基于流式套接字实现多人聊天室

学号：16307130194，姓名：陈中钰

1 项目概况

项目使用 Python 3.7.5 完成，基于 socket 的流式套接字（SOCK.STREAM）、PyQt 的前端框架实现了一个多人聊天室，允许多个用户登陆、加入群组并进行群聊。项目具有以下的特点：

功能齐全 可以通过对应的按钮，进行登入、登出、创建/加入群组、离开群组、发送消息的操作。此外，可以通过用户框查看登陆的用户名，可以通过群组成员框看到群组成员列表，还可以通过消息框显示操作是否成功、群组发送的消息。另外，不仅群组中可以多人同时聊天，而且可以同时存在多个不同的群组，群组之间运行不会相互影响。

协议设计 仿照 HTTP 协议设计了 THTTP 协议 (Trivial Hypertext Transfer Protocol)，通过 Python 面向对象编程实现协议相关的构造函数、序列化函数和解析函数，还实现了协议格式的检查函数。

完整的错误控制 实现了完整的错误控制，比如控制了在没有登入、不在群组中时不可以发送消息，又比如处理了在客户程序关闭时要退出群组和退出登陆的操作。

简洁美观的界面 通过 PyQt 实现了一个简洁易用的界面，可以显示登陆用户名、群组名、群组成员列表、操作反馈信息、群组信息等，可以通过输入框输入想发送的消息，通过对应按钮实现登入、登出、创建/加入群组、离开群组、发送消息的操作。

2 项目架构

2.1 Thread Pool 架构

多人聊天室通过 Thread Pool 来管理线程，架构如 Figure 1所示。当用户尝试与服务器连接时，服务器都会建立线程为用户服务。

2.2 Client/Server 架构

多人聊天室的（单个）用户和服务器之间的关系如 Figure 2所示。

3 THTTP 协议

3.1 THTTP 协议格式

THTTP 协议 (Trivial Hypertext Transfer Protocol) 是仿照 HTTP 协议设计的，主要格式和 HTTP 协议一致，请求协议格式如 Figure 3所示，应答协议格式如 Figure 4所示。另外，从 socket 接收缓冲区取出的字符串可能同时包含多条消息，为了能够划分缓冲区中的多条消息，在协议开头添加 3 位的 length 字段，表示协议剩下部分的长度。通过读取前 3 位可以获得协议长度，再读取出协议，这样可以把多条消息分开。

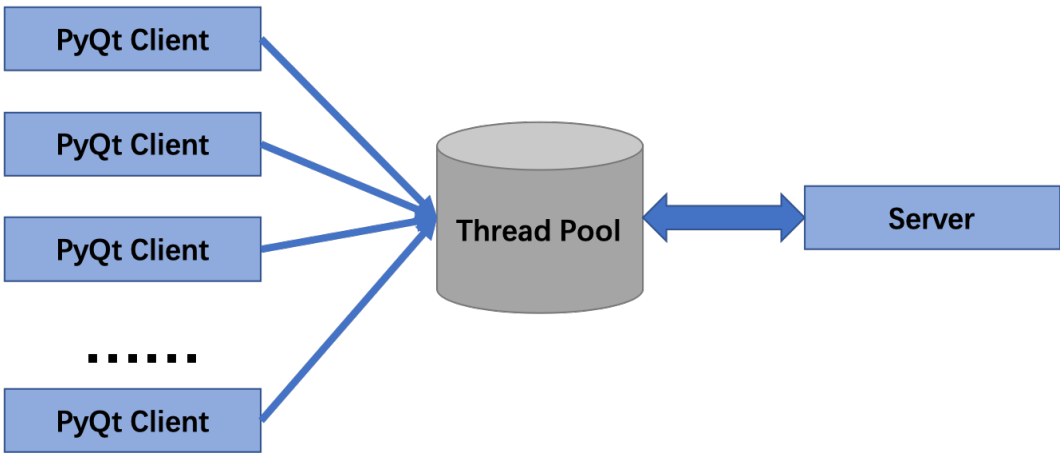

Figure 1: 多人聊天室 Thread Pool 架构

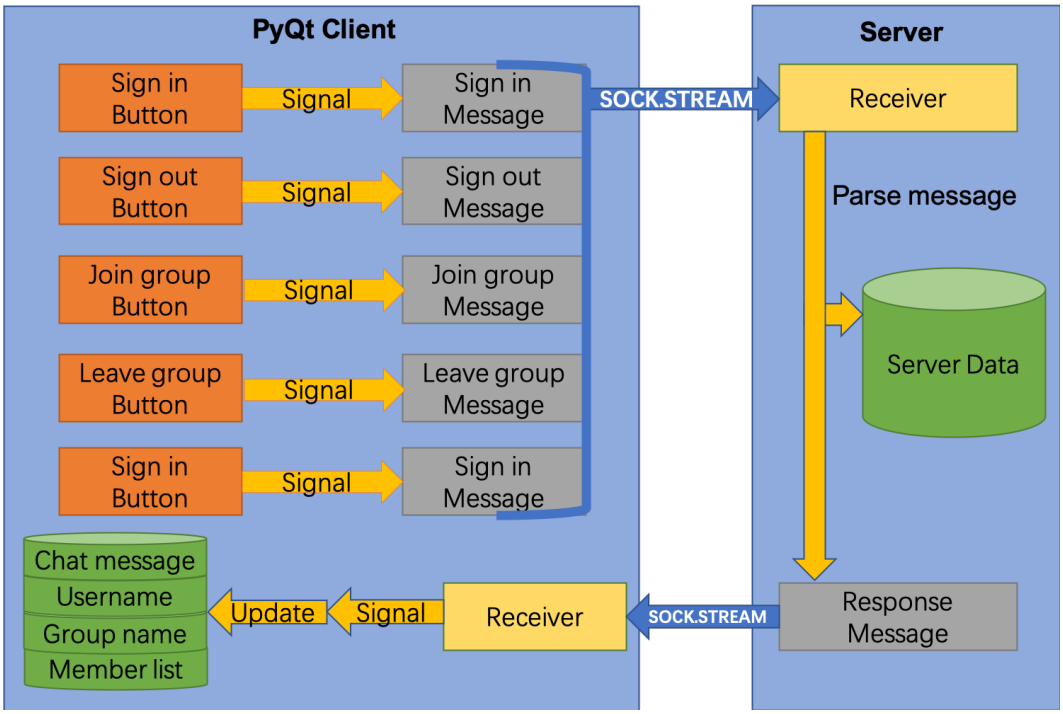

Figure 2: 多人聊天室 Client/Server 架构

3.2 THTTP 协议消息

项目中所用到的 THTTP 协议的请求消息列表如 Table 1所示，应答消息列表如 Table 2所示。

4 代码实现

4.1 文件解释

项目代码文件的对应解析如 Tabel 3所示。

4.2 config.py 实现

文件定义了参数的设置，如 Table 4所示。

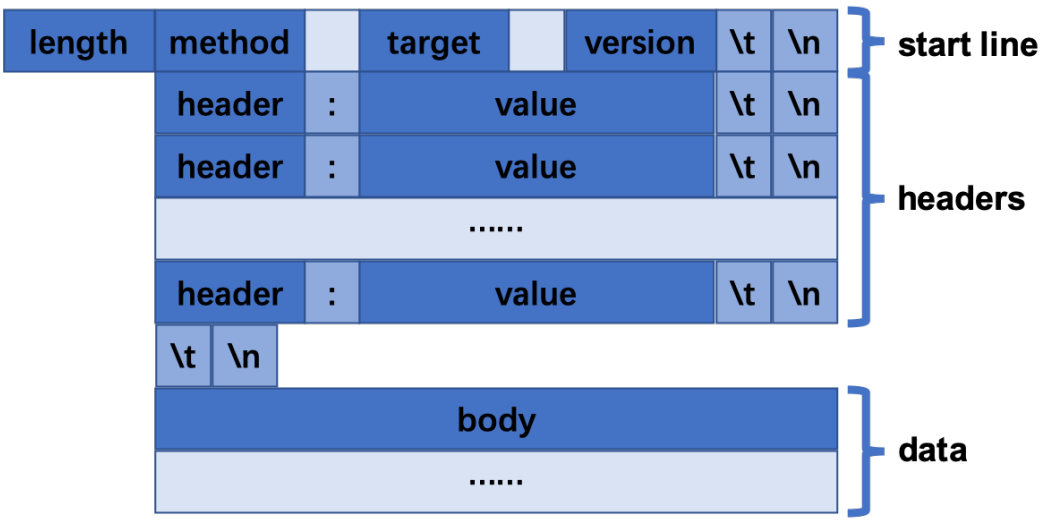

Figure 3: THTTP Request 格式

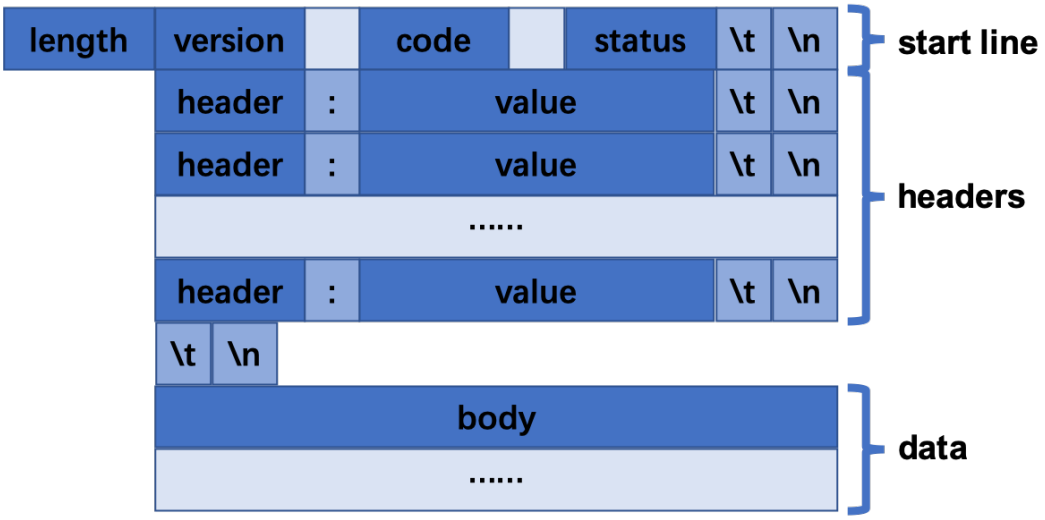

Figure 4: THTTP Response 格式

4.3 protocol.py 实现

使用面向对象编程，实现了 THTTP_Request 和 THTTP_Response 类，两个的共同部分继承于 _THTTP 类。类中记录了协议的 start line、headers、body。另外重构了 __repr__() 的协议序列化方法，用来序列化为可以发送的消息字符串，还重构了 __str__() 的字符串化方法，用来在客户、服务器的控制台中打印。对应地还实现了字符串消息的解析方法，以及分割多条消息的函数。另外，还是实现了消息格式正确性的检查方法。

Table 1: THTTP Request

method	body
signin	username
signout	-
join	group name
leave	-
send	发送的消息内容

Table 2: THTTP Response

status code	status line	body
200	Sign in success	Sign in as .
201	Sign out success	Sign out successful.
202	Join group success	Join group .
203	Leave group success	Leave group successful.
204	Group members	group member list
205	Send chat success	-
300	Sign in error	Connection not exist./Already signed in./Username collision.
301	Sign out error	Connection not exist./Not signed in./Leave group first.
302	Join group error	Connection not exist./Not signed in./Already in a group.
303	Leave group error	Connection not exist./Not signed in./Not in a group.
305	Send chat error	Connection not exist./Not signed in./Not in a group.
400	Wrong format request	Corrupted request!

Table 3: 文件解释

filename	description
config.py	全局参数
requirements.txt	项目使用的 Python 库列表
protocol.py	THTTP 协议
mainwindow.ui	Qt 的 ui 文件
mainwindow.py	用 Qt 的 ui 文件转换的 PyQt 文件
client.py	客户
server.py	服务器

4.4 mainwindow.py 实现

使用 Qt 编辑 mainwindow.ui 文件, 构建出前端的部件, 然后使用 pyuic5 命令把 mainwindow.ui 转化为 mainwindow.py 文件中的 Ui_MainWindow 类，使得可以用 PyQt 运行前端。

4.5 client.py 实现

构建 MyWindow 类，继承 mainwindow.py 文件中的 Ui_MainWindow 前端类。把 5 个按钮的 clicked 操作分别绑定到对应的类操作函数：Sign In 按钮的 clicked 操作绑定到 sign_in() 类函数，使得按钮被按下的时候触发这个函数，函数中启动用户名输入框，获取用户输入的用户名，封装好登陆请求，发送请求；Sign Out 按钮的 clicked 操作绑定到 sign_out() 类函数，函数中封装好登出请求并发送；Join 按钮的 clicked 操作绑定到 join() 函数，函数中启动群名输入框，获取用户输入的群名，封装好加入群组请求并发送；Leave 按钮的 clicked 操作绑定到 leave() 函数，函数中封装退出群请求并发送；Send 按钮的 clicked 操作绑定到 send() 函数，函数获取输入框中的文本，封装发送消息请求并发送。

构建 Receiver 类，继承 QThread 类。在类的运行 run() 函数中，不断尝试从接收缓冲区中取出消息。获得应答消息后，对应答消息进行解析，并按照 status code 来触发对应的信号。MyWindow 类中还使用了 Receiver 类，而且在 MyWindow 类中还要绑定 Receiver 的信号到对应的函数。那么当 Receiver 线程接收到消息后，会根据 status code 触发对应的信号，而信

Table 4: 参数设置

config	value	description
HOST	'127.0.0.1'	IP 地址
PORT	65432	IP 端口
BUFSIZE	1024	接收缓冲区大小
MAX_THREAD	1000	客户线程数量限制
LENGTH_SIZE	3	THTTP 协议头部的长度字段的位数

号会触发对应的函数。通过这个方法，是的客户可以和请求操作并发地接受应答消息，并根据应答消息在前端显示用户名的更改、群名的更改、群成员列表的更改、接收的信息。

另外，在消息框中显示的消息是一条条添加进去的，历史消息是保留的，因此每次在消息框中添加消息的时候，需要取得整个消息框的内容，在内容后添加新的消息，再把消息框设置为新构造的内容。用户名框、群组名框、群组成员列表框就直接设置为新的值就可以了。

4.6 server.py 实现

服务器用 ThreadManger 类和 ThreadPoolManger 类来实现线程池，每当有一个新的用户建立连接的时候，就创建新线程来为新用户服务，进入 handle_request() 函数的 while 循环中，不断尝试接受用户发送的请求消息，进行解析，并执行对应的服务。此外，还实现了 CharRoom 类，用来记录用户的信息，并实现了类函数接口来允许服务器进行登入、登出、加入群组、退出群组、发送消息、获得群组成员的操作。

每个用户都有一个线程在运行 handle_request() 函数，函数中不断尝试接受用户发送的请求消息，并进行解析。若用户的请求格式错误，则返回 400 消息（status code 对应的含义请看 Table 2）。若用户的请求是登入或登出，则通过 ChatRoom 提供的类方法来进行登入或登出操作，并发送对应的应答消息。若用户的请求是加入群组，则要把用户加入群组，并向用户发送加入成功的消息，以及向群组中所有成员发送新的群组成员列表消息。如用户的请求是离开群组，则要把用户从群组中删除，并向用户返回退出成功的消息，以及向群组中的其他成员发送新的群组成员列表消息。如用户的请求是发送消息，则要把发送的消息组播给群组中的全部成员。

另外，当用户因为关闭程序或其他原因导致连接断开时，在 ChatRoom 类中仍然有断开用户的登录信息和群组信息，这时候还需要为用户退出群组、退出登陆操作，以保证数据的一致性。

此外，服务器可能范围的错误消息内容有多种，具体请看 Table 2中的错误消息，也就是以 3 开头和 4 开头的消息。

5 详细功能

登陆 点击 Sign In 按钮后会弹出输入框，要求输入登录的用户名，点击确定。如果没有错误，则消息框中显示登陆成功，且用户名框中显示用户名；如果出现错误，则在消息框中显示对应的错误原因。

登出 点击 Sign Out 按钮，如果没有错误，则消息框中显示登出成功，且用户名框清空；如果有错误，则消息框中显示对应的错误原因。

加入群组 点击 Join 按钮后会弹出输入框，要求输入加入的群组名称，点击确定。如果没有错误，则消息框中显示加入成功、当前的群组成员列表，且群组名框中显示群组名、群组成员列表框中显示群组成员列表，同时，其他群组成员的消息框中也会显示新的群组成员列表，而且群组成员列表框中显示的群组成员列表也会更新；如果有错误，则消息框中显示对应的错误原因。

离开群组 点击 Leave 按钮后，如果没有错误，则消息框中显示退出成功，同时，其他群组成员的消息框中也会显示新的群组成员列表，而且群组成员列表框中显示的群组成员列表也会更新；如果有错误，则消息框中显示对应的错误原因。

发送消息 在文本输入框中输入要发送的消息，点击 Send 按钮，如果没有错误，则消息框中会显示发送消息的用户名以及对应的消息内容，而且群组中的其他成员的消息框中也会同时显示，发送成功后，文本输入框清空；如果有错误，则消息框中会显示对应的错误消息。

6 运行

6.1 运行方法

代码布置 修改 config.py 中的 HOST 参数为服务器 IP 地址，代码在服务器电脑、客户电脑上各放置一份。

客户电脑安装项目需要的 Python 库　pip install -r requirements.txt

在服务器电脑上运行服务器　python server.py

在客户电脑上运行一个客户　python client.py

在客户电脑上运行多个客户　多次执行上述命令皆可。

6.2　运行结果

3 个人在 group 801、2 个人在 group 802、1 个人在 group 803 聊天的运行结果如 Figure 7所示。点击 Sign In 或 Join 按钮后，会弹出输入框，要求输入登录的用户名或加入的群组名，如 Figure 5和6所示。

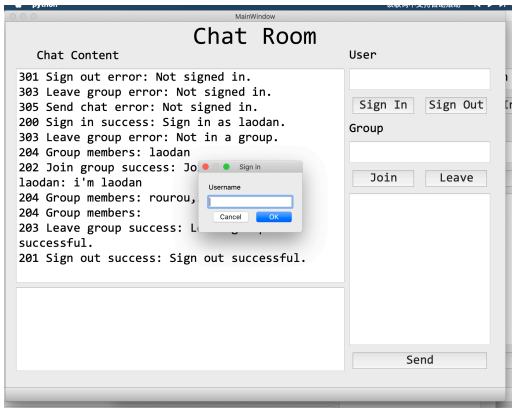

Figure 5: 多人聊天室登陆操作

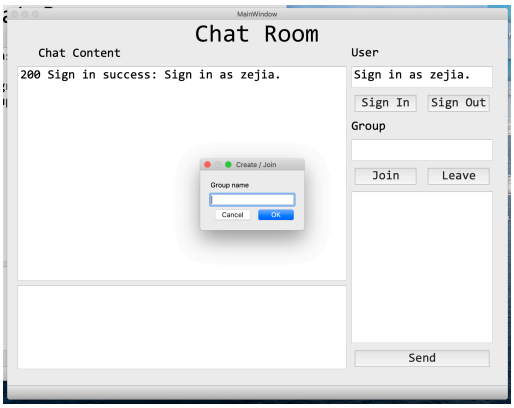

Figure 6: 多人聊天室加入群组操作

Figure 7: 多人聊天室运行结果展示