**Jeffrey Morton**

**Thanh Tran**

**System Network 2**

**Project 2**

**Setup and compilation**

1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.
2. The submission includes:

* makefile
* bbclient.c
* bbserver.c,
* bbutils.c
* bbutils.h
* /screenshot/server\_client\_success
* /screenshot/client\_write
* /screenshot/client\_exit\_read
* /screenshot/client\_exit\_all

1. Environment: The C program has been tested on a Windows 10 computer and a Linux virtual machine. The application will successfully run in these environment as well as the ssh server in the university.
2. Compilation: This program includes a makefile. At the command line in Linux simply type make and the program will produces an executable entitled ‘bbclient’ and ‘bbserver’.

**Running the program.** Issue the command ./bbserver. It takes two command line argument <server port number> <number of initial client>. Depending on the amount of initial client, open up multiple terminal and issue the command ./bbclient. It takes 4 arguments <the host name/address> e.g. localhost, a valid range 60001-60099 <a random port number>, <the server port number>, and <the name of the text file for the bulletin board>.

User input: At the console of the client will be a display for a menu asking for the user input. The user will be given 4 choices to interact with the program. It takes a character as input. Choice 1 allows the user to write a message to the bulletin board. Choice 2 lets the user reads a specific message off the bulletin board. Choice 3 list all the sequences of number of messages currently in the bulletin board. Choice 4 allows the client to exit the bulletin board and releases them from the token ring.

Output: All output goes to the console. Output will be similar to this:

On success(server)

Waiting for client info

Received client info

Received client info

Received client info

Sending client-neighbor info

Token:

<token>

3

localhost 60001

localhost 60002

localhost 60003

Winner! Sending first token

</token>

(Note: This will output once the all client’s info are sent to the server. A winner is picked on a first come first serve basis to give the token, the menu is display for the user to interact with the bulletin board)

On fail (server)

Binding failed with error number 98: Address already in use

(Note: occurs if the user uses the same address)

On success(client)

Sending client info to server

Sent 59 bytes, Waiting for server response

Launching userIO thread.

-->Enter w for Write operation!! Appends a new message to the end of the message board

-->Enter r for Read operation!! Read a particular message from the message board using a message sequence number. # is the sequence number of the message on the board.

-->Enter l for List operation!! Displays the range of valid sequence numbers of messages posted to the board.

-->Enter e for Exit operation!! Closes the message board. Exit requires that the user leaves the logical token ring.

(Note: The client that has the token is able to interact with the menu for as long as they want. Message written to the bulletin board are posted to the text file, if it already exit it appends the message)

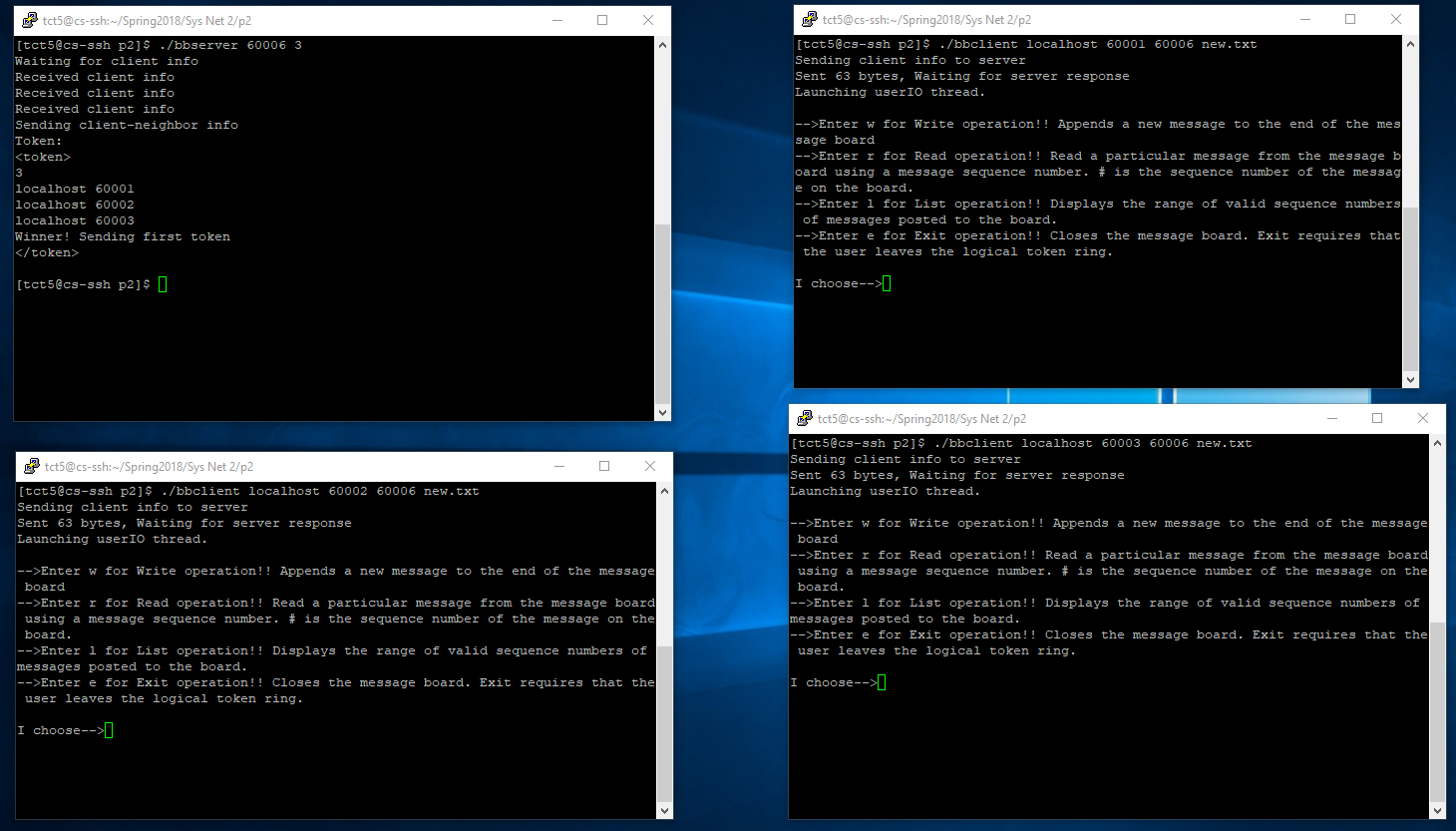
On fail (client)

Binding failed with error number 13: Permission denied

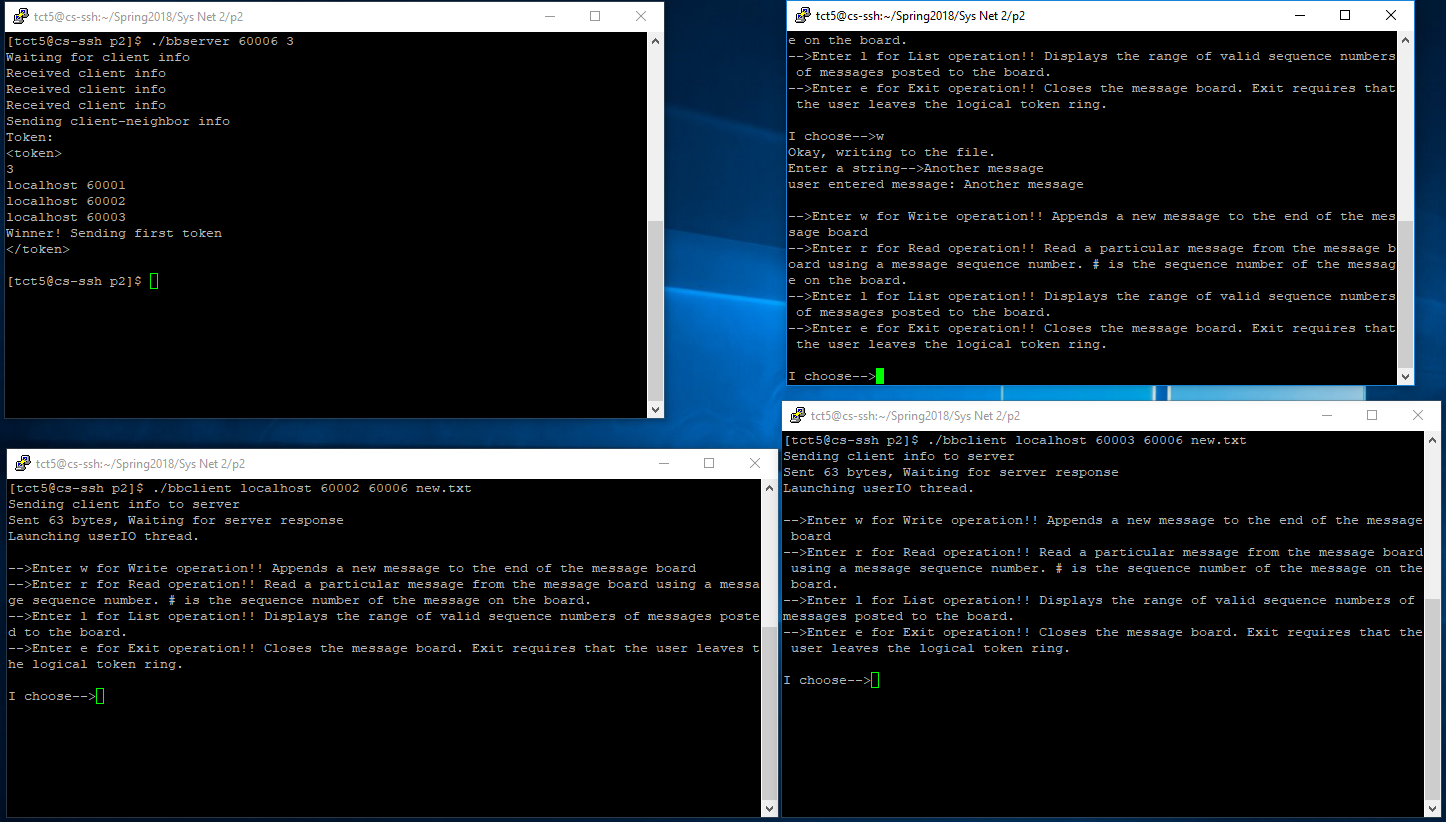
(Note: occurs if the user enter an invalid host/port number)

**Screenshot.**

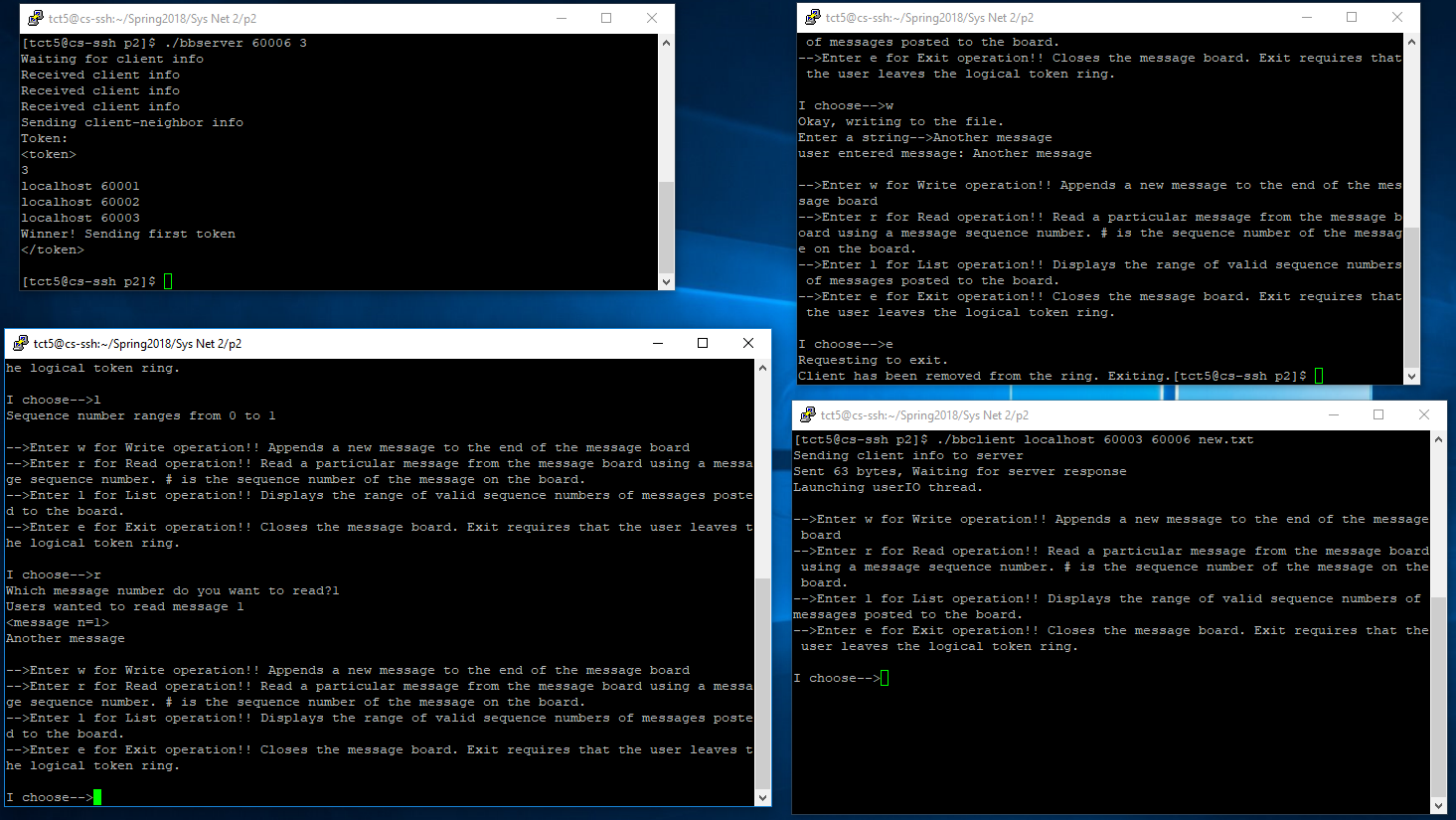
**Figure 1.** The text file new.txt is created along with 3 terminal which sends its info to the server to establish the token ring.



**Figure 2** The upper right window came first so it was chosen as the winner. It is issue a message write to the bulletin board creating(if it hasn’t been created)/appending to the text file.



**Figure 3.** The client in the upper right corner leaves and releases the token. Afterward the client at the bottom left picks up the token and issue a message read to get the message off the bulletin board.



**Figure 4.** The client in the bottom right leaves automatically once it is the only client left in the token ring.

