

# University of Tehran

School of Electrical and Computer Engineering



Computer Networks

---

## Computer Assignment 1

---

**Student Name**

Alireza Javid

**Student ID**

810198375

**Instructor:**

Dr. Shahmansouri

# Contents

<b>1</b>	<b>How to run ?</b>	<b>2</b>
<b>2</b>	<b>Client Structure</b>	<b>3</b>
2.1	Connection . . . . .	3
2.2	List of Users . . . . .	4
2.3	Chat between Users . . . . .	5

# 1 How to run ?

For run project:

1. Compile server:

```
g++ -g -pthread message.h server.cpp main.cpp -o server
```

2. Compile client:

```
g++ -g -pthread message.h server.cpp client.cpp -o client
```

3. Run server:

```
./server
```

4. Run client:

```
./client localhost:9000 {client name}
```

## 2 Client Structure

In this assignment we create a simple chatroom by first creating a pair of server and client which communicate via a certain port, and next implementing functions such as sending and displaying messages including sender's and recipient's names in a common environment acting as a chatroom.

## 2.1 Connection

Client writes a message according given structure, then server replies and confirm connection. As we can see in figure 1 connection is established until end of program.

```
void connect(char* username, int sock) {
    Header reply_message, message;
    message.message_type = CONNECT;
    message.message_id = MESSAGE_ID;
    message.length = HEADER_LENGTH + strlen(username);
    write(sock, (uint8_t*)&message, sizeof(Header));
    write(sock, username, strlen(username));
    read(sock, (uint8_t*)&reply_message, sizeof(Header));
}
```

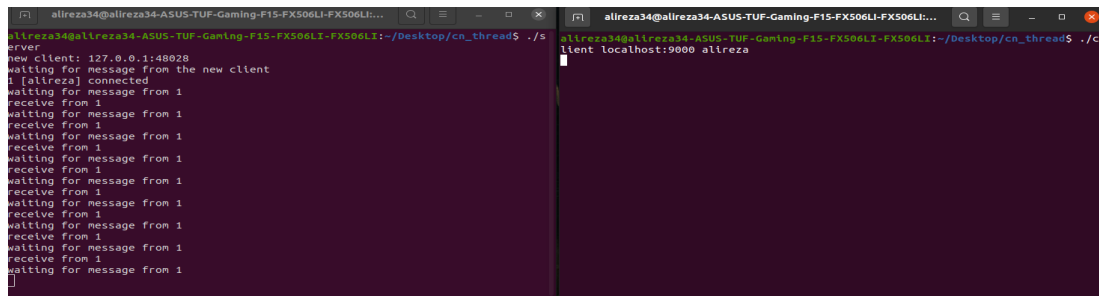
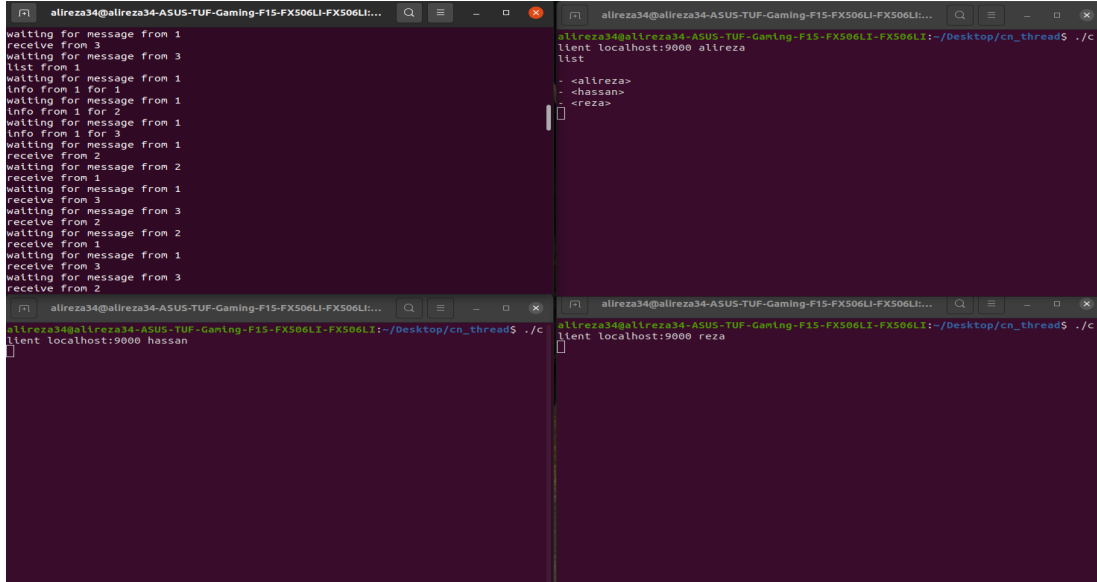


Figure 1: Client connection

## 2.2 List of Users

Client can get list of names of every users connected to the server with list command. First user writes list command, then Client writes a message according given structure, then server replies and gives client users id. Client writes info message for every id and gets name of users from server. We can see in figure 2 how user get list of every user connected to the server.



```
allireza34@allireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI:~$ ./c
waiting for message from 1
receive from 3
waiting for message from 3
list from 1
waiting for message from 1
info from 1 for 1
waiting for message from 1
info from 1 for 2
waiting for message from 1
info from 1 for 3
waiting for message from 1
receive from 2
waiting for message from 2
receive from 1
waiting for message from 1
receive from 3
waiting for message from 3
receive from 2
waiting for message from 2
receive from 1
waiting for message from 1
receive from 3
waiting for message from 3
receive from 2

allireza34@allireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI:~$ ./c
lient localhost:9000 allireza
list
- <allireza>
- <hassan>
- <reza>

allireza34@allireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI:~$ ./c
lient localhost:9000 hassan

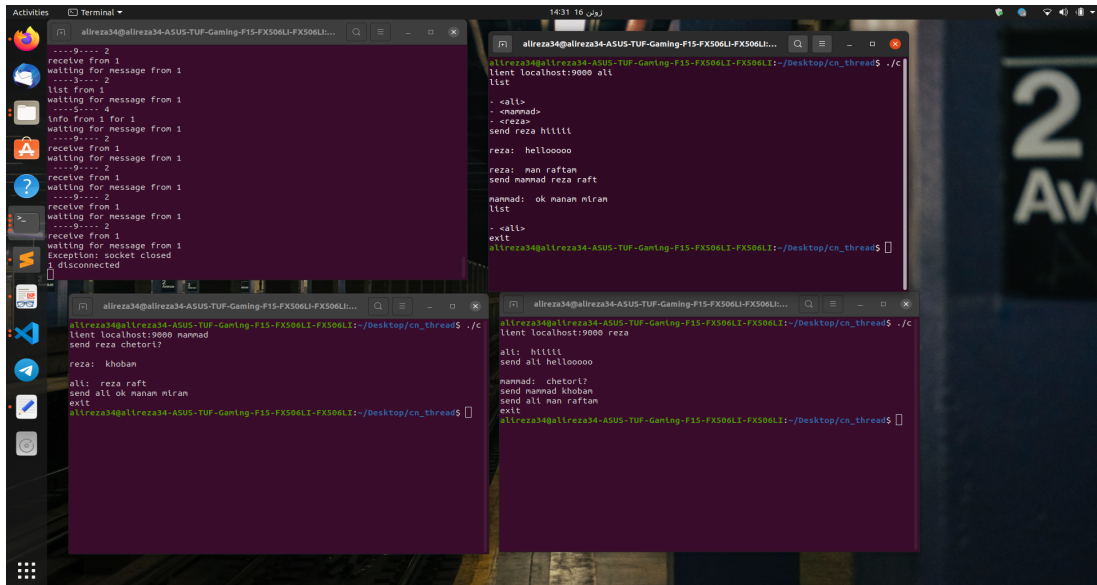
allireza34@allireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI:~$ ./c
lient localhost:9000 reza
```

Figure 2: Get list of users

Details of implication is attached with report.

## 2.3 Chat between Users

For this purpose we use multi thread socket programming and include pthread.h library. Every 2 second client checks if have any message from other users or not. With every check client writes receive message and waits for server to answer. After client receives message, writes receive reply to server. For sending message, after client gets id of destination user from server, writes send message and waits for send reply of server. In this program users communicate with the server with a TCP socket. Figure 3 shows example of running program.



```
alireza34@alireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI...  
...9... 2  
receive from 1  
waiting for message from 1  
...9... 2  
list from 1  
waiting for message from 1  
...9... 4  
info from 1 for 1  
waiting for message from 1  
...9... 2  
receive from 1  
waiting for message from 1  
...9... 2  
receive from 1  
waiting for message from 1  
...9... 2  
receive from 1  
waiting for message from 1  
...9... 2  
receive from 1  
waiting for message from 1  
Exception: socket closed  
1 disconnected
```

```
alireza34@alireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI...  
lient localhost:9000 all  
list  
- <all>  
- <namad>  
- <reza>  
send reza hlllll  
reza: hellooooo  
reza: man raftan  
send namnad reza raft  
namnad: ok nanan niran  
list  
- <all>  
- exit  
alireza34@alireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI...  
alireza34@alireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI...  
lient localhost:9000 reza  
all: hlllll  
send all hellooooo  
namnad: chetori?  
send namnad khoban  
send all nan raftan  
exit
```

```
alireza34@alireza34-ASUS-TUF-Gaming-F15-FX506LI-FX506LI...  
lient localhost:9000 namnad  
send reza chetori?  
reza: khoban  
all: reza raft  
send all ok nanan niran  
exit
```

Figure 3: Users communicate in program

Details of implication is attached with report.