Python Chat Server Project Report

Emmanuel Perez

Panther ID: 4132644

1. Introduction

For this project, I faced the problem of learning some Python rules as I made a server program that I had to learn some basic understanding of as well. I had to read the “Internet Relay Chat Protocol” and use https://stackoverflow.com to understand the basics of servers, their commands and basic python implementations. The final product, while is not as neat as desired, still ended up following most of IRC protocols.

2. Problem Statement

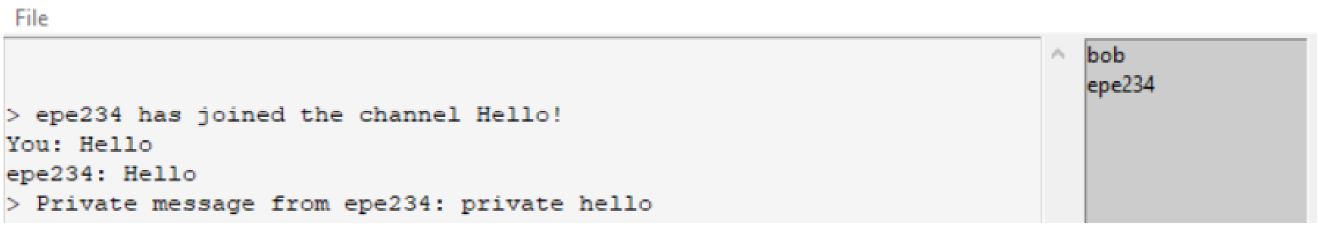
I needed to create an IRC chat server in python, despite getting a basic chat server program to work with, I had to understand the basics of how an RTC chat server works and what the basic program was and what was not implementing. Multiple commands required me to change the basic program a bit to implement the RTC server rules. Users didn’t have the variables necessary to implement all IRC protocols, Main and Channel also required a bit of tweaking, but ChatServer needed major change. Also implementing multiple channels proved to be very problematic since it required changing previous code as well as the skeleton code even more to match.

3. Methodology

I solved this problems by first implementing a command at a time and checking what types of changes where required to follow the protocol. Such as implementing the /mode command required me to add mode variables to Users and Channels to be check according to their mode, /silence command required me to change channel’s broadcasting to only broadcast messages to socket of users that didn’t have the sender in the ignore list, etc. Some commands had to be done together to work properly such as /away and /privmsg or the /ping /pong commands.

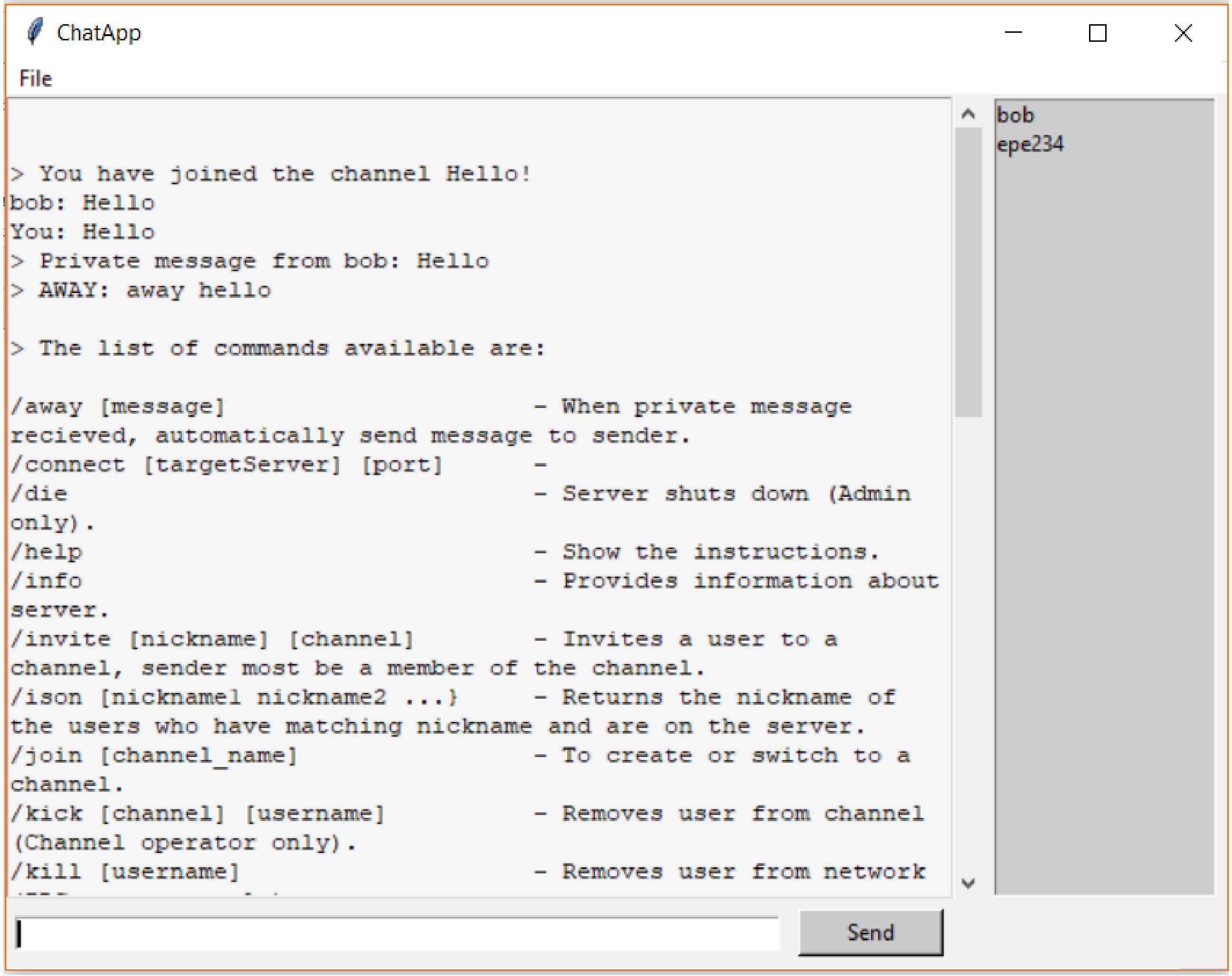
4. Results

The results of the project is a chat server that simulates a number of the of the RTC protocol, with some change in certain protocol, the only protocol that couldn’t be implemented at the end was the connect command to change to other servers. For implementing the multi-channel part I was successful with commands usage to traverse the channels. I also have spotted some display glitches with the usernames, but couldn’t completely fix them.



```
> epe234 has joined the channel Hello!
You: Hello
epe234: Hello
> Private message from epe234: private hello
```

Here is some sample of the program where two clients on the same server and same channel (Hello) are communicating and testing the /away and /privmsg calls. The other user also used the /help command, but on a small client window the description get warped around which makes it hard to read sometimes.



```
> You have joined the channel Hello!
bob: Hello
You: Hello
> Private message from bob: Hello
> AWAY: away hello

> The list of commands available are:

/away [message]                      - When private message
recieved, automatically send message to sender.
/connect [targetServer] [port]       -
/die                                 - Server shuts down (Admin
only).
/help                                - Show the instructions.
/info                                - Provides information about
server.
/invite [nickname] [channel]         - Invites a user to a
channel, sender most be a member of the channel.
/ison [nickname1 nickname2 ...}      - Returns the nickname of
the users who have matching nickname and are on the server.
/join [channel_name]                 - To create or switch to a
channel.
/kick [channel] [username]           - Removes user from channel
(Channel operator only).
/kill [username]                     - Removes user from network
```

5. Analysis

I learned a lot about how IRC chat servers are run, their rules, and an idea of how they are created. I as well memorized better more basic Python functions and implementations, mainly the for loops and string splitting. Some problems I encounter where mostly with the commands, such as /restart gave me problems to implement because I couldn’t figure out how to add them onto the basic program, while other commands, such /ping and /pong I didn’t know what to do to follow IRC protocols.

6. References

Oikarinen, J. and Reed, D. (1993, May). Internet Relay Chat Protocol. Retrieved from *https://tools.ietf.org/html/rfc1459#section-4.2.8*

https://en.wikipedia.org/wiki/List_of_Internet_Relay_Chat_commands#PART

https://stackoverflow.com