LAB2 – Sockets

Leslie Dahlberg (ldg14001@student.mdh.se)

Assignment

The assignment consists of modifying a client-server application only capable of one-way communication. The following functionality was implemented:

1. The server answers the clients when it receives messages from them
2. The client prints the servers answers on screen
3. The server broadcasts information about any new connected clients to already connected clients
4. The server refuses connections from banned IP-adresses
   - To simulate this on one computer the client can fake its IP-address and port number (./client serverIP clientIP clientPort).

Usage

To compile the application run “make” in the application directory. This results in two files: client and server.

Usage: server

Run the server by typing “./server” in the application directory. The server will print incoming messages and information about connected clients.

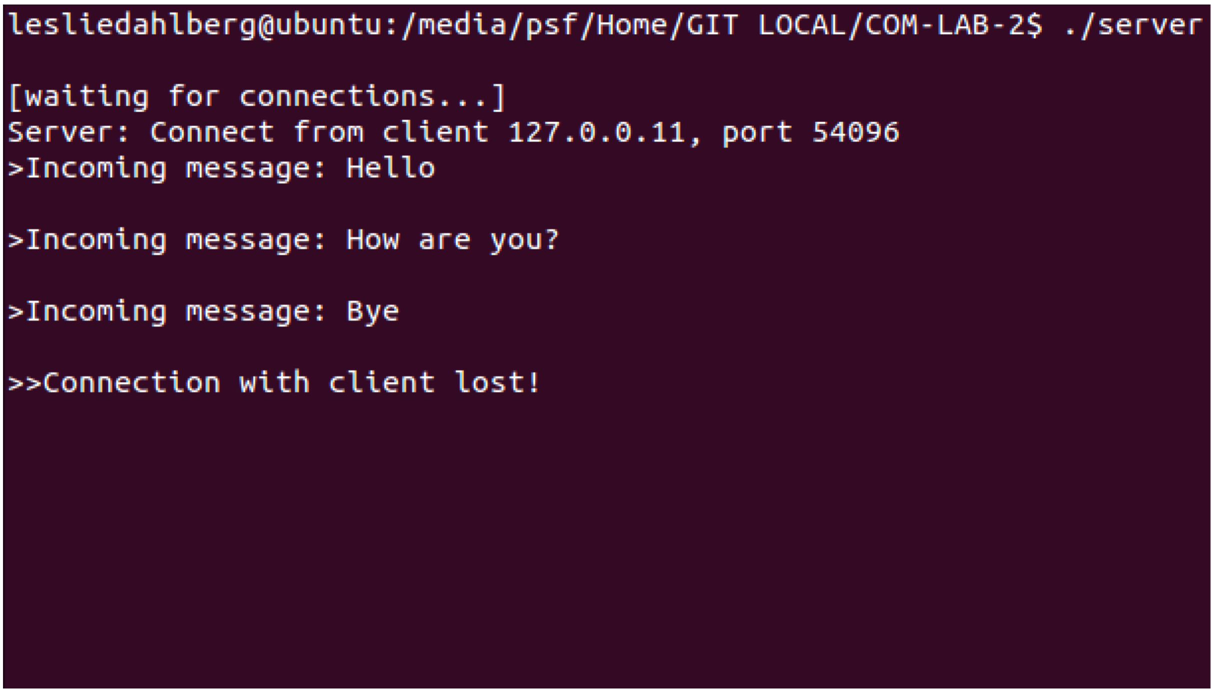
Usage: client

Run the client by typing “./client [server IP] [client IP] [client PORT]” in the application directory. The client will try to connect to the server with the specified IP-address and port number. Type any message and hit enter to send to the server. Type “quit” to close the connection. The client will also receive updates about new clients when they connect.

Design

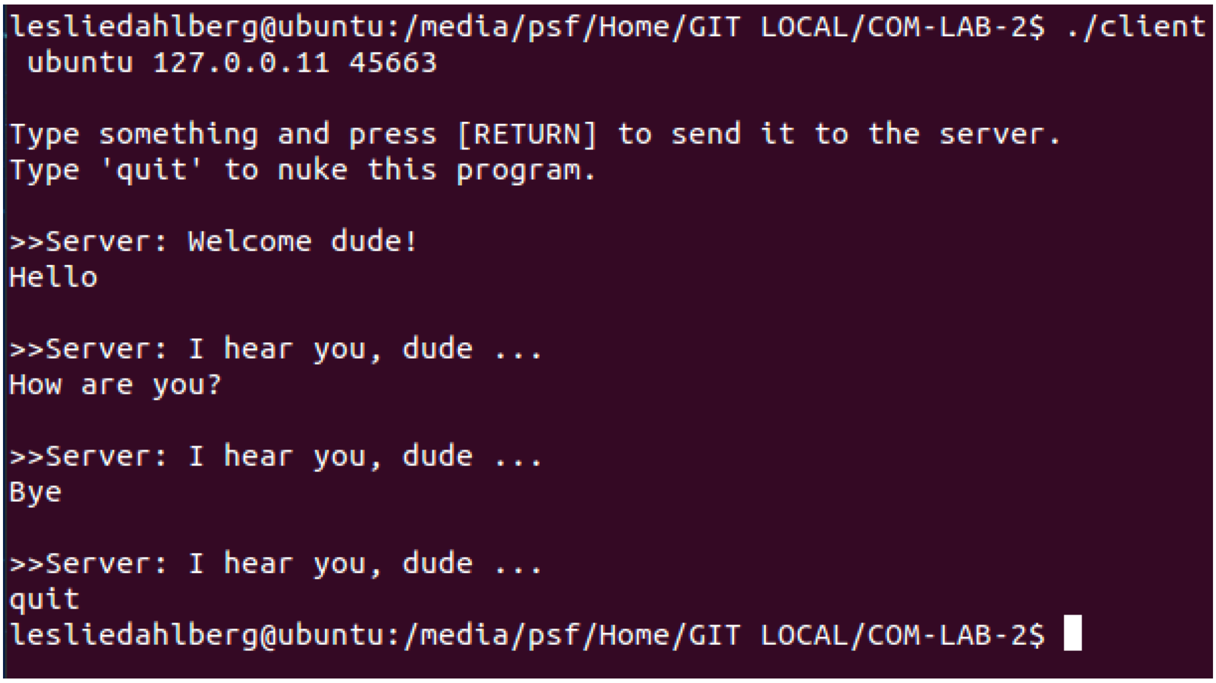
The client runs two threads. One for establishing a connection, listening to user input and sending data to the server and one for listening to replies from the server. The client and the server use the same functions to send and receive data from each other.

The server creates a socket and binds it to its IP-address. A loop checks for new connections and receives messages from connected clients. New clients sockets are stored in an array used to send broadcasts. Se figures 1-4.



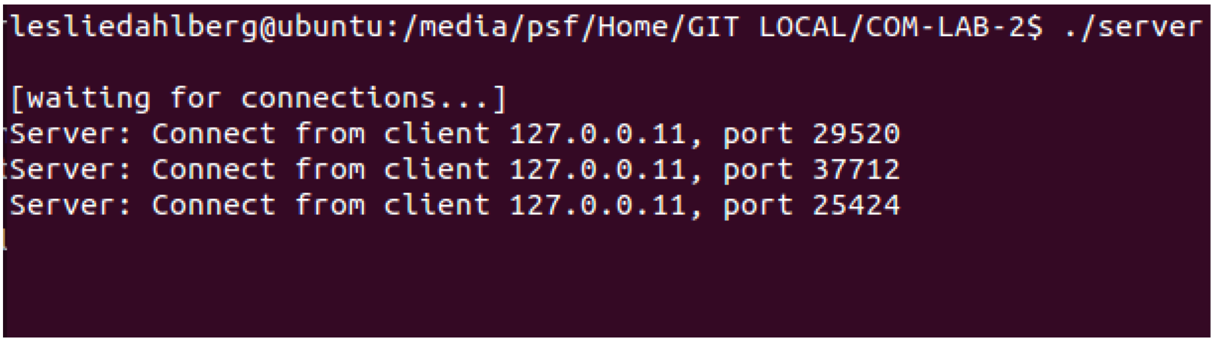
```
lesliedahlberg@ubuntu:/media/psf/Home/GIT LOCAL/COM-LAB-2$ ./server

[waiting for connections...]
Server: Connect from client 127.0.0.11, port 54096
>Incoming message: Hello

>Incoming message: How are you?

>Incoming message: Bye

>>Connection with client lost!
```

Figure 1: Server with one connected client



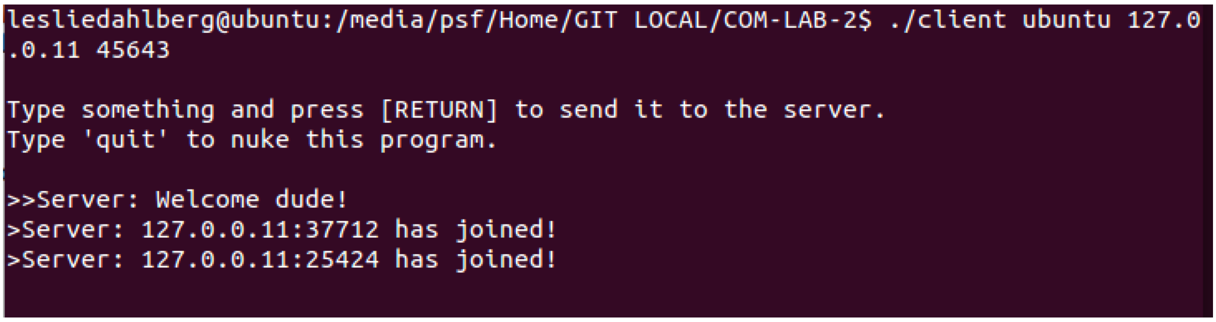
```
lesliedahlberg@ubuntu:/media/psf/Home/GIT LOCAL/COM-LAB-2$ ./client
 ubuntu 127.0.0.11 45663

Type something and press [RETURN] to send it to the server.
Type 'quit' to nuke this program.

>>Server: Welcome dude!
Hello

>>Server: I hear you, dude ...
How are you?

>>Server: I hear you, dude ...
Bye

>>Server: I hear you, dude ...
quit
lesliedahlberg@ubuntu:/media/psf/Home/GIT LOCAL/COM-LAB-2$ 
```

Figure 2: Client connected to server



```
lesliedahlberg@ubuntu:/media/psf/Home/GIT LOCAL/COM-LAB-2$ ./server

[waiting for connections...]
Server: Connect from client 127.0.0.11, port 29520
Server: Connect from client 127.0.0.11, port 37712
Server: Connect from client 127.0.0.11, port 25424
```

Figure 3: Server with multiple clients



```
lesliedahlberg@ubuntu:/media/psf/Home/GIT LOCAL/COM-LAB-2$ ./client ubuntu 127.0
.0.11 45643

Type something and press [RETURN] to send it to the server.
Type 'quit' to nuke this program.

>>Server: Welcome dude!
>Server: 127.0.0.11:37712 has joined!
>Server: 127.0.0.11:25424 has joined!
```

Figure 4: Client receiving updates about other clients