

Introduction to Network Programming

Final Exam

Date: 2022/12/29

Time: 10:10-12:00

Submission deadline: 12:05

Chatroom (100%)

- You have to implement it with **C++**.
- You have to create a TCP server which is a chatroom. At least three clients in the chatroom.
- Server will give the client a name by connection order, for example, user0, user1, then user2.
- Each client will receive system welcome messages when entering the chatroom.
- Clients can send the private message to another user. Any client can not send messages to himself.
- Communication command format:
 - o Server: `./server {port}`
 - o Client: `./client {server-ip} {server-port}`
- You should ensure your code can run on the one of Docker images provided in HW2:
 - o yuthomas/np_hw2_x86
 - o jayzhan/np_hw2_arm

Grading

- system welcome message – (10%)
- mute command – (20%)
- unmute command – (20%)
- yell command – (20%)
- tell command – (20%)
- exit command – (10%)

Command	Description	Output	
mute	The client will not receive any message, including public and private messages.	Success	Mute mode.
		Fail	You are already in mute mode.
unmute	The client can receive messages again.	Success	Unmute mode.
		Fail	You are already in unmute mode.
yell <message>	Send a message to others in the chatroom.	Success	<username>: <message> Ex. user1: Hello world.
tell <receiver> <message>	Send a private message to <receiver> in the chatroom.	Success	<sender> told you: <message> Ex. user1 told you: It is nice to see you.
		Fail	<receiver> does not exist.
Exit	Close the chatroom and disconnect from the server.		

Scenario

PS: bold means command.

user0	user1	user2
bash\$./client 127.0.0.1 7890 ***** *Welcome to the BBS server.* ***** Welcome, user0.	bash\$./client 127.0.0.1 7890 ***** *Welcome to the BBS server.* ***** Welcome, user1.	bash\$./client 127.0.0.1 7890 ***** *Welcome to the BBS server.* ***** Welcome, user2.
user1: Hello world mute Mute mode.	yell Hello world yell Hello world2	user1: Hello world user1: Hello world2
unmute Unmute mode. user1: Hello world3	yell Hello world3	user1: Hello world3
	tell user2 It is nice to see you.	user1 told you: It is nice to see you.
exit	exit	exit

Hint

At `client.cpp`, you can take the below code for reference.

```
pthread_t send_t, rec_t;

if(pthread_create(&send_t, NULL, (void * (*)(void *)) send_msg, NULL) < 0) {

    perror("error: ");

    exit(1);

}

if(pthread_create(&rec_t, NULL, (void * (*)(void *)) rec_msg, NULL) < 0) {

    perror("error: ");

    exit(1);

}
```

Submission

All your files should be organized in the following hierarchy and zipped into a .zip file, named NP_XXXXXXX.zip, where XXXXXXX is your student ID.

Directory structure inside the zipped file:

- NP_XXXXXXX.zip (root)
 - server.cpp
 - client.cpp
 - Makefile (or README.txt)

Upload your NP_XXXXXXX.zip to E3.