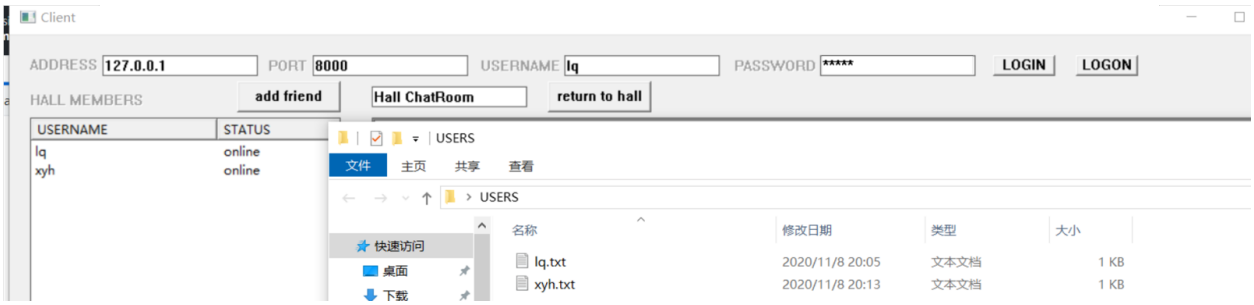
```
src/
  -server-masm32/          服务端
    -server/               服务端源文件
      header.inc           全局变量定义
      main.asm             服务端通讯部分逻辑
      data.asm             服务端数据存储/读取逻辑
  -server.sln              服务端项目文件

  -client-masm32/          客户端
    -client/               客户端源文件
      header.inc           全局变量定义
      main.asm             客户端GUI部分
      client.asm           客户端通讯部分
    -client.sln            客户端项目文件
doc/
  report.pdf               实验报告
```

一、功能展示

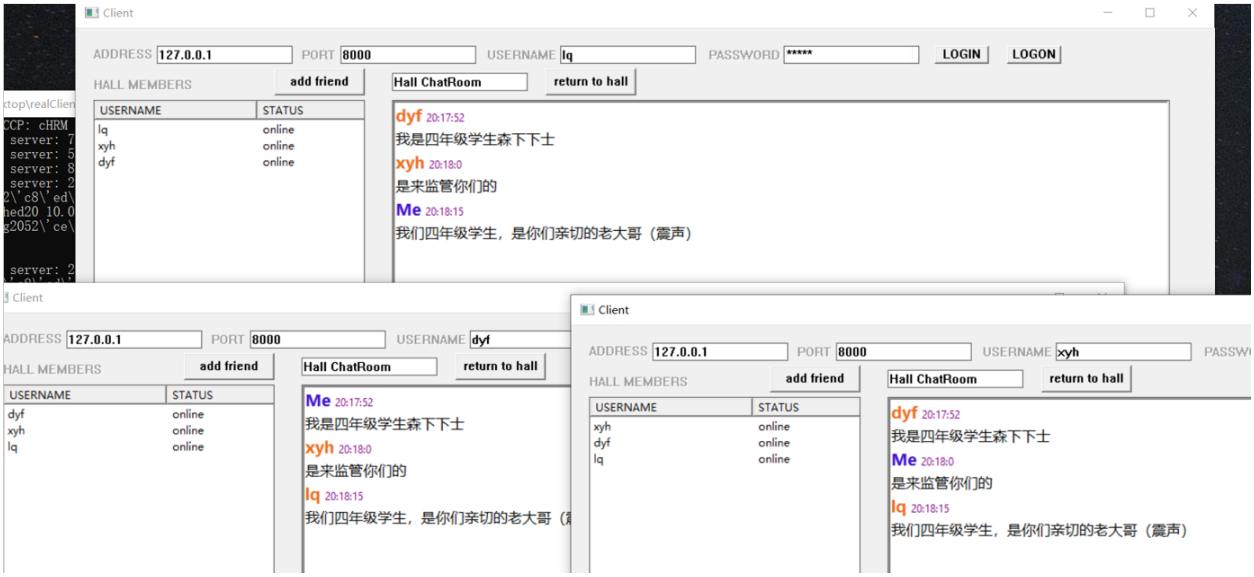
1、注册与登录

用户填写IP地址，端口号，用户名，密码后，点击logon或login按钮，即可进行注册和登录。



2、大厅聊天室

用户登录之后，会进入大厅聊天室，这是一个群聊的聊天室，所有已登录的用户都能在该聊天室发送消息。聊天室中的用户显示在左侧上方的列表中。点击列表上方的return to hall 按钮可以切换到大厅聊天室。



3、好友私聊

用户登录之后，会加载所有添加过的好友，好友显示在左侧下方的列表中，双击好友可以切换到和他/她的私聊窗口，好友的状态会在列表的status列中显示。

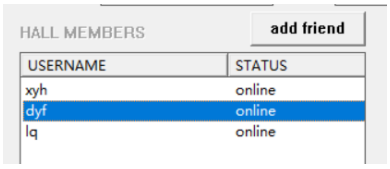
4、添加好友

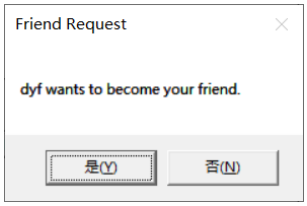
用户可以添加大厅中的在线用户为好友，单击大厅聊天室用户列表的用户后，点击上方的add friend按钮，即可向对方发送好友添加请求。

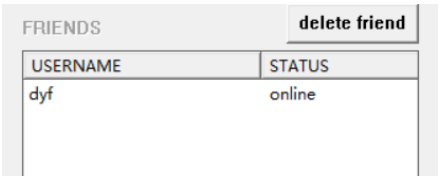
此时该好友会出现在下方的好友列表中，状态为padding（等待对方同意）

此时对方会接收到一个申请的弹窗，当对方同意请求后， 我方的好友列表中对方的状态会变为online（在线）

当对方拒绝请求，我方好友列表中对方的状态会变为 apply reject（申请被拒绝）



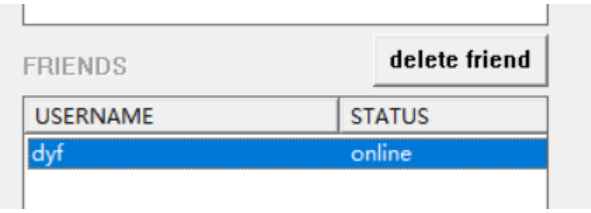




5、删除好友

用户可以删除自己的好友，单击好友列表中的用户，点击上方的 delete friend按钮，即可向服务器发送删除好友的请求。

发送后该好友会被删除。如果对方在线，服务器会向对方发送一条信息，在对方的好友列表中，我方的状态会变为 delete（被删除），对方再次上线后我方会从对方的好友列表中消失。



6、聊天消息编辑与发送（富文本/图片）

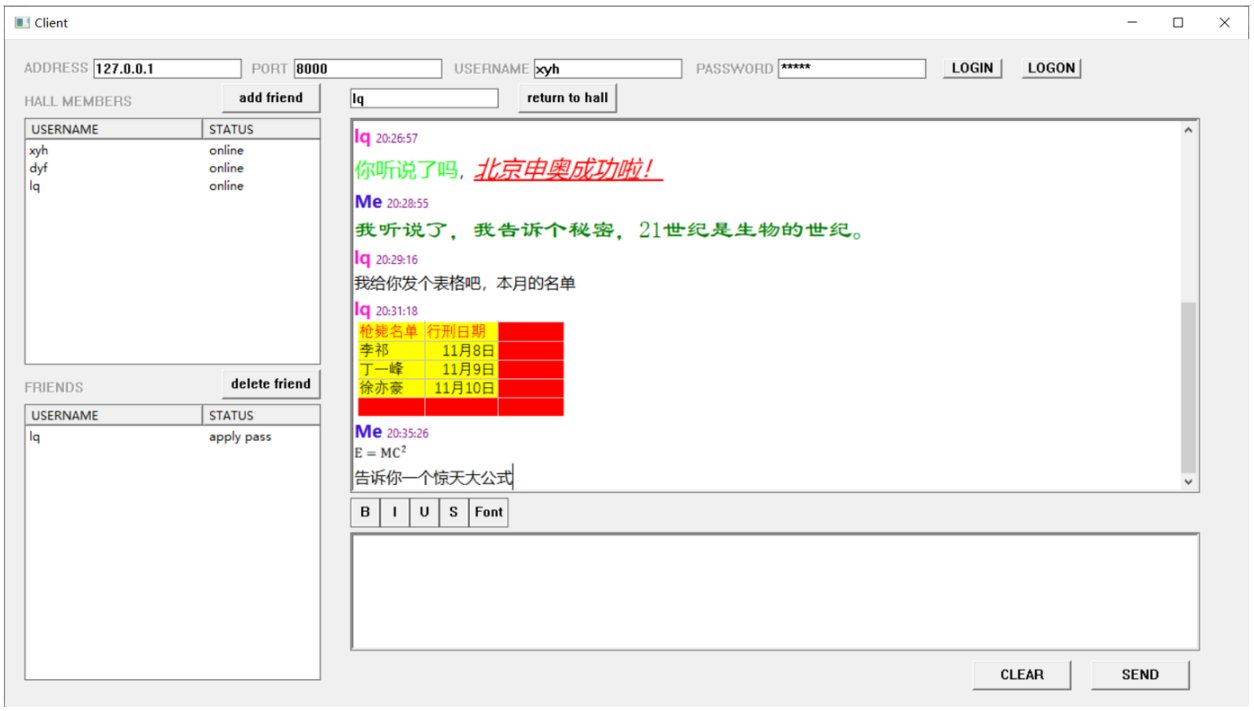
用户可以在下方的编辑栏中写入自己要发送的消息，消息支持富文本编辑。

我们实现了富文本编辑器和富文本显示窗口。

在下方的编辑栏中，用户可以对自己输入的文本进行调整，可以调整字体粗细，下划线，斜体，删除线，字体大小，字体颜色，字体样式。还可以插入图片和表格。

发送消息或接受消息，富文本会显示在聊天显示框中。消息的传输是以RTF格式传输的。


Client
ADDRESS 127.0.0.1
PORT 8000
USERNAME xyh
PASSWORD *****
LOGIN
LOGON
HALL MEMBERS
add friend
lq
return to hall
USERNAME STATUS
xyh online
dyf online
lq online
FRIENDS
delete friend
USERNAME STATUS
lq apply pass
你好
认识你很高兴
lq 20:25:45
你好，我也在网上冲浪，天翼3G真是太快啦！
lq 20:26:57
你听说了吗，北京申奥成功啦！
B I U S Font
我听说了，我告诉你一个秘密，21世纪是生物的世纪。
CLEAR
SEND


Client
ADDRESS 127.0.0.1
PORT 8000
USERNAME xyh
PASSWORD *****
LOGIN
LOGON
HALL MEMBERS
add friend
lq
return to hall
USERNAME STATUS
xyh online
dyf online
lq online
FRIENDS
delete friend
USERNAME STATUS
lq apply pass
lq 20:26:57
你听说了吗，北京申奥成功啦！
Me 20:28:55
我听说了，我告诉个秘密，21世纪是生物的世纪。
lq 20:29:16
我给你发个表格吧，本月的名单
lq 20:31:18
枪毙名单 行刑日期
李祁 11月8日
丁一峰 11月9日
徐亦豪 11月10日
Me 20:35:26
E = MC²
告诉你一个惊天大公式
B I U S Font
CLEAR
SEND

二、实现思路

1. 网络编程

使用ws2_32.inc提供的win32 socket api进行编程。具体使用的函数有：

- `invoke socket, AF_INET, SOCK_STREAM, 0` 创建socket
- `invoke bind, sockfd, addr @stSin, sizeof @stSin` 绑定端口
- `invoke listen, sockfd, BACKLOG` 进行监听
- `invoke CreateThread, NULL, 0, offset serviceThread, @clientid, NULL, esp` 创建一个新线程处理Client请求
- `invoke select, 0, addr @stFdset, NULL, NULL, addr @stTimeval` 进行轮询请求，近似异步地处理每一条请求
- `invoke send, sockfd, addr @buffer, @bufLen, 0` 发送消息
- `invoke recv, sockfd, addr @buffer, BUFSIZE, 0` 接收消息

2. 通讯协议

为了处理注册、登录、接发消息等一系列请求，我们自行定义了一套通讯协议，具体的我们规定:

```
client -> server:
  0                             正确收到信息
  1                             收到信息但有错误
  2   username   password       用户注册
  3   username   password       用户登录
  4                             用户登出
  5   message                   聊天室聊天
  6   tgtUser    message        与目标用户私聊
  7   tgtUser                   发送好友申请
  8   tgtUser    0/1            回复好友申请 (0:拒绝; 1:同意)
  9   tgtUser                   删除好友

server -> client:
  0   failMsg                   客户端请求失败
  1                             客户端请求成功
  2   srcUser    message        聊天室聊天
  3   srcUser    message        私信聊天
  4   srcUser                   好友申请
  5   U1 s1 U2 s2 ....          好友列表
  6   friend     1/2/4/5/6      好友上线/下线/申请通过/申请不通过/被删
  7   U1 U2 U3 ...              聊天室用户列表
  8   username   0/1            用户进入/离开聊天室
```

3. 数据存储

为了实现密码、好友信息等的持久化存储，我们将数据存在了文件夹 `USERS` 中。具体格式为

```
PASSWORD
mypassword
FRIENDS
friend1:status1 friend2:status2 ...
```

其中`status` 表示好友状态，具体可能的取值为：

```
friend status:
  0    -为好友
  1    -好友在线
  2    -好友不在线
  3    -发出好友申请但尚未通过
  5    -发出好友申请但不通过
  6    -好友被删
```

4. GUI与Client通讯

GUI与Client通讯有2种方式，一是GUI主动向Client发送消息，二是Client接收到Server的消息后向GUI发送消息。

Client向GUI发送消息

Client接收到Server传递的信息后，会通过SendMessage函数，向主窗口发送信号，主窗口接收到信号之后会在本地进行相应的操作，所有的信号如下：

```
WM_USER 之后的值用于给开发者自定义信号

WM_USERJOIN          EQU WM_USER + 1    -有用户登录进入大厅
WM_USERLEAVE         EQU WM_USER + 2    -有用户下线离开大厅
WM_APPENDFRIEND      EQU WM_USER + 3    -向好友列表添加好友
WM_CHANGEFRISTATUS   EQU WM_USER + 4    -好友状态变化
WM_APPENDROOMMSG     EQU WM_USER + 5    -Client接收到发送到大厅的消息
WM_APPEND1TO1MSG     EQU WM_USER + 6    -Client接收到私聊消息
```

GUI向Client发送消息

GUI要主动发起请求时，会主动调用Client中的函数，向Server发送信息，同时在本地进行相应操作，所有可能的通信如下。

- 注册：GUI拿到输入框中的IP地址、端口号、用户名、密码后，使用Client中的函数向Server发起注册请求。
- 登录：GUI拿到输入框中的IP地址、端口号、用户名、密码后，使用Client中的函数向Server发起登录请求。
- 添加好友：选中大厅的用户，并点击添加好友按钮，会使用Client中的函数向Server提交添加好友的请求。
- 删除好友：选中好友列表的用户，并点击添加好友按钮，会使用Client中的函数向Server提交删除好友的请求。
- 发送信息：点击发送按钮后，会使用Client中的函数向Server发送对应的聊天信息。

5. 富文本编辑器、阅览器

富文本编辑器与阅览器是利用RichEdit这一控件实现的。这一部分的实现有两个内容：一，实现富文本的输入与显示；二，实现富文本在不同控件之间的传输。

第一，实现富文本的输入与显示。首先，我们利用RichEdit提供的RTF格式，成功地在聊天窗口里实现了对图片、表格等富文本的输入与显示。其次，我们设计了一系列修改文本样式的按钮（例如加粗（B）、下划线（U）等），然后在逻代码辑上将它们与RichEdit连接起来，利用RichEdit的EM_SETCHARFORMAT消息，实现选择文本后点击按钮修改样式的功能，即在聊天窗口里实现了对多样式文本的输入与显示。

第二，实现富文本在不同控件之间的传输。传输文本的一般思路是利用GetWindowText与SetWindowText函数来实现，但是在传输富文本的过程中，这种思路既不能将富文本的格式传输出去，在传输图片等时候也有可能出现传输不完整、传输中断等情况。因此，我们利用RichEdit提供的EM_STREAMIN和EM_STREAMOUT消息，在发送富文本和接收富文本的两个控件之间实现对富文本的分段传输，以完成多样式文本与大富文本的传输。在输入框输入完成后，点击Send按钮，即可将信息发送出去。点击Clear按钮，可以清空当前输入框里的内容。

三、难点与创新点

1、好友逻辑复杂

在本程序中，添加/删除好友的逻辑较为复杂。好友共有6个状态，用来处理上线/下线/等待申请/申请拒绝/申请通过/被对方删除 等逻辑。

添加好友需要对方同意，对方同意后，同意的信息又要返回到申请方，客户端和服务器之间需要进行多次通信来改变服务器以及两个客户端中的数据状态。此外，当好友上线\下线时需要广播给在线好友，以便其及时更新好友状态。

2、群聊功能

本程序额外实现了群聊功能，用户可以看到所有当前在线的人，并和他们进行群聊，同时也可以添加他们为好友。这种添加好友的方法比起通过用户名添加更加方便。

3、富文本编辑器

本程序实现了富文本编辑的功能，可以用方便的按钮工具栏，对输入的文字进行加粗/下划线/斜体/字体颜色/字体样式/字体大小/背景色等设置，并且能够发送和显示。可以直接拖拽图片、表格等进入文本框进行发送。

四、小组分工

我们将本次作业的重点分为GUI、网络通讯和富文本编辑器\阅览器三个部分。采用前后端分离的原则进行敏捷开发，使用Github进行协作。具体分工为：

- **李祁：**客户端和服务端的通讯，服务端的数据存储、读取部分
- **徐亦豪：**GUI部分
- **丁一峰：**富文本编辑器\阅览器

五、实验小结

- **李祁：**

  我主要负责客户端和服务端的通讯与存储功能。最开始做的时候思路比较乱，结果越写逻辑越复杂耦合，难以梳理。后来重新整理了思路，先提前整理好了通讯协议和数据存储格式，并将其定义在头文件中；将每一个命令划分成一个独立的函数进行处理，尽量使用函数内部的临时变量。这样的开发方式让我思路非常清晰，极大的提升了我的开发效率。

开发所遇到的另一个难点是使用汇编进行网络编程比较难以测试，写完一个函数之后没有像GUI一样的直观反馈。因此，我在开发时尽量解耦进行测试，比如服务端的数据读取\存储功能与通讯功能不耦合，就先完成文件读写功能并测试其读写逻辑是否正确。在完成了Server端的逻辑之后先使用Python的socket进行测试。

使用汇编这种比较底层的语言进行开发还是遇到了很多困难的，比如说有时候本来就是指针但是又取了指针的地址；比如说clients[10]这种写法不是取到第10个client而是地址位clients+10；比如有时候eax的值还要使用，但是调用了其他函数之后eax的值却已经被改变了。通过本次的汇编大作业，我对底层的代码实现逻辑有了更加深刻的理解，也更加明白如何写出bug-free的代码。

- **徐亦豪：**

  我主要负责GUI中除富文本编辑器之外的功能，注册、登录、大厅与好友列表的显示和更新，聊天窗口的切换等。

  在我做的时候，我主要大部分时间都花在用户列表上。聊天软件需要频繁地添加/更新用户列表信息，这需要我去通过某个文本，有时是行数来找到列表中对应的行，汇编中查看这些信息不能像其他语言一样简单的通过一个.xxx来访问，需要使用SendMessage加上一些值和参数来访问，这就需要我去查阅微软提供的文档，但是这些文档都是十几年前的了，有些写的很暧昧，使我不知道究竟参数该填写什么，填写什么类型，在这上面我费了很多功夫。经过多次试错与经验总结，我终于写完了我的这一部分。

  这次作业对我对汇编语言的运用能力提出了很大的挑战，经过这次大作业，我对汇编语言的应用水平得到了很大的提升。对于一个Win32程序究竟是如何运行、通信的，程序中大量出现的SendMessage究竟是起什么作用，我有了一个比较完备的了解。

- **丁一峰：**

  在GUI部分里，富文本编辑器\阅览器是难度较高的一块，原因有二：一，虽然RichEdit控件提供了修改文本样式的消息，但是由于文档解释不够清晰、参数结构较为复杂，在初期学习与调用的过程中遇到了许多困难；二，由于富文本的特殊性（样式多、体积大），使用RichEdit控件传输富文本时，需要利用类似于流传输的思路来实现，这让我在实现信息的发送与实现时遇到了一些困难。除此之外，由于网络上有关汇编语言编程的信息相对较少、年代相对久远，在很多时候，我需要在没有任何信息帮助的情况下独自debug，这也给开发带来了不小的困难。幸运的是，我们在开发初期的调研里找到了一本名为《琢石成器—Windows环境下32位汇编语言程序设计》的教材。在阅读与学习这本书的过程中，我们逐渐理解并构建起了整个GUI的框架，也学会了在程序里使用各类控件。