

Name : Subham Subhasis Sahoo
Entry No : 2020CSB1317

Task-1 :

Step 1 :

Run the server.out file

Commands:

To compile: `gcc server.c -o server.out`

To run: `./server.out`

A screenshot of a terminal window with a dark background. The terminal shows the command `subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-1$./server.out` and its output: `[LISTENING] Server is listening on port 5555`. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and COMMENTS. The TERMINAL tab is active. There are icons for window management in the top right corner.

```
subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-1$ ./server.out
[LISTENING] Server is listening on port 5555
```

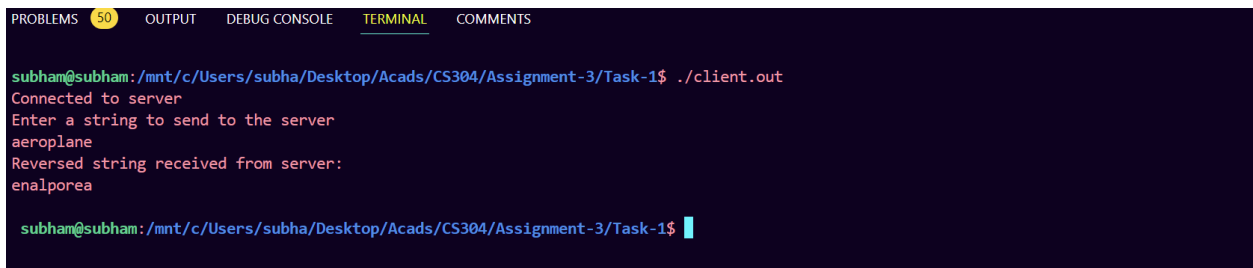
Step 2:

Run any of the client files (client-1.c , client-2.c , client-3.c , client-4.c , client-5.c)

To compile : `gcc client-1.c -o client.out`

To run: `./client.out`

Input the string you want to send to the server.

A screenshot of a terminal window with a dark background. The terminal shows the command `subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-1$./client.out` and its output: `Connected to server`, `Enter a string to send to the server`, `aeroplane`, `Reversed string received from server:`, and `enalporea`. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and COMMENTS. The TERMINAL tab is active. There are icons for window management in the top right corner.

```
subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-1$ ./client.out
Connected to server
Enter a string to send to the server
aeroplane
Reversed string received from server:
enalporea

subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-1$
```

Task-2:

Step 1 :

Run the server.out file

Commands:

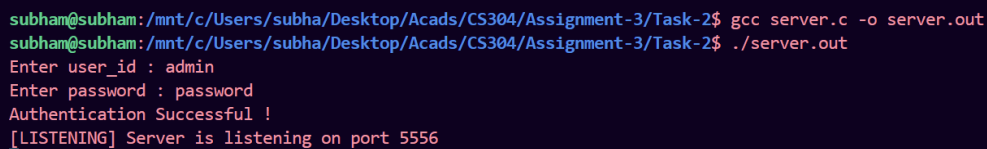
To compile: `gcc server.c -o server.out`

To run: `./server.out`

Enter the credentials :

User_id : admin

Password : password



```
subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-2$ gcc server.c -o server.out
subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-2$ ./server.out
Enter user_id : admin
Enter password : password
Authentication Successful !
[LISTENING] Server is listening on port 5556
```

Step 2:

Run any of the client files (client-1.c , client-2.c , client-3.c , client-4.c , client-5.c)

To compile : `gcc client-1.c -o client.out`

To run: `./client.out`

Enter the ticket booking info in this format :

client-A:B

where

A = client id

B = number of tickets to be booked

Attached is the screenshot of the process.

The initial number of tickets 1000 is stored in the server_records.txt file itself.

[INFO] Tickets Available : 1000

The initial number of tickets can be changed here in place of 1000.

```
server_records.txt M X
Task-2 > server_records.txt
1 [INFO] Tickets Available : 1000
2
3
4
5 Client 1 booked : 366
6
7 [INFO] Tickets Available : 634
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-2$ gcc client-1.c -o client.out
subham@subham:/mnt/c/Users/subha/Desktop/Acads/CS304/Assignment-3/Task-2$ ./client.out
Connected to server
=====
Enter tickets in this format

client-A:B

where
A = client id
B = number of tickets to be booked

Enter QUIT to quit
=====
client-1:366
Data sent to the server
Data received from the server: Tickets Booked : 366
Tickets Now Available : 634

Tickets Booked : 366
Tickets Now Available : 634
```

After booking through 5 clients, servers_records.txt would look like :

```
server_records.txt M X
Task-2 > server_records.txt
1 [INFO] Tickets Available : 1000
2
3
4
5 Client 1 booked : 366
6
7 [INFO] Tickets Available : 634
8
9 Client 3 booked : 45
10
11 [INFO] Tickets Available : 589
12
13 Client 4 booked : 234
14
15 [INFO] Tickets Available : 355
16
17 Client 5 booked : 90
18
19 [INFO] Tickets Available : 265
20
21 Client 2 booked : 89
22
23 [INFO] Tickets Available : 176
24
```