

## TCP Server and Clients

To make the server accept connection it should be listening to any new connection all the time so I used a thread for listening for new client and establish connection with them.

The server should be able for sending a commands for the clients at any time so I used another thread and it's responsible for send a message for selected clients.

I used 1997 as port number for my server and it's important to use any port larger than 1024.

```
Set Server port number As: 1997
Set Server host name As: MaNarOdEh
Binding the port number: 1997
```

When any new client connected to that server it will prints both the IP addresses and Port number for that client

```
Connection has been established! IP 192.168.1.113 | Port 59487

Connection has been established! IP 192.168.1.113 | Port 59488
|
```

In the server you can show all the connected clients by typing [list](#)

[Note:](#) - I send an empty message to each client to make sure that the connection is still and the client is not crash and If I receive any response then the connection with that client is still good, if not I delete that connection from the list.

```
input (list to show all connected clients or select any target clientby input select id_client:
list
<<Clients>>:
0 192.168.1.113 59487
1 192.168.1.113 59488
```

And also, you can [select](#) any of them to send any direct message.

```
input (list to show all connected clients or select any target clientby input select id_client:
select 0
You are now connected to :192.168.1.113
192.168.1.113 > send target messages!

Input the Message You wan't to send to the chosen client:
```

When you send any message to the choose client it will show in the client console and if the message received correctly the server also will received from the client that [the message is Received Correctly](#).

Server: -

```
Input the Message You wan't to send to the chosen client: Hello From The Server!!
the Result Message From Our client:  Message Is Received Correctly
```

Client: -

```
Received Message is : Hello From The Server!!
```

Close:

When the server sends close to any clients the client will response with close message and the server will closed that connections

Server: -

```
Input the Message You wan't to send to the chosen client: close
the Result Message From Our client:  close
Connection with that clients is closed!
```

Client: -

```
Received Message is : close
The connection with that server is closed!!
```

In the other side the client should be able to [send a request to connect with the server](#) and [accept any incoming message](#) and [also send message to the server at any time](#).

So, I used a thread for receiving any incoming message.

And also, another thread for sending message to the server.

Receiving message from server

```
Received Message is : Hello From The Server!!
```

Sending message from client to the server

```
Input the Message You wan't to send: hey!!

The Server Received The Message!
```

