LINUX CLIENT-SERVER DOCUMENTATION

COMP 4981 – LINUX SOCKETS

SPENSER LEE A00925785

THOMAS YU A00915638

Contents

User Manual

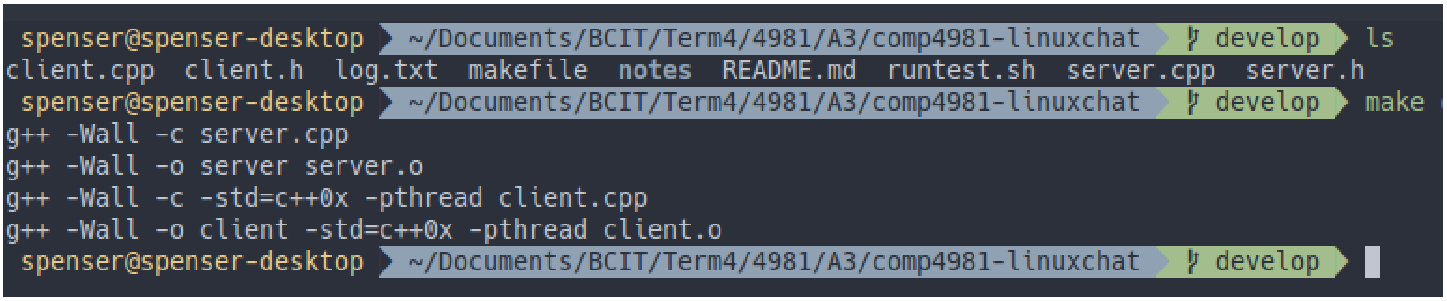
Application Overview

This application demonstrates the use of sockets and the select call with a server and client chat program. Multiple client connect to a server and send messages; each message sent to the server is then echoed to all other clients.

Application Setup

To compile the client and server executables:

1. Open a terminal and navigate to the folder containing the source files
2. Execute the command “make”



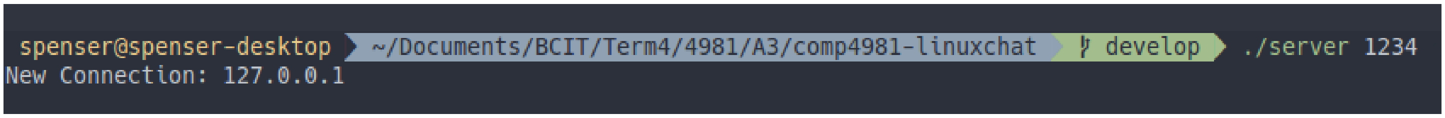
```
 spenser@spenser-desktop  ~/Documents/BCIT/Term4/4981/A3/comp4981-linuxchat   develop  ls
client.cpp  client.h  log.txt  makefile  notes  README.md  runtest.sh  server.cpp  server.h
 spenser@spenser-desktop  ~/Documents/BCIT/Term4/4981/A3/comp4981-linuxchat   develop  make
g++ -Wall -c server.cpp
g++ -Wall -o server server.o
g++ -Wall -c -std=c++0x -pthread client.cpp
g++ -Wall -o client -std=c++0x -pthread client.o
 spenser@spenser-desktop  ~/Documents/BCIT/Term4/4981/A3/comp4981-linuxchat   develop 
```

Application Usage

With the applications compiled you can run multiple clients and a server.

First launch a server, in a new terminal window execute the following command: **`./server [port]`**

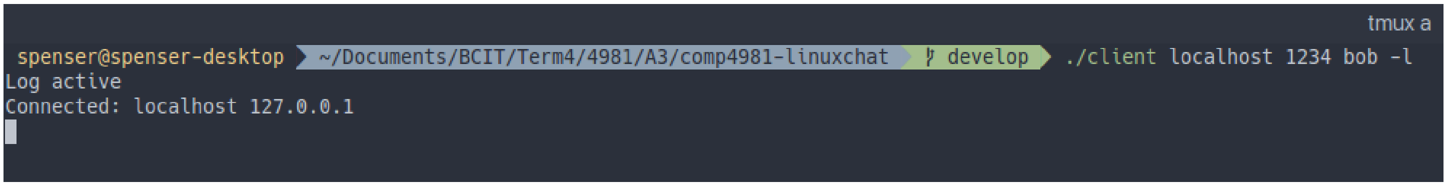
You can optionally choose a port to run on (default 7000).



```
 spenser@spenser-desktop  ~/Documents/BCIT/Term4/4981/A3/comp4981-linuxchat   develop  ./server 1234
New Connection: 127.0.0.1
```

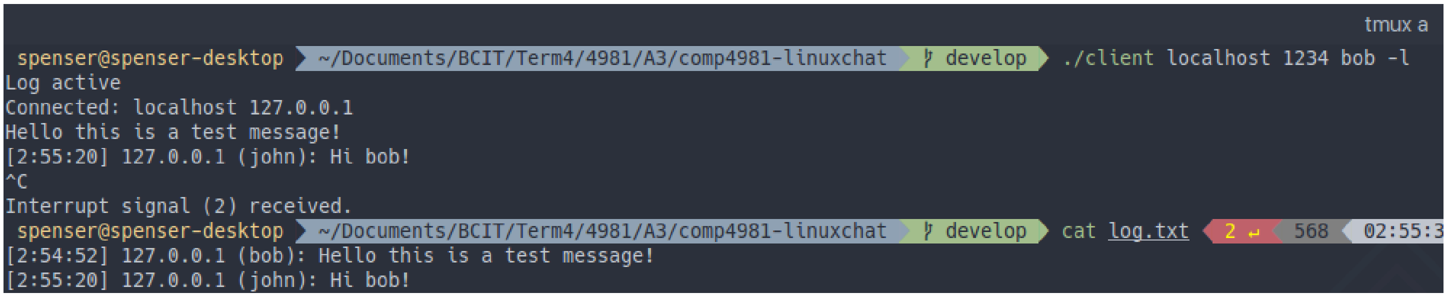
To launch a client, execute the following command: **`./client hostname [port] [username] [-l]`**

You must enter a hostname, and can optionally choose a port (default 7000), set a username, and enable logging with -l.



```
                                                                                           tmux a
 spenser@spenser-desktop  ~/Documents/BCIT/Term4/4981/A3/comp4981-linuxchat   develop  ./client localhost 1234 bob -l
Log active
Connected: localhost 127.0.0.1
```

With the client running, enter a message and press enter.



```
                                                                                           tmux a
 spenser@spenser-desktop  ~/Documents/BCIT/Term4/4981/A3/comp4981-linuxchat   develop  ./client localhost 1234 bob -l
Log active
Connected: localhost 127.0.0.1
Hello this is a test message!
[2:55:20] 127.0.0.1 (john): Hi bob!
^C
Interrupt signal (2) received.
 spenser@spenser-desktop  ~/Documents/BCIT/Term4/4981/A3/comp4981-linuxchat   develop  cat log.txt  2 ↵  568  02:55:3
[2:54:52] 127.0.0.1 (bob): Hello this is a test message!
[2:55:20] 127.0.0.1 (john): Hi bob!
```

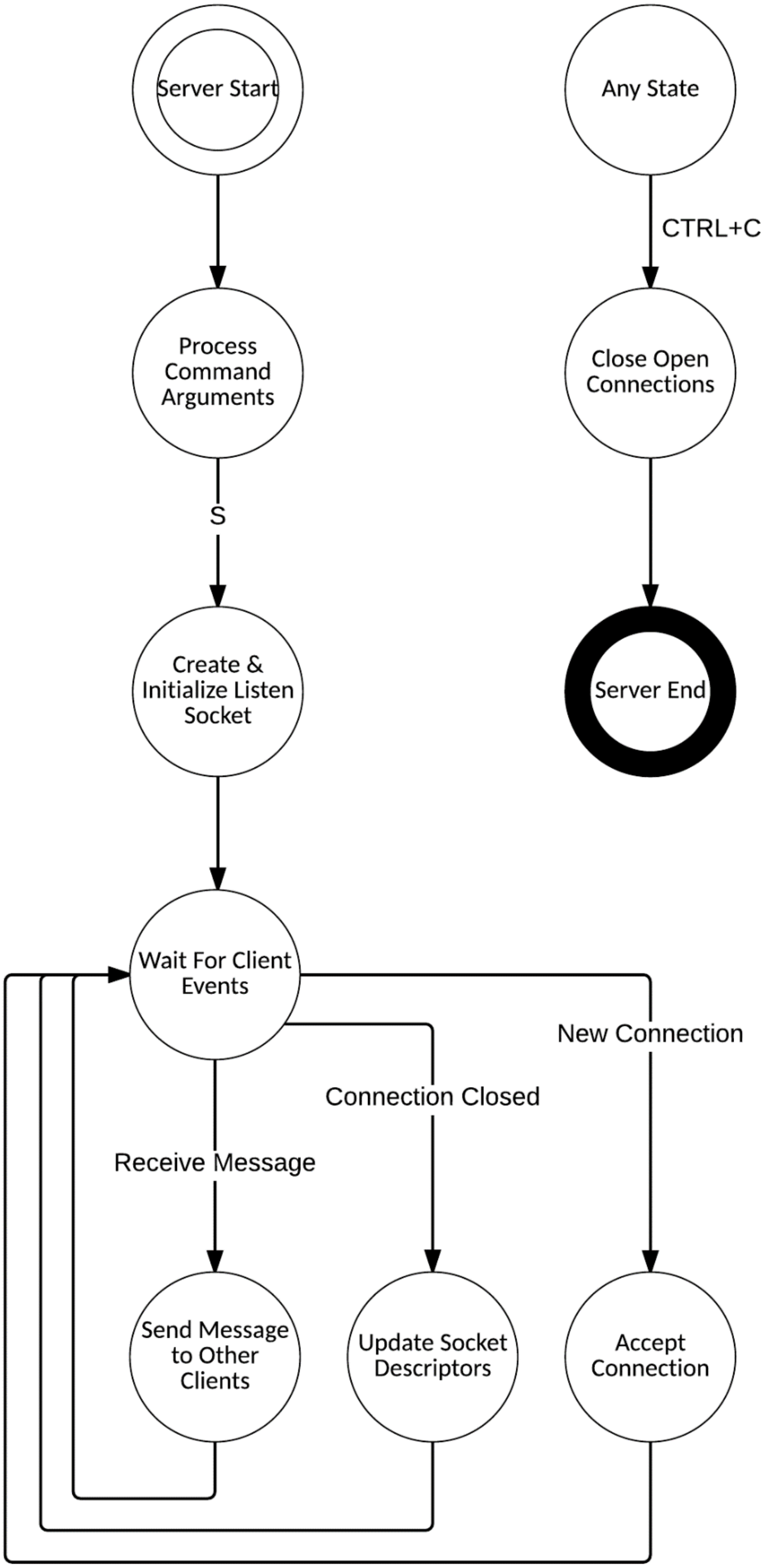
Bash Script For Testing

To aid with testing, I also created a script which will launch a server and 3 named clients applications on localhost.

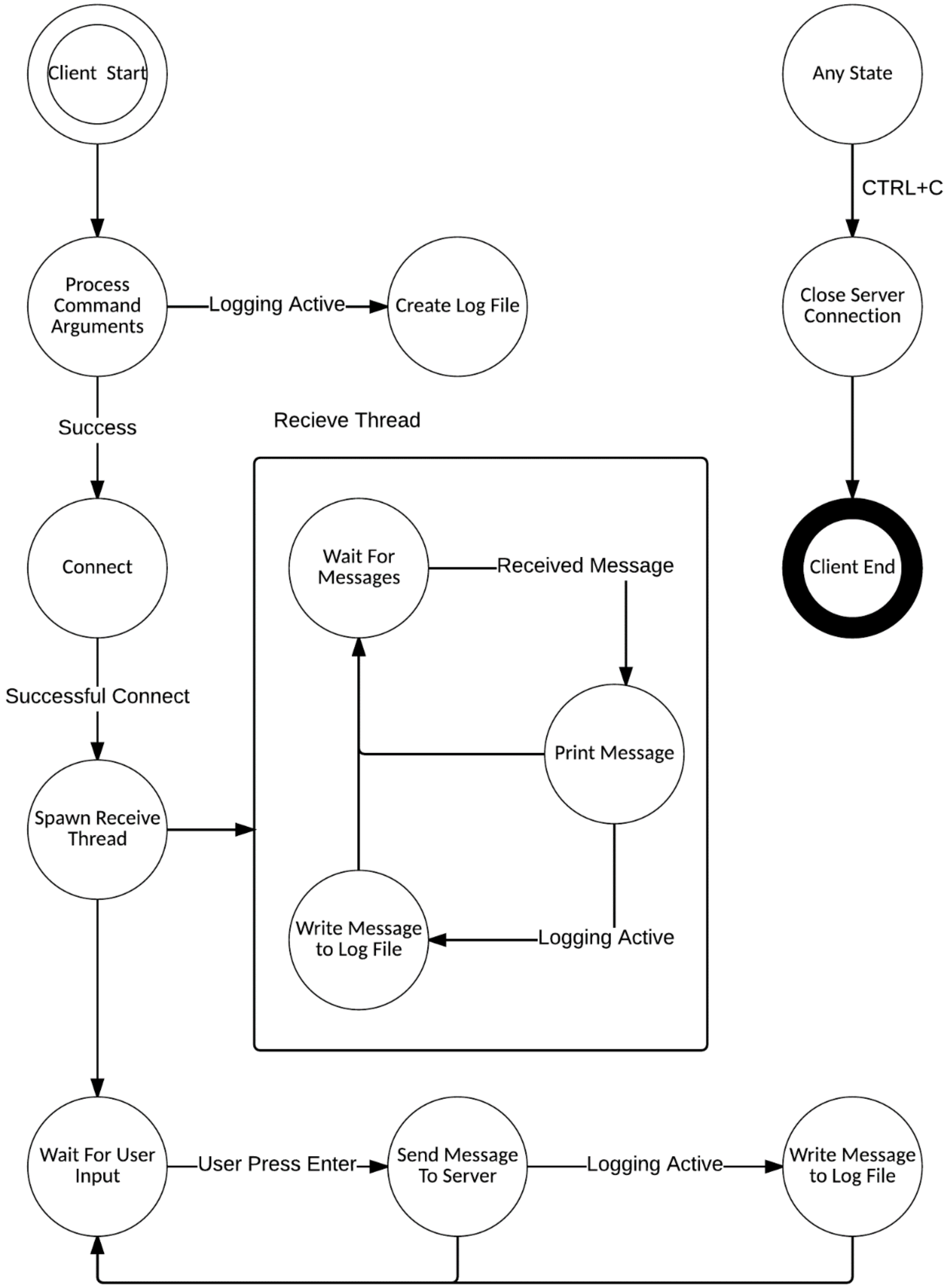
To execute it, run the following command in a terminal:

```
bash runtest.sh
```

State Chart Diagram (Server)



State Chart Diagram (Client)



Pseudo Code

Client

```
Process Command Arguments
    if invalid host
        return false
    if invalid port
        return false
    assign username
    if logactive
        set global switch
    return true

Create Log File
    if logactive
        open log file in append mode

Connect
    create tcp socket
    create sockaddr struct
    initialize socketaddr members
    create host pointer
    connect call
    go to Spawn Receive Thread

Spawn Receive Thread
    spawn receive thread
    go to Wait For User Input

Receive Thread
    loop forever
        loop on blocking recv call until buflen
            if nothing read
                exit program
        if logactive
            write received message to log
        print out received message to console

Wait For User Input
    loop forever
        read in line from stdin to message string
        append timestamp to message front
        if username set
            append username to message front
        if (message length <= buffer length)
            send message to server
            write message to log file
        else
            print message too long error

Any State
    if CTRL + C
        go to Close Server Connection
```

```
Close Server Connection
    close server socket descriptor
    if logactive
        close log file
    program exit
```

Pseudo Code (continued)

Server

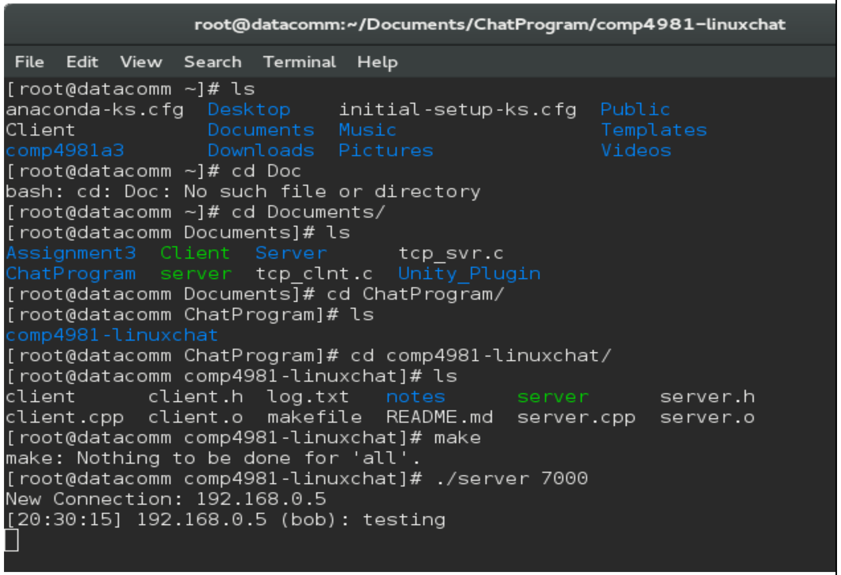
```
Process Command Arguments
    if invalid port
        print error message
        exit program
    go to Create & Initialize Socket

Create & Initialize Socket
    create TCP listen socket
    set socket options (reuse)
    bind listen socket
    listen for connections
    initialize client socket desriptor array
    go to Wait For Client Events

Wait For Client Events
    select call
    if FD_ISSET on listen socket
        go to Accept Connection
    loop through client array
        assign temp socket descriptor to current index in client array
        if temp socket > 0 and FD_ISSET
            if receive message
                go to Send Message To Other Clients
                if 0 bytes received
                    go to Update Socket Descriptors

Accept Connection
    accept new connection from listen socket
    print new connection
    update client array
    set accepted socket
    go to Wait For Client Events

Send Message To Other Clients
    loop through client array
        if FD_ISSET and not sender
            send recieved message
    go to Wait For Client Events
```

```
Update Socket Descriptors
    print connection closed message
    close socket descriptor
    FD_CLR temp socket
    update client array index
    go to Wait For Client Events

Any State
    if CTRL + C
        go to Close Server Connection

Close Open Connection
    close open sockets
    program exit
```

Testing

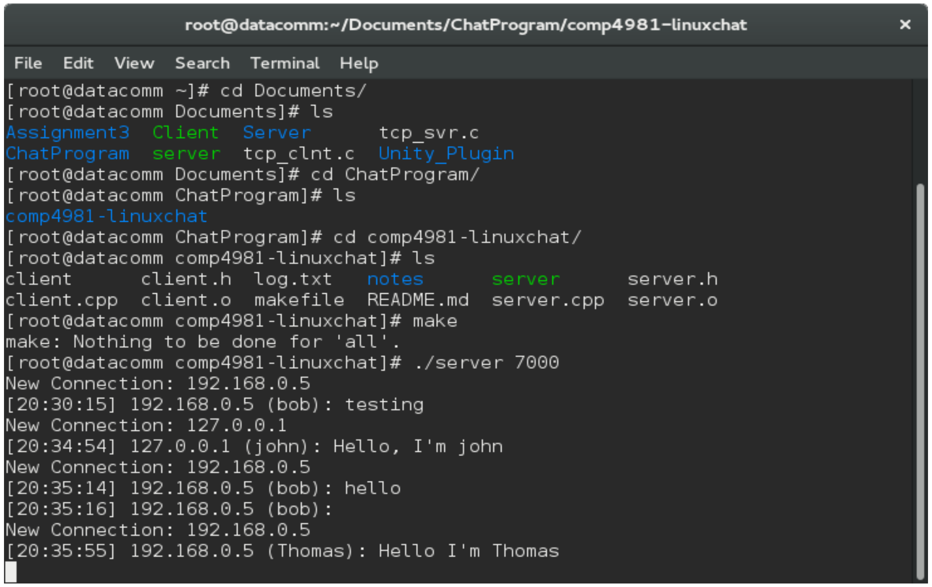
#	DESCRIPTION	STEPS	EXPECTED RESULT	ACTUAL RESULT	RESULT
1	Single client connects to the server	1. Create server 2. Create Client 3. Send a message to the server	No echo on client side, server displays the message from client.	No echo on client side, server displays the message from client.	PASS
2	Three clients connect to the server	1. Create server 2. Create Clients 3. Send a message from each client to the server	Each client sees the other clients' messages. Server displays the messages from all the clients.	Each client sees the other clients' messages. Server displays the messages from all the clients.	PASS
3	Client can reconnect to the server.	1. Create the server 2. Create client 3. Client disconnects 4. Client connects again. 5. Client sends a message.	No echo on client side, server displays the message from client.	No echo on client side, server displays the message from client.	PASS
4	Disconnect the server with clients	1. Create server 2. Create Clients 3. Disconnect server	The clients disconnect and the program exits with an error indicating the server has gone offline.	The clients disconnect and the program exits with an error indicating the server has gone offline.	PASS
5	Single client connects to the server with chat log enabled	1. Create server 2. Create Client with log enabled 3. Send a message to the server	No echo on client side, server displays the message from client. The log file displays the message	No echo on client side, server displays the message from client. The log file displays the message	PASS
6	Multiple clients connect to the server with chat log enabled	1. Create server 2. Create Clients 3. Send a message from each client to the server	Each client sees the other clients' messages. Server displays the messages from all the clients. Log file displays the messages	Each client sees the other clients' messages. Server displays the messages from all the clients. Log file displays the messages	PASS
7	Starting the Client/Server with wrong parameters	1. Start Client/Server with incorrect parameters	Error message is displayed indicating proper command line arguments.	Error message is displayed indicating proper command line arguments.	PASS
8	Client Sends a message with a size greater than 511	1. Create server 2. Create Client 3. Send a long message to the server	Client prints an error indicating the maximum message size.	Client prints an error indicating the maximum message size.	PASS

TEST 1


root@datacomm:~/Documents/ChatProgram/comp4981-linuxchat
File Edit View Search Terminal Help

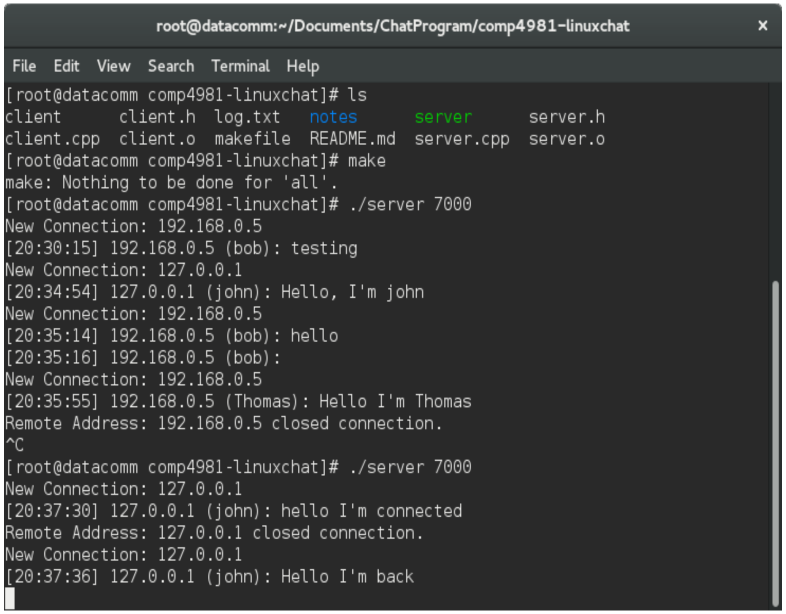
```
[root@datacomm ~]# ls
anaconda-ks.cfg  Desktop    initial-setup-ks.cfg  Public
Client           Documents  Music                 Templates
comp4981a3       Downloads  Pictures              Videos
[root@datacomm ~]# cd Doc
bash: cd: Doc: No such file or directory
[root@datacomm ~]# cd Documents/
[root@datacomm Documents]# ls
Assignment3  Client  Server     tcp_svr.c
ChatProgram  server  tcp_clnt.c  Unity_Plugin
[root@datacomm Documents]# cd ChatProgram/
[root@datacomm ChatProgram]# ls
comp4981-linuxchat
[root@datacomm ChatProgram]# cd comp4981-linuxchat/
[root@datacomm comp4981-linuxchat]# ls
client      client.h  log.txt   notes      server      server.h
client.cpp  client.o  makefile  README.md  server.cpp  server.o
[root@datacomm comp4981-linuxchat]# make
make: Nothing to be done for 'all'.
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 192.168.0.5
[20:30:15] 192.168.0.5 (bob): testing
```

TEST 2


root@datacomm:~/Documents/ChatProgram/comp4981-linuxchat
×
File Edit View Search Terminal Help

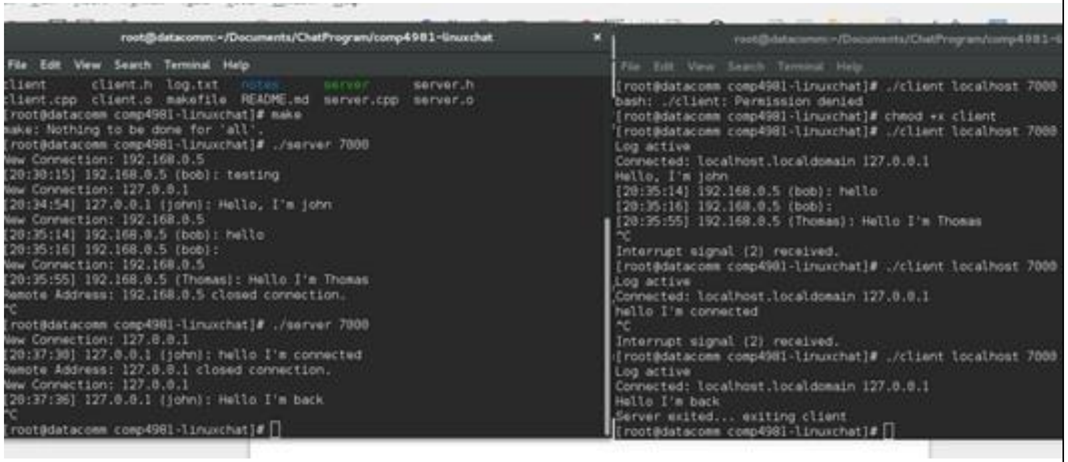
```
[root@datacomm ~]# cd Documents/
[root@datacomm Documents]# ls
Assignment3  Client  Server     tcp_svr.c
ChatProgram  server  tcp_clnt.c  Unity_Plugin
[root@datacomm Documents]# cd ChatProgram/
[root@datacomm ChatProgram]# ls
comp4981-linuxchat
[root@datacomm ChatProgram]# cd comp4981-linuxchat/
[root@datacomm comp4981-linuxchat]# ls
client      client.h  log.txt   notes      server      server.h
client.cpp  client.o  makefile  README.md  server.cpp  server.o
[root@datacomm comp4981-linuxchat]# make
make: Nothing to be done for 'all'.
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 192.168.0.5
[20:30:15] 192.168.0.5 (bob): testing
New Connection: 127.0.0.1
[20:34:54] 127.0.0.1 (john): Hello, I'm john
New Connection: 192.168.0.5
[20:35:14] 192.168.0.5 (bob): hello
[20:35:16] 192.168.0.5 (bob):
New Connection: 192.168.0.5
[20:35:55] 192.168.0.5 (Thomas): Hello I'm Thomas
```

TEST 3


root@datacomm:~/Documents/ChatProgram/comp4981-linuxchat
×
File Edit View Search Terminal Help

```
[root@datacomm comp4981-linuxchat]# ls
client      client.h  log.txt   notes      server      server.h
client.cpp  client.o  makefile  README.md  server.cpp  server.o
[root@datacomm comp4981-linuxchat]# make
make: Nothing to be done for 'all'.
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 192.168.0.5
[20:30:15] 192.168.0.5 (bob): testing
New Connection: 127.0.0.1
[20:34:54] 127.0.0.1 (john): Hello, I'm john
New Connection: 192.168.0.5
[20:35:14] 192.168.0.5 (bob): hello
[20:35:16] 192.168.0.5 (bob):
New Connection: 192.168.0.5
[20:35:55] 192.168.0.5 (Thomas): Hello I'm Thomas
Remote Address: 192.168.0.5 closed connection.
^C
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 127.0.0.1
[20:37:30] 127.0.0.1 (john): hello I'm connected
Remote Address: 127.0.0.1 closed connection.
New Connection: 127.0.0.1
[20:37:36] 127.0.0.1 (john): Hello I'm back
```

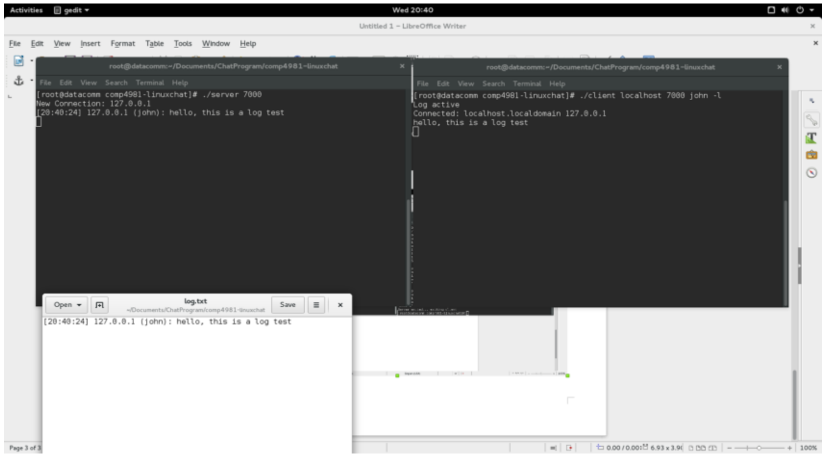
TEST 4


root@datacomm:~/Documents/ChatProgram/comp4981-linuxchat
×
File Edit View Search Terminal Help
root@datacomm:~/Documents/ChatProgram/comp4981-l
File Edit View Search Terminal Help

```
client      client.h  log.txt   notes      server      server.h
client.cpp  client.o  makefile  README.md  server.cpp  server.o
[root@datacomm comp4981-linuxchat]# make
make: Nothing to be done for 'all'.
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 192.168.0.5
[20:30:15] 192.168.0.5 (bob): testing
New Connection: 127.0.0.1
[20:34:54] 127.0.0.1 (john): Hello, I'm john
New Connection: 192.168.0.5
[20:35:14] 192.168.0.5 (bob): hello
[20:35:16] 192.168.0.5 (bob):
New Connection: 192.168.0.5
[20:35:55] 192.168.0.5 (Thomas): Hello I'm Thomas
Remote Address: 192.168.0.5 closed connection.
^C
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 127.0.0.1
[20:37:30] 127.0.0.1 (john): hello I'm connected
Remote Address: 127.0.0.1 closed connection.
New Connection: 127.0.0.1
[20:37:36] 127.0.0.1 (john): Hello I'm back
^C
[root@datacomm comp4981-linuxchat]#
```

```
[root@datacomm comp4981-linuxchat]# ./client localhost 7000
bash: ./client: Permission denied
[root@datacomm comp4981-linuxchat]# chmod +x client
[root@datacomm comp4981-linuxchat]# ./client localhost 7000
Log active
Connected: localhost.localdomain 127.0.0.1
Hello, I'm john
[20:35:14] 192.168.0.5 (bob): hello
[20:35:16] 192.168.0.5 (bob):
[20:35:55] 192.168.0.5 (Thomas): Hello I'm Thomas
^C
Interrupt signal (2) received.
[root@datacomm comp4981-linuxchat]# ./client localhost 7000
Log active
Connected: localhost.localdomain 127.0.0.1
hello I'm connected
^C
Interrupt signal (2) received.
[root@datacomm comp4981-linuxchat]# ./client localhost 7000
Log active
Connected: localhost.localdomain 127.0.0.1
Hello I'm back
Server exited... exiting client
[root@datacomm comp4981-linuxchat]#
```

TEST 5

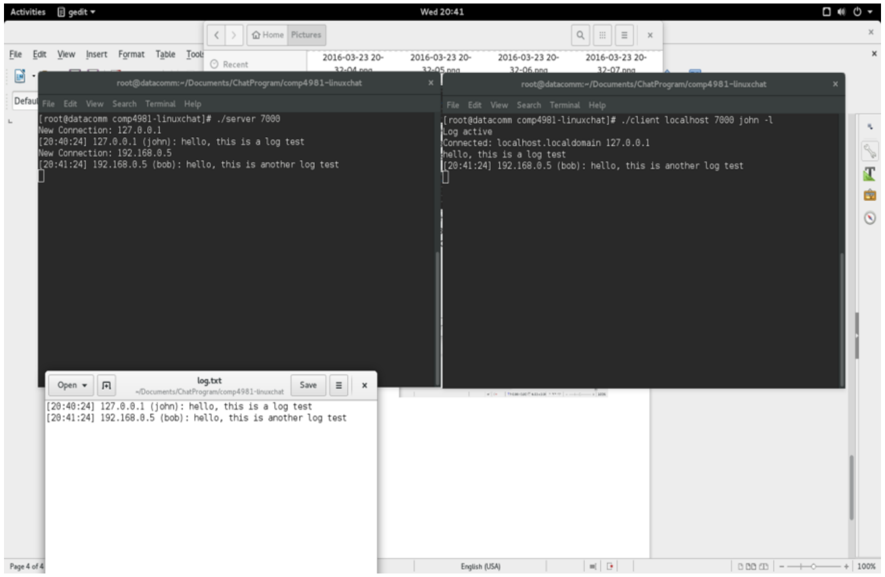

Activities
gedit
Wed 20:40
Untitled 1 – LibreOffice Writer
File Edit View Insert Format Table Tools Window Help
root@datacomm:~/Documents/ChatProgram/comp4981-linuxchat
File Edit View Search Terminal Help
log.txt
~/Documents/ChatProgram/comp4981-linuxchat
Open
Save

```
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 127.0.0.1
[20:40:24] 127.0.0.1 (john): hello, this is a log test
```

```
[root@datacomm comp4981-linuxchat]# ./client localhost 7000 john -l
Log active
Connected: localhost.localdomain 127.0.0.1
hello, this is a log test
```

```
[20:40:24] 127.0.0.1 (john): hello, this is a log test
```

TEST 6

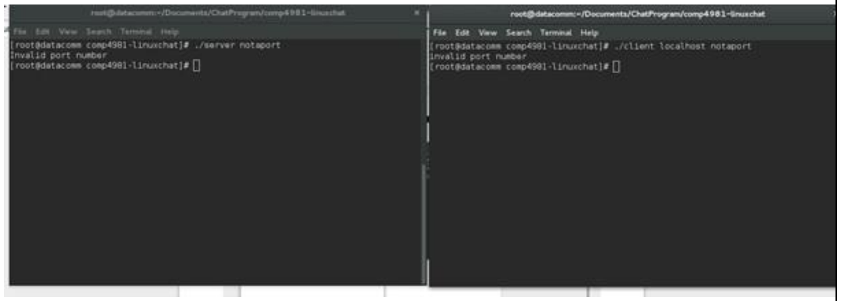


```
[root@datacomm comp4981-linuxchat]# ./server 7000
New Connection: 127.0.0.1
[20:40:24] 127.0.0.1 (john): hello, this is a log test
New Connection: 192.168.0.5
[20:41:24] 192.168.0.5 (bob): hello, this is another log test
```

```
[root@datacomm comp4981-linuxchat]# ./client localhost 7000 john -l
Log active
Connected: localhost.localdomain 127.0.0.1
hello, this is a log test
[20:41:24] 192.168.0.5 (bob): hello, this is another log test
```

```
[20:40:24] 127.0.0.1 (john): hello, this is a log test
[20:41:24] 192.168.0.5 (bob): hello, this is another log test
```

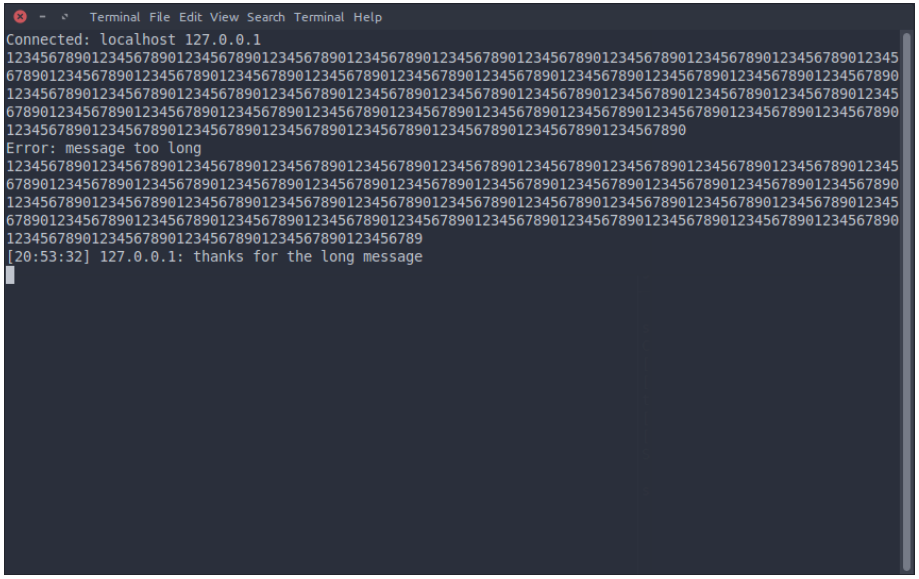
TEST 7


root@datacomm:~/Documents/ChatProgram/comp4981-linuxchat
File Edit View Search Terminal Help

```
[root@datacomm comp4981-linuxchat]# ./server notaport
Invalid port number
[root@datacomm comp4981-linuxchat]# 
```

```
[root@datacomm comp4981-linuxchat]# ./client localhost notaport
Invalid port number
[root@datacomm comp4981-linuxchat]# 
```

TEST 8



```
Connected: localhost 127.0.0.1
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Error: message too long
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890123456789
[20:53:32] 127.0.0.1: thanks for the long message
```