Comp 4981 Computer Systems Technology January 2014

Data Communication Option

Assignment #3 Design Doc

Ian Davidson, Josh Campbell
Set 40
March 24th, 2014

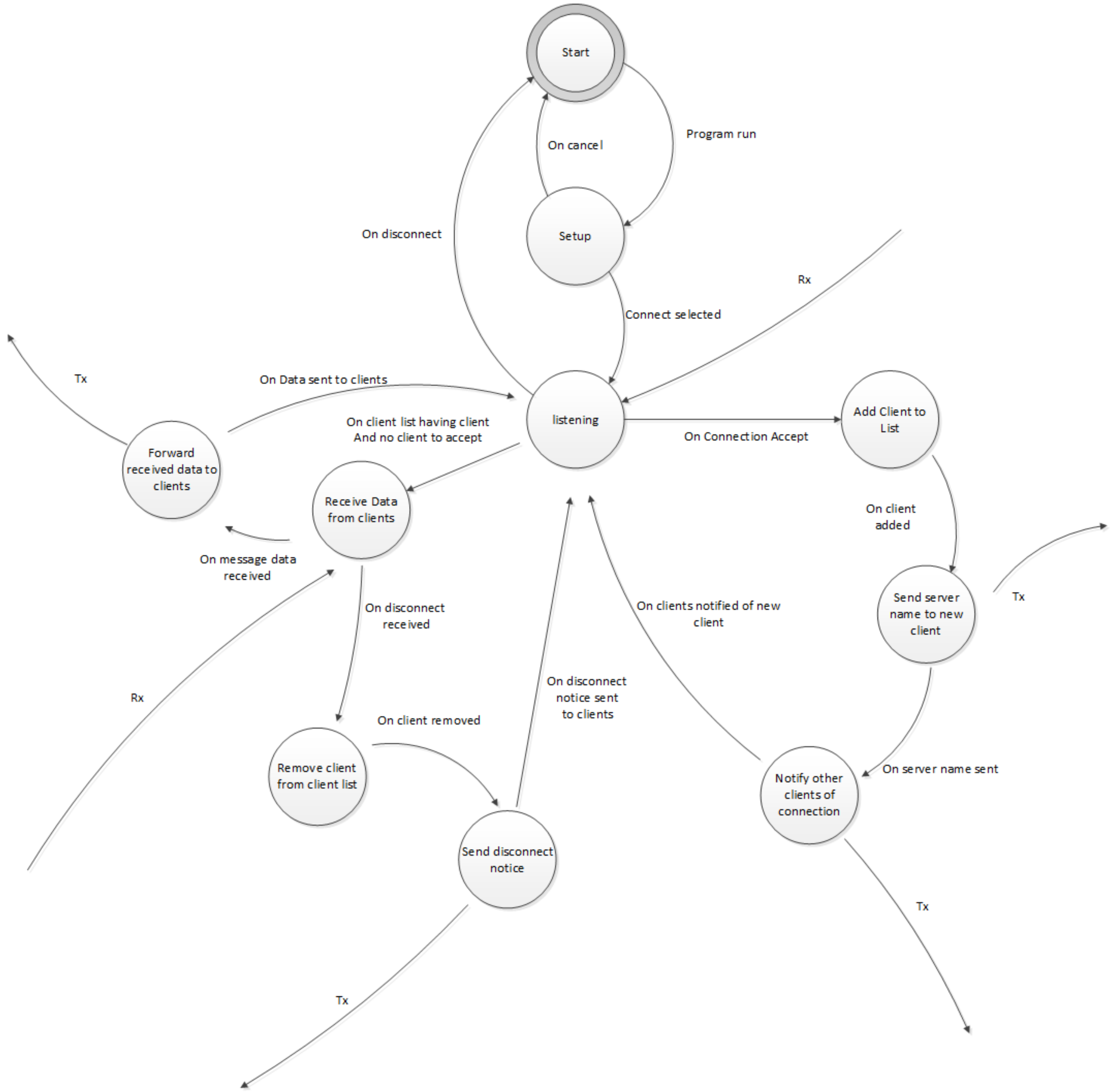
Design

Testing

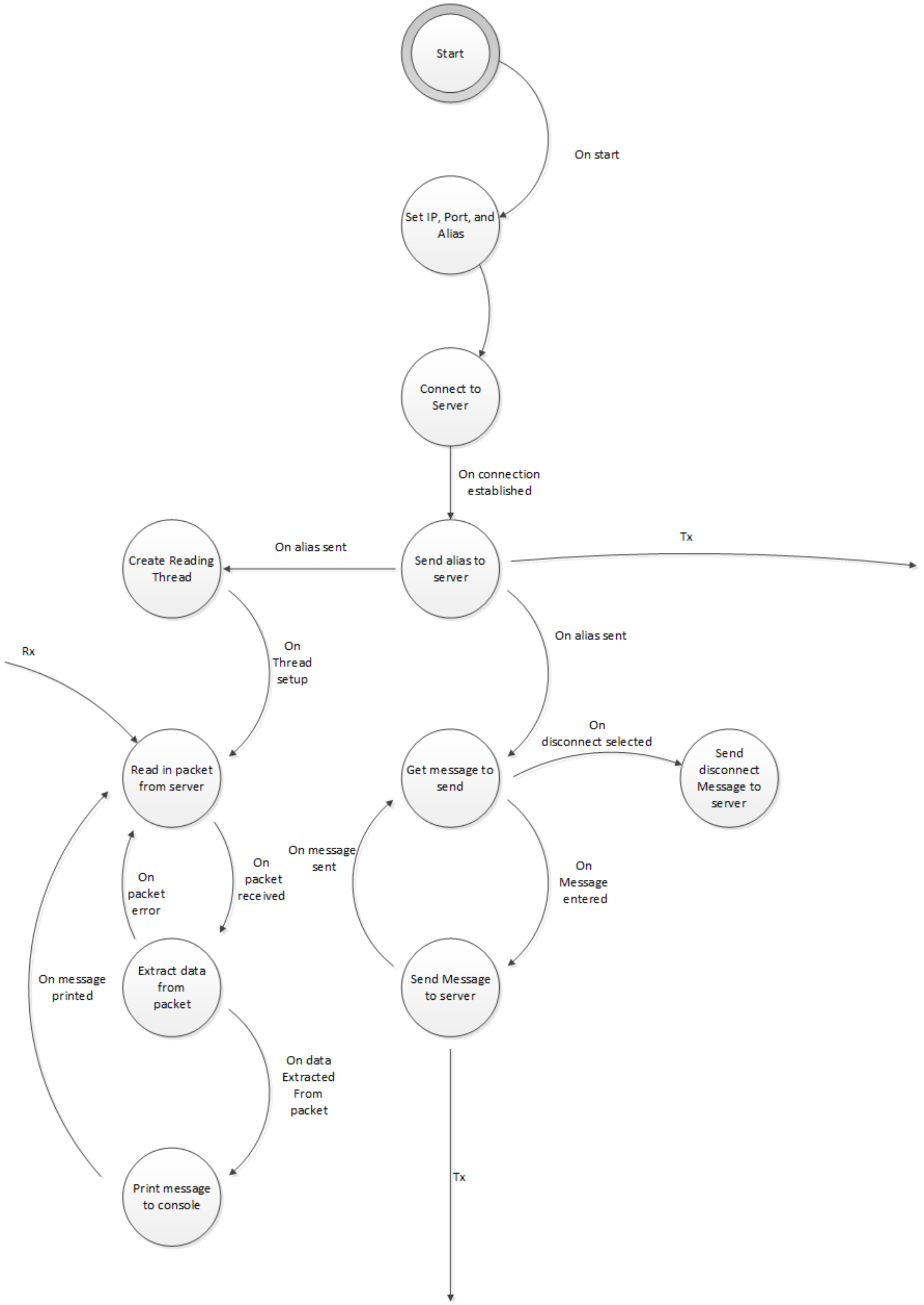
Use

State Transition Diagram

Server


Start
Program run
On cancel
Setup
On disconnect
Connect selected
Rx
listening
Tx
On Data sent to clients
On client list having client And no client to accept
On Connection Accept
Add Client to List
Forward received data to clients
Receive Data from clients
On client added
On message data received
On disconnect received
On clients notified of new client
Send server name to new client
Tx
Rx
On disconnect notice sent to clients
On client removed
Remove client from client list
On server name sent
Notify other clients of connection
Send disconnect notice
Tx
Tx

Client


Start
On start
Set IP, Port, and Alias
Connect to Server
On connection established
Create Reading Thread
On alias sent
Send alias to server
Tx
Rx
On Thread setup
On alias sent
On disconnect selected
Read in packet from server
Get message to send
Send disconnect Message to server
On message sent
On packet received
On packet error
On Message entered
On message printed
Extract data from packet
Send Message to server
On data Extracted From packet
Tx
Print message to console

Pseudo Code

```
function Client()
        display prompt for ip address
        read in ip

        display prompt for port
        read in port

        display prompt for displayname
        read in display name

        create socket connection to server

        Send display name to server

        create ReadingThread

        while running
                get message to send

                store message in send packet

                send message to server
        end while

        send disconnect packet

end function

function ReadingThread()
        while running
                read in data

                if data is server join packet
                        print the name of the server
                else if data is client join packet
                        print the name of the user that joined
                else if data is client message packet
                        print message from the client
                else if data is client disconnect packet
                        print user that disconnected
                end
        end while
end function
```

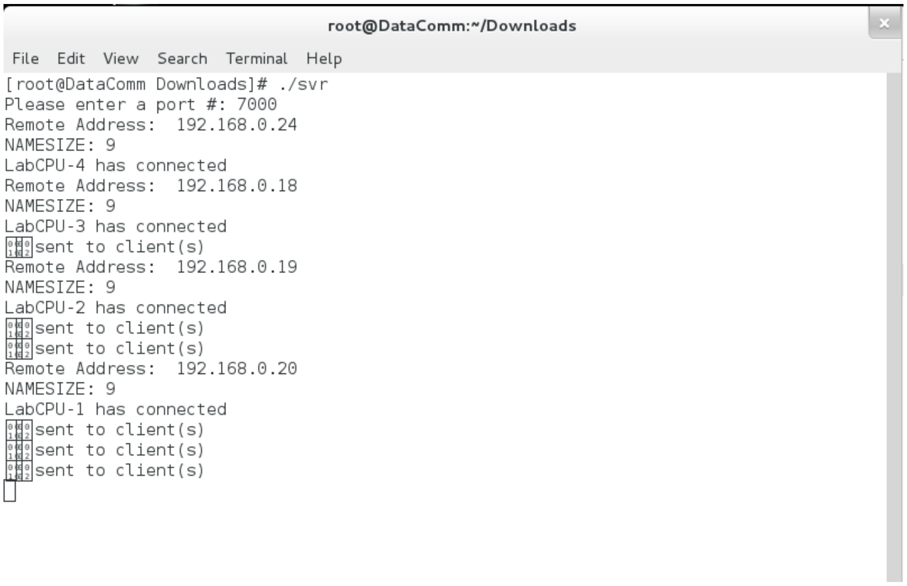
```
function Server()
    get port to listen on

    create new listening socket

    set "select" on the socket so that we can check if there are new connections

    while running
        check if there is a new connection

        if there is a new connection
            create new socket using the accept function

            send server name to new client

            if there are clients in client list
                send alert regarding new client connecting to previously connected clients
            end if

            add new client to client list
        end if

        check if there is data to receive

        if there is data to be received
            read in data

            if data is client message packet
                forward message to clients that are not the sending client
            else if data is client disconnect packet
                remove client from client list

                send message to all clients in client list regarding the disconnected client
            end
        end if

    end while

end function
```

Testing Table

Test	Test Description	Tools Used	Expected Result	Pass/Fail
1	Server accepts multiple clients.	N/A	The server handles the connections	PASS, see fig 1.
2	Server tells other clients when a new user connects.	N/A	The server sends the client has joined msg to other clients	PASS, see fig 2.
3	Server forwards messages to clients that didn't send the message.	N/A	Clients receive other clients text	PASS, see fig 3. and 4.
4	Server alerts users when a user disconnects.	N/A	Clients receive msg that other clients have disconnected	PASS, see fig 5. and 6.
5	Client connects to the server.	N/A	The client gets connected to the server and receives the room name	PASS, see fig 7.
6	Client sets user name and sends it to the server.	N/A	server and clients receive user name	PASS, see fig 1. and 7.
7	Client sends messages to the server to be forwarded to the other clients.	N/A	The server sends out the msg to all other clients	PASS, see fig 3. and 4.

8	Client alerts server when it disconnects.	N/A	Server disconnects them and sends a msg to other clients	PASS, see fig 6.

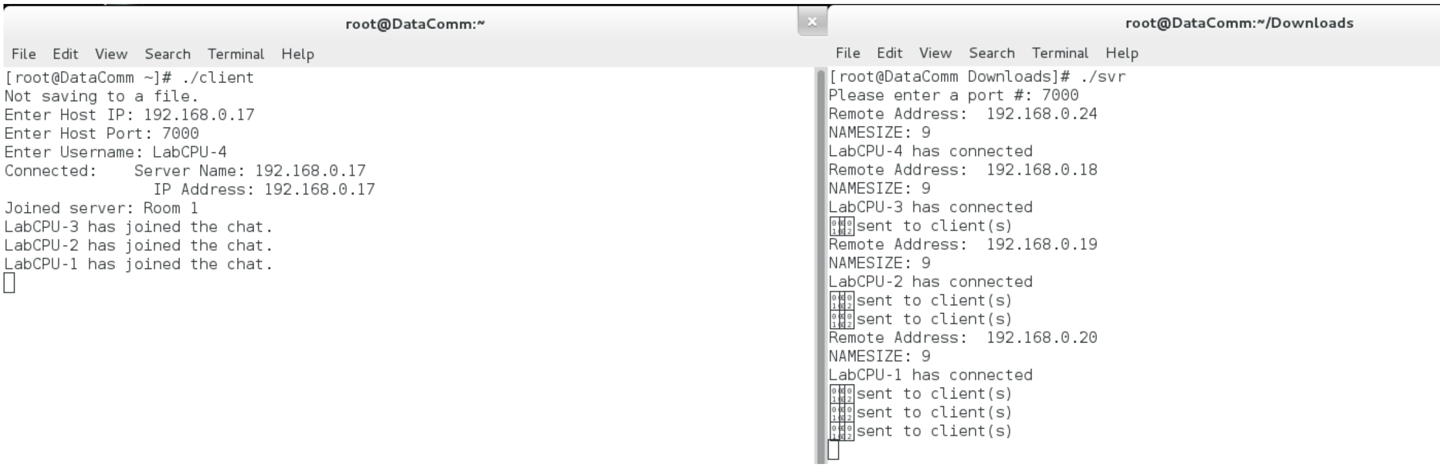
Figures



```
[root@DataComm Downloads]# ./svr
Please enter a port #: 7000
Remote Address:  192.168.0.24
NAMESIZE: 9
LabCPU-4 has connected
Remote Address:  192.168.0.18
NAMESIZE: 9
LabCPU-3 has connected
▯▯▯sent to client(s)
Remote Address:  192.168.0.19
NAMESIZE: 9
LabCPU-2 has connected
▯▯▯sent to client(s)
▯▯▯sent to client(s)
Remote Address:  192.168.0.20
NAMESIZE: 9
LabCPU-1 has connected
▯▯▯sent to client(s)
▯▯▯sent to client(s)
▯▯▯sent to client(s)
```

root@DataComm:~/Downloads
File Edit View Search Terminal Help

Figure 1. The server accepts multiple clients.

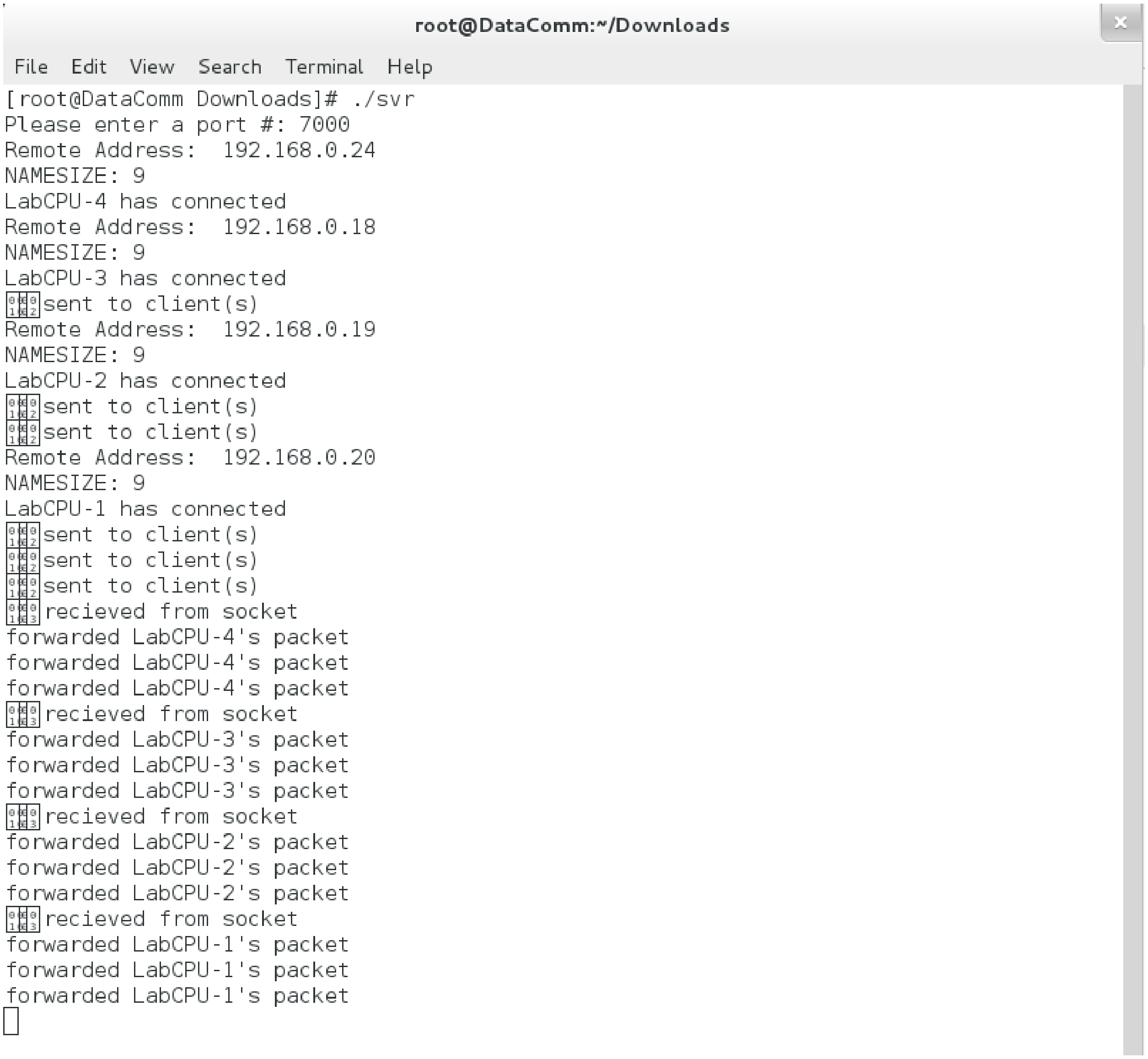


```
[root@DataComm ~]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-4
Connected:    Server Name: 192.168.0.17
                IP Address: 192.168.0.17
Joined server: Room 1
LabCPU-3 has joined the chat.
LabCPU-2 has joined the chat.
LabCPU-1 has joined the chat.
```

```
[root@DataComm Downloads]# ./svr
Please enter a port #: 7000
Remote Address:  192.168.0.24
NAMESIZE: 9
LabCPU-4 has connected
Remote Address:  192.168.0.18
NAMESIZE: 9
LabCPU-3 has connected
▯▯▯sent to client(s)
Remote Address:  192.168.0.19
NAMESIZE: 9
LabCPU-2 has connected
▯▯▯sent to client(s)
▯▯▯sent to client(s)
Remote Address:  192.168.0.20
NAMESIZE: 9
LabCPU-1 has connected
▯▯▯sent to client(s)
▯▯▯sent to client(s)
▯▯▯sent to client(s)
```

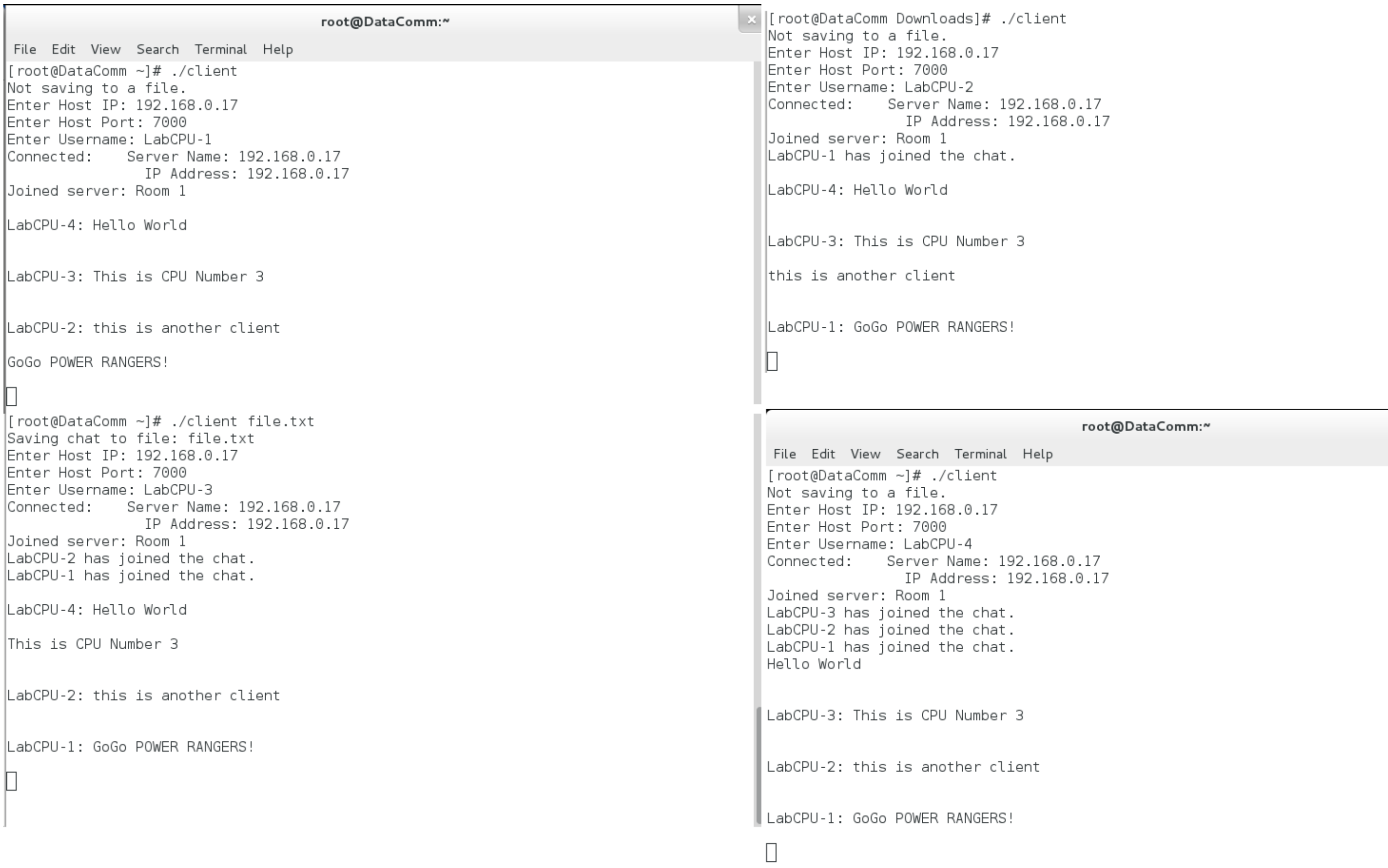
root@DataComm:~
File Edit View Search Terminal Help
root@DataComm:~/Downloads
File Edit View Search Terminal Help

Figure 2. The server tells other clients when a new user connects.


root@DataComm:~/Downloads
File Edit View Search Terminal Help

```
[root@DataComm Downloads]# ./svr
Please enter a port #: 7000
Remote Address:  192.168.0.24
NAMESIZE: 9
LabCPU-4 has connected
Remote Address:  192.168.0.18
NAMESIZE: 9
LabCPU-3 has connected
□□□sent to client(s)
Remote Address:  192.168.0.19
NAMESIZE: 9
LabCPU-2 has connected
□□□sent to client(s)
□□□sent to client(s)
Remote Address:  192.168.0.20
NAMESIZE: 9
LabCPU-1 has connected
□□□sent to client(s)
□□□sent to client(s)
□□□sent to client(s)
□□□recieved from socket
forwarded LabCPU-4's packet
forwarded LabCPU-4's packet
forwarded LabCPU-4's packet
□□□recieved from socket
forwarded LabCPU-3's packet
forwarded LabCPU-3's packet
forwarded LabCPU-3's packet
□□□recieved from socket
forwarded LabCPU-2's packet
forwarded LabCPU-2's packet
forwarded LabCPU-2's packet
□□□recieved from socket
forwarded LabCPU-1's packet
forwarded LabCPU-1's packet
forwarded LabCPU-1's packet
```

Figure 3. Server forwards messages to clients that didn’t send the original message.



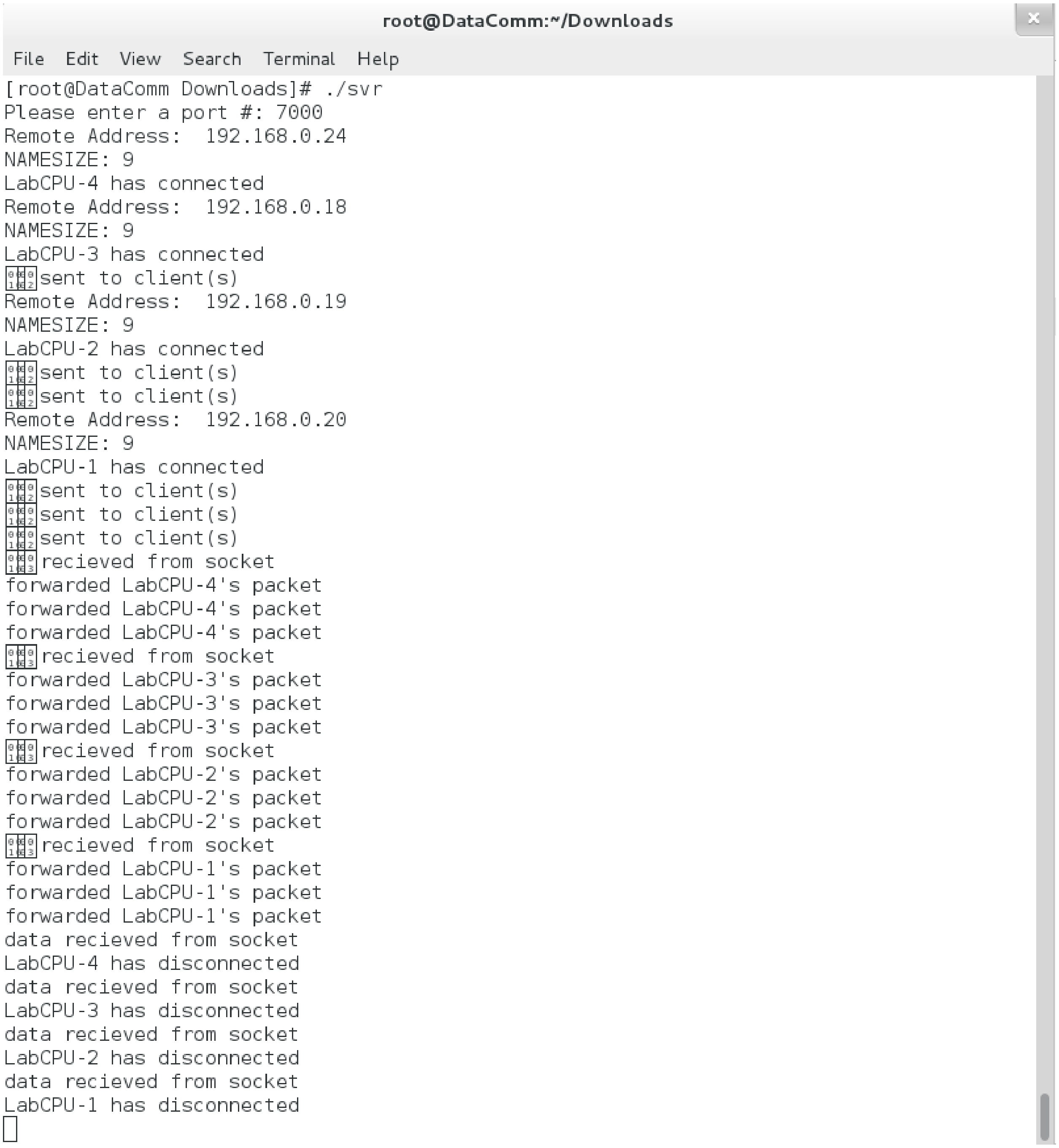
```
[root@DataComm ~]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-1
Connected:    Server Name: 192.168.0.17
                IP Address: 192.168.0.17
Joined server: Room 1

LabCPU-4: Hello World


LabCPU-3: This is CPU Number 3


LabCPU-2: this is another client

GoGo POWER RANGERS!
```

```
[root@DataComm ~]# ./client file.txt
Saving chat to file: file.txt
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-3
Connected:    Server Name: 192.168.0.17
                IP Address: 192.168.0.17
Joined server: Room 1
LabCPU-2 has joined the chat.
LabCPU-1 has joined the chat.

LabCPU-4: Hello World

This is CPU Number 3


LabCPU-2: this is another client


LabCPU-1: GoGo POWER RANGERS!
```

```
[root@DataComm Downloads]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-2
Connected:    Server Name: 192.168.0.17
                IP Address: 192.168.0.17
Joined server: Room 1
LabCPU-1 has joined the chat.

LabCPU-4: Hello World


LabCPU-3: This is CPU Number 3

this is another client


LabCPU-1: GoGo POWER RANGERS!
```

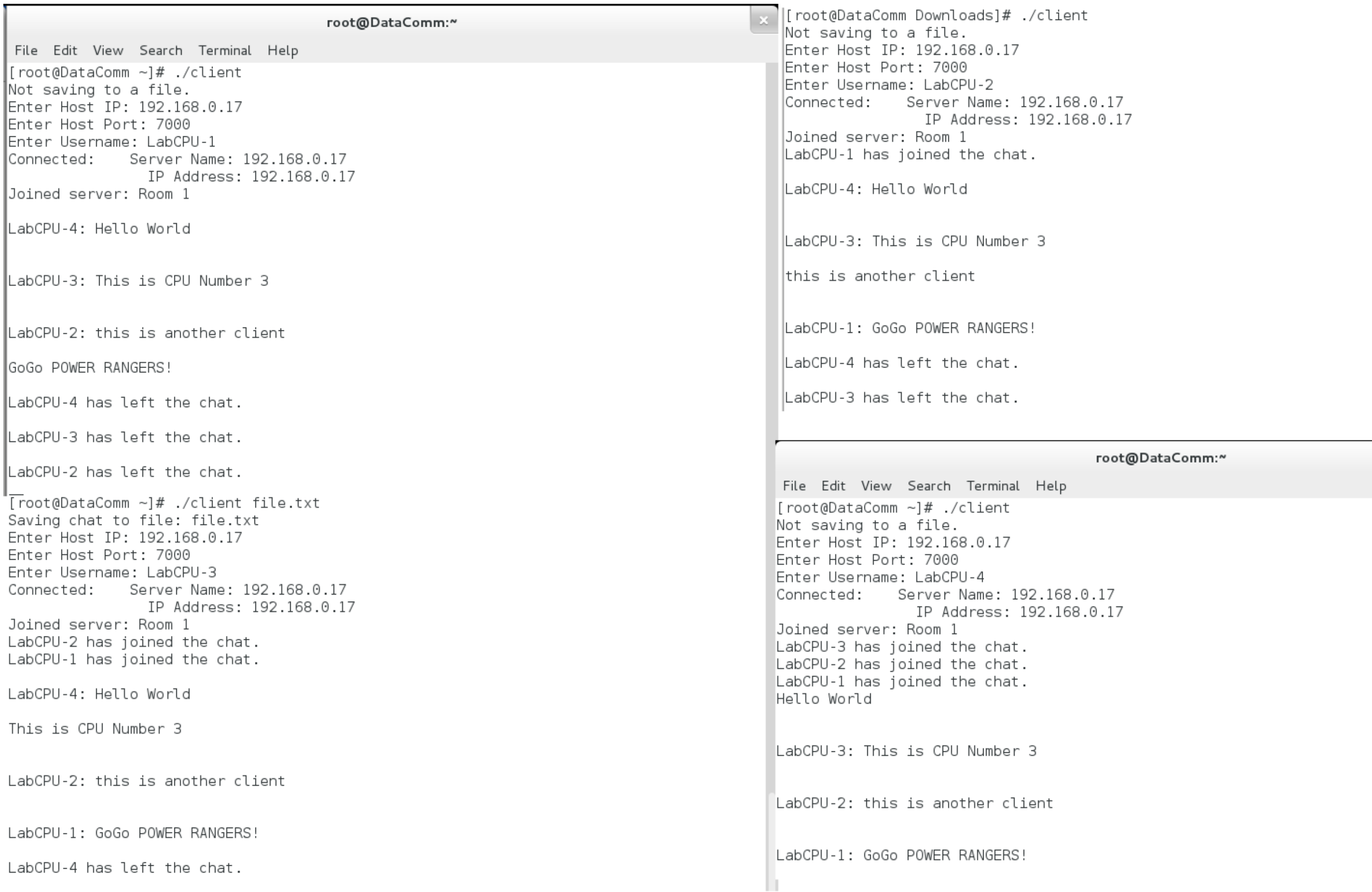
```
[root@DataComm ~]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-4
Connected:    Server Name: 192.168.0.17
                IP Address: 192.168.0.17
Joined server: Room 1
LabCPU-3 has joined the chat.
LabCPU-2 has joined the chat.
LabCPU-1 has joined the chat.
Hello World


LabCPU-3: This is CPU Number 3


LabCPU-2: this is another client


LabCPU-1: GoGo POWER RANGERS!
```

Figure 4. Clients receive messages they didn't send out from the server.


root@DataComm:~/Downloads
File Edit View Search Terminal Help

```
[root@DataComm Downloads]# ./svr
Please enter a port #: 7000
Remote Address:  192.168.0.24
NAMESIZE: 9
LabCPU-4 has connected
Remote Address:  192.168.0.18
NAMESIZE: 9
LabCPU-3 has connected
□□sent to client(s)
Remote Address:  192.168.0.19
NAMESIZE: 9
LabCPU-2 has connected
□□sent to client(s)
□□sent to client(s)
Remote Address:  192.168.0.20
NAMESIZE: 9
LabCPU-1 has connected
□□sent to client(s)
□□sent to client(s)
□□sent to client(s)
□□recieved from socket
forwarded LabCPU-4's packet
forwarded LabCPU-4's packet
forwarded LabCPU-4's packet
□□recieved from socket
forwarded LabCPU-3's packet
forwarded LabCPU-3's packet
forwarded LabCPU-3's packet
□□recieved from socket
forwarded LabCPU-2's packet
forwarded LabCPU-2's packet
forwarded LabCPU-2's packet
□□recieved from socket
forwarded LabCPU-1's packet
forwarded LabCPU-1's packet
forwarded LabCPU-1's packet
data recieved from socket
LabCPU-4 has disconnected
data recieved from socket
LabCPU-3 has disconnected
data recieved from socket
LabCPU-2 has disconnected
data recieved from socket
LabCPU-1 has disconnected
```

Figure 5. The server alerts clients when a client disconnects.


root@DataComm:~
File Edit View Search Terminal Help
root@DataComm:~
File Edit View Search Terminal Help

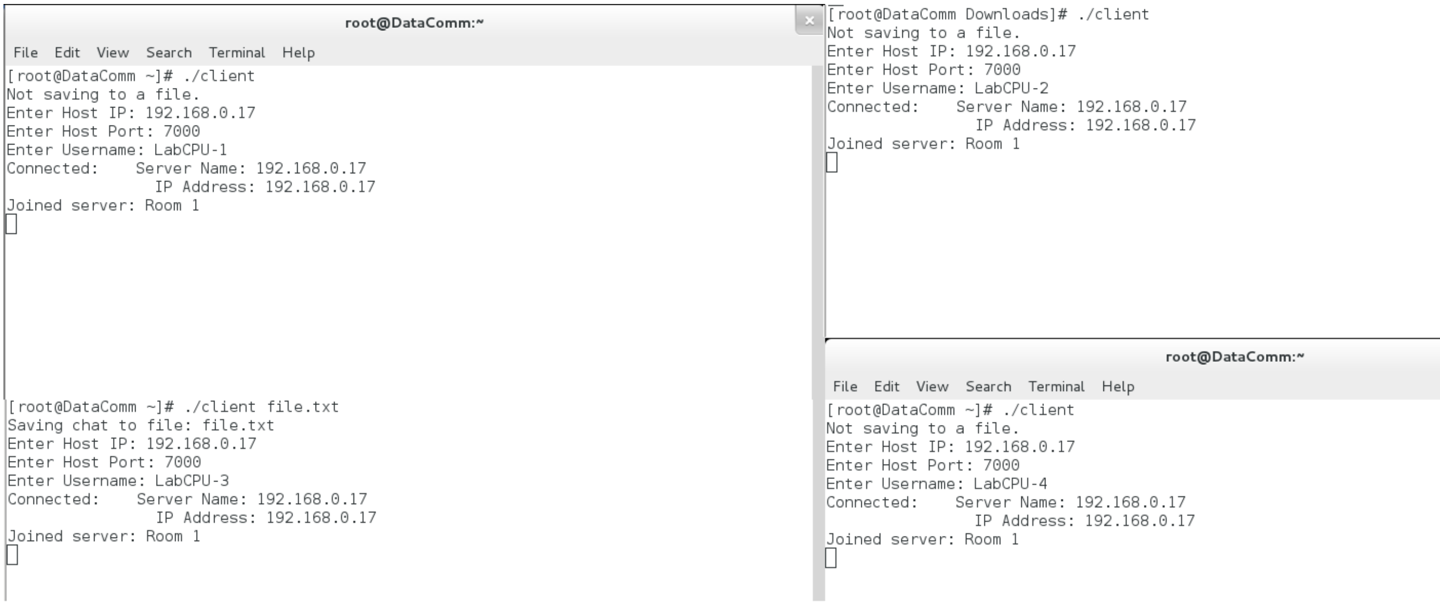
```
[root@DataComm ~]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-1
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1

LabCPU-4: Hello World


LabCPU-3: This is CPU Number 3


LabCPU-2: this is another client

GoGo POWER RANGERS!

LabCPU-4 has left the chat.

LabCPU-3 has left the chat.

LabCPU-2 has left the chat.
```

```
[root@DataComm Downloads]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-2
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1
LabCPU-1 has joined the chat.

LabCPU-4: Hello World


LabCPU-3: This is CPU Number 3

this is another client


LabCPU-1: GoGo POWER RANGERS!

LabCPU-4 has left the chat.

LabCPU-3 has left the chat.
```

```
[root@DataComm ~]# ./client file.txt
Saving chat to file: file.txt
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-3
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1
LabCPU-2 has joined the chat.
LabCPU-1 has joined the chat.

LabCPU-4: Hello World

This is CPU Number 3


LabCPU-2: this is another client


LabCPU-1: GoGo POWER RANGERS!

LabCPU-4 has left the chat.
```

```
[root@DataComm ~]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-4
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1
LabCPU-3 has joined the chat.
LabCPU-2 has joined the chat.
LabCPU-1 has joined the chat.
Hello World


LabCPU-3: This is CPU Number 3


LabCPU-2: this is another client


LabCPU-1: GoGo POWER RANGERS!
```

Figure 6. The server alerts other clients when a client disconnects


root@DataComm:~
File Edit View Search Terminal Help
root@DataComm:~
File Edit View Search Terminal Help

```
[root@DataComm ~]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-1
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1
```

```
[root@DataComm Downloads]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-2
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1
```

```
[root@DataComm ~]# ./client file.txt
Saving chat to file: file.txt
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-3
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1
```

```
[root@DataComm ~]# ./client
Not saving to a file.
Enter Host IP: 192.168.0.17
Enter Host Port: 7000
Enter Username: LabCPU-4
Connected:    Server Name: 192.168.0.17
               IP Address: 192.168.0.17
Joined server: Room 1
```

Figure 7. Clients can connect to the server.

How to compile the project

- open a new terminal in Linux(tested to work on a system running Fedora)
- navigate to folder containing the client.cpp file or server.cpp file using “cd”
- type in: “g++ -Wall -o client client.cpp” and/or “g++ -Wall -o server server.cpp” to compile the program
- check there are no errors and use “ls” to to see that the .exe has been created in the folder
- run the program by typing in ./client to start the client or ./server to start the server
- optionally the client program can be set to save the chat log to a file by running the program with ./client [filename].txt where [filename].txt is the name of the file you wish to create or overwrite with the chat log.

How to use the program

1. Start the server program.

2. Set the port that the server will be listening on (after the prompt)

3. Start up the clients

3. b. Optionally, start up the clients with a specified file to save the chat log too.

4. Set the IP and ports of the clients

5. Set the alias names of the clients (usernames)

6. Send messages to the server from the clients by typing in (enter sends the data).