Portland State University

Internet Relay Chat Project

Team Members: Akhila Chiluveru, Siddhi Mittal, Deepa Hegde.

TABLE OF CONTENT

1. Introduction

This IRC project is implemented using protocols that implement multi-client communication which allows users to create, join and leave a room. Besides this, users can also send a personal message to another client.
In a case with multiple clients in a room, all the messages are displayed to everyone participating in the room.

1.1 Server and Client

Server and client are the heart of the application. We have chosen a single server to which any number of clients can connect and have their own name associated with them which allows alphanumerical characters along with underscore.

2. Specifications

2.1 Communication

The mode of communication is through rooms and personal chats. When a user sends a message in the room, the server receives it and forwards it to all the users who are part of that room excluding the sender.

2.2 Messages

The important parts of any message are the command's name, transaction ID, and payload. These parts are separated by the delimiter which is the space. Here the transaction ID is an unsigned integer that increments by 1 whenever a message is sent by the client. However, it remains 0 for the server to client messages.

2.3 Replies

Messages sent to the server generate a reply in numeric form. It consists of the transaction ID and the status of the request. It represents a 0 for a successful request and a non-zero for any errors.

3. Infrastructure

3.1 Menu

When a client successfully connects, they have presented an option to type their name, after it is done a menu is displayed that shows possible actions which can be performed. Users can choose any of the options from that list.

```
C:\Users\siddh\Desktop\Winter 2022\DSA project\InternetRelayChat\InternetRelayChat>python client.py
Enter your nickname: Mary

YAY! Connected to the server!


Application Menu:
1.menu (lists the menu)
2.list (lists all the available rooms)
3.create roomname - like "create room1" (creates a new room)
4.join roomname - like "join room1" (joins the room)
5.switch roomname - like "switch room2" (switches the room)
6.leave (leave from the room)
7.personal name message - like "personal nickname hello" (sends personal message)
8.exit (exits the client app)
```

3.2. Create room

No rooms are available initially. The first user can create a room using create command. This created room will now be visible to all the other users to join and can start communicating using the room name.

```
create maryroom
maryroom created

create maryroom2
maryroom2 created
```

3.3. Join a room

Once the rooms are available, any client can see the available rooms and join a room of their choice using the join command. Right after establishing communication, all the clients in the room will be known of the incoming client and can start communication.

```
join maryroom
[maryroom] gold joined the room
[maryroom] , gold Welcome to the room
```

3.4 Leave the room

A user can leave a room at any point in time by using the leave command.
If there are other users, the room is still available for other users without any disruptions else if the last person is leaving the room, rooms get deleted as nobody is in the room anymore.

```
leave maryroom
You left the room
```

3.5 List the rooms

A user can view the list of rooms with the command list. The user can join a room using the name of the room using the join command.
If no rooms are available, it displays “No rooms available” else a list of available rooms are displayed.

```
list
No rooms available
```

```
list
List of available roomdetails:

maryroom

maryroom2
```

3.6 Switch to another room

A user can switch from one room to another room using the switch command.

```
join maryroom
[maryroom] gold joined the room
[maryroom] , gold Welcome to the room
join maryroom1
[maryroom1] gold joined the room
[maryroom1] , gold Welcome to the room
switch maryroom
Switched to maryroom
```

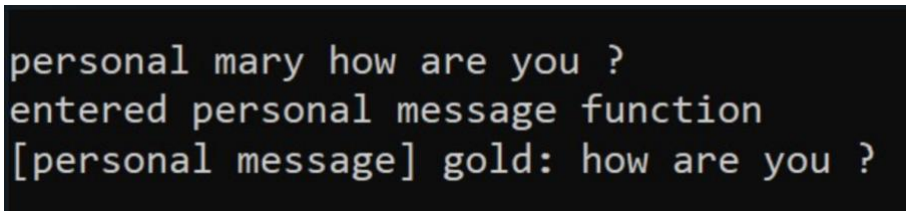
3.7 Exit from server

A client exits from a connection using the exit command. The client terminal is simply closed. If this user is a part of any room, the user will be notified of the same. Besides this, the server also knows about this forceful disconnection.

4. IRC Concepts

4.1 One-to-one communication

In this type of communication, client and server communicates in the form of personal communication that is not visible to others. This is achieved by using the "personal user_name message" command
Sender:



Receiver:

4.2 One-to-many communication

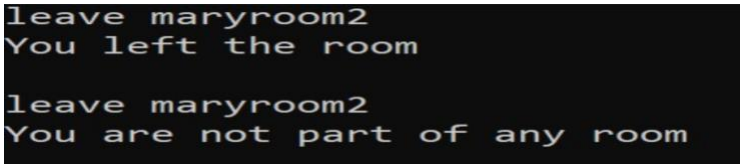
A room is created, multiple users join that room and start communicating with others through messages using which are viewed by all the other users who are also a part of that room. This is one-to-many communication as a message sent by one person is delivered to many people in the group.

4.3 One-to-all communication

A client can send a broadcast message that will be delivered to each client and server.

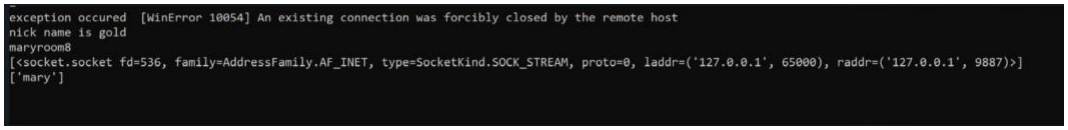
5. Error Handling

a) If a user tries to leave a room that he is not a part of 'You are not part of any room' message will be displayed.



b) If a client is crashed, servers, rooms, and clients are notified.

Server:



Rooms:

```
[maryroom8] gold left the room
```

c) If the server is crashed, all the clients are logged out and notified with a server not available message.

```
Server not responding
```

d) If a client tries to establish a connection with another client who doesn't exist or an incorrect name is used, the "User not found" message is shown.

```
personal prabhu Hi, lets catchup
entered personal message function
User not found
```

e) If a user tries to switch to the room in which the user is already in, it indicates the user saying the same.

```
switch maryroom
You are already in the room, choose another available room to change
```

6. Conclusion & Future Work

This application can be furthermore enhanced by implementing features implementing security features and media sharing options. The message sharing has been extended to a single server and multiple clients either in the form of broadcast messages or private messages.