

Final Report

110550162 洪慎廷

Topic: Design and Implement Simple Chat Server

Program Description:

My program consists of three python files “chat_server.py”, “datastructures.py”, “static_messages.py”. The descriptions of these files are below.

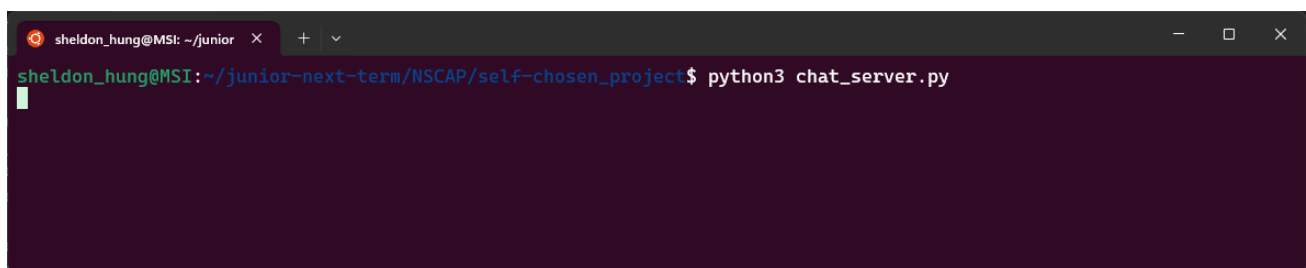
- chat_server.py
 - The main program of the server. It is responsible for handling multiple clients' TCP connection, doing the authentication of each client, and sending and receiving commands or messages to and from the clients.
 - For handling multiple clients. I use a monitor “select” to handle multiple clients.
 - The server interacts with the clients by receiving user command from the clients and send the corresponding message back to the clients. The program will parse the received command and do some actions. The detail of the commands and actions will be described at the next section.
- datastructures.py
 - All the objects that will be used by the main program are defined in here. There are two objects that I defined, “UserInfo” and “ChatRoom”.
 - “UserInfo”: It is an object that stores a client's information, including authentication data (username, password), user profile (age, birthday, job, phone_number, self_description), and some control data of the client.
 - “ChatRoom”: It is an object that stores the information of a chatroom, for example, the name of the chatroom, the owner of the chatroom, is the chatroom private or public, who can or cannot enter the chatroom. Also, it has some functions for handling user commands that may modify the chatroom's data.

- `static_messages.py`
 - Just simply stores some static string that may be sent to the users.

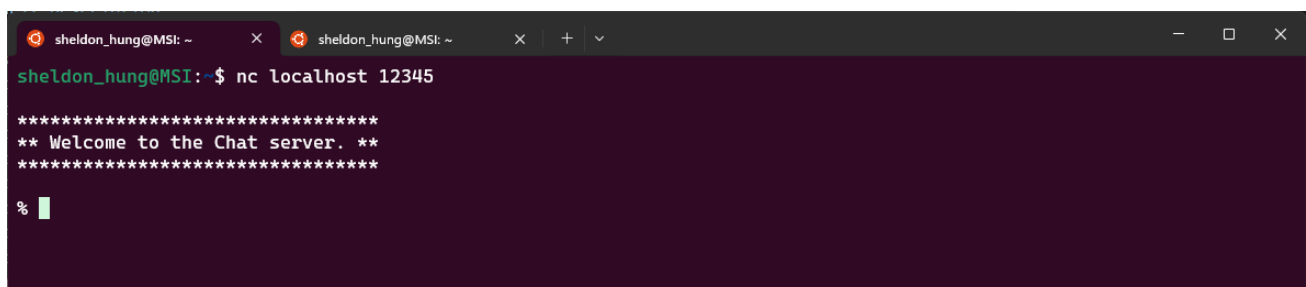
Final Results:

Below, I will display the final results by running the server program and open multiple bashes to play as the clients and interact with the server. Additionally, I will describe the commands and messages the clients send and the respond from the server.

- Run the server and open two bashes playing as two clients (Tom and Bob), connecting to the server. The server will show a welcome message once a client is connected.



```
sheldon_hung@MSI: ~/junior
sheldon_hung@MSI:~/junior-next-term/NSCAP/self-chosen_project$ python3 chat_server.py
```



```
sheldon_hung@MSI: ~
sheldon_hung@MSI:~$ nc localhost 12345

*****
** Welcome to the Chat server. **
*****

%
```



```
sheldon_hung@MSI: ~
sheldon_hung@MSI:~$ nc localhost 12345

*****
** Welcome to the Chat server. **
*****

%
```

- Supplement: Type the command “help” to see all the commands the server supports. For input any invalid or authorized command, the server will send the correspond error messages.

```
sheldon_hung@MSI: ~  
sheldon_hung@MSI: ~  
sheldon_hung@MSI: ~  
sheldon_hung@MSI:~$ nc localhost 12345  
  
*****  
** Welcome to the Chat server. **  
*****  
  
% help  
-----  
Supported Commands:  
register <username> <password>  
login <username> <password>  
logout  
whoami  
set-profile <attribute> <value>  
view-profile <username>  
list-user  
open-chatroom <chatroom name>  
list-chatroom  
enter-chatroom <chatroom name>  
close-chatroom <chatroom name>  
help  
-----
```

- The two clients (Tom and Bob) register an account with password and login the account, using the command “register {username} {password}” and “login {username} {password}”.

```
sheldon_hung@MSI: ~  
sheldon_hung@MSI: ~  
sheldon_hung@MSI: ~  
sheldon_hung@MSI:~$ nc localhost 12345  
  
*****  
** Welcome to the Chat server. **  
*****  
  
% register Tom 420420  
Register successfully.  
% login Tom 420420  
Welcome, Tom.  
%  
█
```

```
sheldon_hung@MSI: ~  
sheldon_hung@MSI: ~  
sheldon_hung@MSI: ~  
sheldon_hung@MSI:~$ nc localhost 12345  
  
*****  
** Welcome to the Chat server. **  
*****  
  
% register Bob 777777  
Register successfully.  
% login Bob 777777  
Welcome, Bob.  
%  
█
```

- Tom uses the command “set-profile {attribute} {value}” to set his own profile.

```
sheldon_hung@MSI: ~$ nc localhost 12345

*****
** Welcome to the Chat server. **
*****

% register Tom 420420
Register successfully.
% login Tom 420420
Welcome, Tom.
% set-profile age 20
Attribute age set successfully.
% set-profile birthday 2004/8/20
Attribute birthday set successfully.
% set-profile job student
Attribute job set successfully.
% set-profile phone_number 0905-123-456
Attribute phone_number set successfully.
% 
```

- Bob uses the command "view-profile {username}" to view Tom's profile.

```
sheldon_hung@MSI: ~$ nc localhost 12345

*****
** Welcome to the Chat server. **
*****

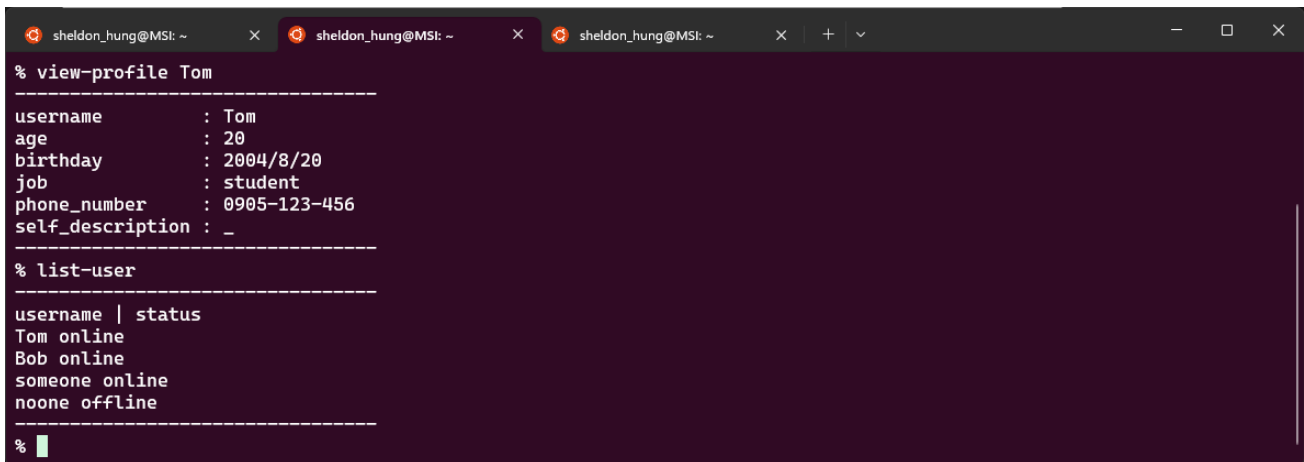
% register Bob 777777
Register successfully.
% login Bob 777777
Welcome, Bob.
% view-profile Tom
-----
username      : Tom
age           : 20
birthday      : 2004/8/20
job           : student
phone_number  : 0905-123-456
self_description : _
-----
% 
```

- Tom can also view his own profile by using the command “whoami”.

```
sheldon_hung@MSI: ~$ nc localhost 12345

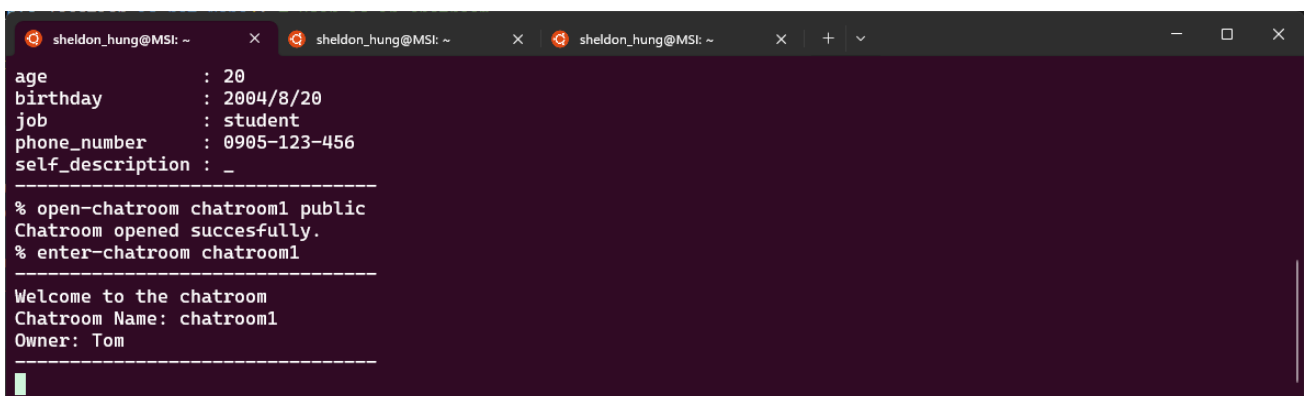
Attribute birthday set successfully.
% set-profile job student
Attribute job set successfully.
% set-profile phone_number 0905-123-456
Attribute phone_number set successfully.
% whoami
-----
username      : Tom
age           : 20
birthday      : 2004/8/20
job           : student
phone_number  : 0905-123-456
self_description : _
-----
% 
```

- Bob uses the command “list-user” to view all the users in the server. The server will print all the username and if they are online or offline.



```
sheldon_hung@MSI: ~  
% view-profile Tom  
-----  
username      : Tom  
age           : 20  
birthday      : 2004/8/20  
job           : student  
phone_number  : 0905-123-456  
self_description : _  
-----  
% list-user  
-----  
username | status  
Tom online  
Bob online  
someone online  
noone offline  
-----  
% 
```

- Tom opens a public chatroom by using the command “open-chatroom {chatroom name} {public or private}”. A public chatroom means that anyone can enter in the chatroom if they aren’t in the blacklist. Then Tom enters the chatroom by using the command “enter-chatroom {chatroom name}”.



```
sheldon_hung@MSI: ~  
age           : 20  
birthday      : 2004/8/20  
job           : student  
phone_number  : 0905-123-456  
self_description : _  
-----  
% open-chatroom chatroom1 public  
Chatroom opened successfully.  
% enter-chatroom chatroom1  
-----  
Welcome to the chatroom  
Chatroom Name: chatroom1  
Owner: Tom  
-----  
% 
```

- Bob uses the command “list-chatroom” to view all the chatrooms in the server. The server will print out the chatroom name, chatroom owner, and if the chatroom is public or private. Then use the command “enter-chatroom {chatroom name}” to enter Tom’s chatroom.

```
sheldon_hung@MSI: ~  
-----  
% list-chatroom  
-----  
chatroom name | owner | authority  
chatroom1 | Tom | public  
-----  
% enter-chatroom chatroom1  
-----  
Welcome to the chatroom  
Chatroom Name: chatroom1  
Owner: Tom  
-----
```

- Supplement: Type the command “/help” to see all the chatroom commands the server supports. In the chatroom, commands must start with a prefix “/” to let the server know the client is sending a command, or the sever will view it as a chatting message.

```
sheldon_hung@MSI: ~  
-----  
% enter-chatroom some_other_chatrooms  
-----  
Welcome to the chatroom  
Chatroom Name: some_other_chatrooms  
Owner: someone  
-----  
/help  
-----  
Chatroom Commands:  
/exit-chatroom  
/list-user  
/get-whitelist  
/get-blacklist  
/kick-user <username>  
/ban-user <username>  
/permit-user <username>  
/help  
-----
```

- When Bob enters Tom’s chatroom, the server will inform all the other users (Tom) in the chatroom that Bob enters the chatroom. Then Tom and Bob can start chatting by just typing the chatting message.

```
sheldon_hung@MSI: ~  
-----  
[System]: Bob had enter the chatroom  
Hello  
<Tom>: Hello  
<Bob>: Hi  
Some chatting message #1  
<Tom>: Some chatting message #1  
<Bob>: Some chatting message #2  
<Bob>: Some chatting message #3  
Some chatting message #4  
<Tom>: Some chatting message #4  
<Bob>: Goodbye.  
-----
```

```
sheldon_hung@MSI: ~
-----
<Tom>: Hello
Hi
<Bob>: Hi
<Tom>: Some chatting message #1
Some chatting message #2
<Bob>: Some chatting message #2
Some chatting message #3
<Bob>: Some chatting message #3
<Tom>: Some chatting message #4
Goodbye.
<Bob>: Goodbye.
```

- Bob uses the chatroom command “/exit-chatroom” to leave Tom’s chatroom. The server will inform all the other users (Tom) in the chatroom that Bob leaves the chatroom.

```
sheldon_hung@MSI: ~
<Bob>: Goodbye.
/exit-chatroom
[System]: Bob had left the chatroom
%
```

```
sheldon_hung@MSI: ~
<Tom>: Some chatting message #4
<Bob>: Goodbye.
[System]: Bob had left the chatroom
%
```

- Tom adds Bob into the chatroom’s blacklist by using the command “/ban-user {username}”. Then Bob cannot enter Tom’s chatroom.

```
sheldon_hung@MSI: ~
<Tom>: Some chatting message #4
<Bob>: Goodbye.
[System]: Bob had left the chatroom
/ban-user Bob
The user is added to the blacklist successfully.
```

```
sheldon_hung@MSI: ~
<Bob>: Goodbye.
/exit-chatroom
[System]: Bob had left the chatroom
% enter-chatroom chatroom1
You are banned from the chatroom.
%
```

- Tom can view the chatroom’s blacklist by using the command “get-blacklist”.

```
sheldon_hung@MSI: ~  
/get-blacklist  
-----  
Banned Users:  
Bob  
-----  
|
```

- Bob opens a private chatroom by using the command “open-chatroom {chatroom name} {public or private}”. A private chatroom means that only the chatroom owner or user in the whitelist can enter the chatroom. Then Bob enters the chatroom by using the command “enter-chatroom {chatroom name}”.

```
sheldon_hung@MSI: ~  
You are banned from the chatroom.  
% open-chatroom chatroom2 private  
Chatroom opened successfully.  
% enter-chatroom chatroom2  
-----  
Welcome to the chatroom  
Chatroom Name: chatroom2  
Owner: Bob  
-----  
|
```

- Tom leaves his own chatroom and intend to enter Bob’s private chatroom. However, he isn’t in the chatroom’s whitelist. Thus, he can’t enter Bob’s chatroom.

```
sheldon_hung@MSI: ~  
<Tom>: Some chatting message #4  
<Bob>: Goodbye.  
[System]: Bob had left the chatroom  
/ban-user Bob  
The user is added to the blacklist successfully.  
/exit-chatroom  
[System]: Tom had left the chatroom  
% enter-chatroom chatroom2  
You do not have the permission.  
%  
|
```

- Bob adds Tom to the chatroom’s whitelist by using the command “/permit-user {username}”. Then Tom can enter the chatroom.


```
sheldon_hung@MSI: ~  
-----  
Welcome to the chatroom  
Chatroom Name: chatroom2  
Owner: Bob  
-----  
/permit-user Tom  
The user is added to the whitelist successfully.  
[System]: Tom had enter the chatroom  
%  
  
sheldon_hung@MSI: ~  
% enter-chatroom chatroom2  
You do not have the permission.  
% enter-chatroom chatroom2  
-----  
Welcome to the chatroom  
Chatroom Name: chatroom2  
Owner: Bob  
-----  
%
```

- Bob can view the chatroom's whitelist by using the command "get-whitelist".

```
sheldon_hung@MSI: ~  
-----  
/get-whitelist  
-----  
Permitted Users:  
Tom  
-----  
%
```

- Bob and Tom chat in the chatroom and then exit the chatroom.

```
sheldon_hung@MSI: ~  
Owner: Bob  
-----  
Hi  
<Tom>: Hi  
<Bob>: Hello  
<Bob>: some chatting message #1  
<Bob>: some chatting message #2  
some chatting message #3  
<Tom>: some chatting message #3  
Goodbye.  
<Tom>: Goodbye.  
/exit-chatroom  
[System]: Tom had left the chatroom  
%  
  
sheldon_hung@MSI: ~  
[System]: Tom had enter the chatroom  
<Tom>: Hi  
Hello  
<Bob>: Hello  
some chatting message #1  
<Bob>: some chatting message #1  
some chatting message #2  
<Bob>: some chatting message #2  
<Tom>: some chatting message #3  
<Tom>: Goodbye.  
[System]: Tom had left the chatroom  
/exit-chatroom  
[System]: Bob had left the chatroom  
%  
%
```

- Tom and Bob close their own chatroom by using the command “close-chatroom”.

```
sheldon_hung@MSI: ~  
Banned Users:  
Bob  
-----  
/exit-chatroom  
[System]: Tom had left the chatroom  
% close-chatroom chatroom1  
The chatroom was closed  
%  
|
```

```
sheldon_hung@MSI: ~  
Permitted Users:  
Tom  
-----  
/exit-chatroom  
[System]: Bob had left the chatroom  
% close-chatroom chatroom2  
The chatroom was closed  
%  
|
```

- Tom logout his account by using the command “logout”. Then close the connection by using the command “exit”.

```
sheldon_hung@MSI: ~  
<Tom>: Goodbye.  
/exit-chatroom  
[System]: Tom had left the chatroom  
% logout  
Bye, Tom.  
% exit  
sheldon_hung@MSI: ~$  
|
```

- Bob closes the connection by using the command “exit” without logging out the server. The server will still logout Bob implicitly.

```
sheldon_hung@MSI: ~  
<Tom>: Goodbye.  
[System]: Tom had left the chatroom  
/exit-chatroom  
[System]: Bob had left the chatroom  
% exit  
Bye, Bob.  
sheldon_hung@MSI: ~$  
|
```