Jacob Rook

Systems & Networks II

Project 2

User's Manual

Setup and Compilation:

1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.
2. The includes:
   a. bbserverc
   b. bbserver.h
   c. bbpeer.c
   d. bbpeer.h
   e. makefile
   f. UsersManual_README.docx
   g. Protocol_Document.docx
3. Environment: This project was created using on a combination of the Eclipse and jGrasp IDE. However, the project was mainly tested on a personal Linux VM.
4. Compiling. This program includes a makefile. At the command line in Linux in the folder with the extracted files, type make into the command line. The program produces executables: bbserver and bbpeer.

Project Summary:

The Token Ring Bulletin Board Program is a peer to peer network that requires a server to link the peers into a ring. Once the peers are placed in the ring, the server terminates and does not assist the peers in the ring further.

The peers pass a token to be able to access a local file to read from and write to. Whenever a peer has the token, they are allowed to write to the local bulletin board file, read specific messages from the local bulletin board file, list all of the messages from the local bulletin board file, or exit the ring. Note, the peers are only allowed to exit when they have the token.

After the ring of peers has been initiated, new peers can join the ring by contacting one of the peers in the ring and the joining peer will receive the address of their next peer in the ring from one of the existing peers. It is also worth noting that message passing only go in one direction.

Server Start-up:

Obviously, the server will need to start up before the peers try to contact the server. The server program takes two command line arguments, the server's port number and the number of

peers to accept before sending off the peer's next peer socket address. The command should look like:

./bbserver <portNum> <numberHosts>

The server will bind a socket with the given port address then wait for peers to send the server a message to join a ring.

The peers will continue to contact the server until the number of peers the server needs to start the ring contact the server. Once the specified number of peers have contacted the server, the server stops accepting joining peers and sends the peers the socket address of their next peer in the ring.

Peer Start-up:

This implementation of the Token Ring Bulletin Board program only allows process to process communication, meaning that the server and all of the peers need to be on the same host. Since the server and peers will be on the same host, the IP addresses of the hosts and peers are not needed for communication. Also, the peers can share the same local bulletin board file. The general command to start a peer is:

./bbpeer [-new] <PortNum> <HostPort> <FilenameBulletinBoard>

The PortNum command line argument is the integer port number of the server or peer of an existing ring, the HostPort command line argument is the integer port number the host would like to have bound to the host's socket, and the FilenameBulletinBoard command line argument is the name of the bulletin board file that will be used to read and write to.

Peers can either contact a server that is starting a new ring or a peer of an existing ring. Either way, the message that they receive will be the same, i.e. the socket address of their next peer in the ring. The –new option distinguishes between a peer wanting to contact a server and a peer of an existing ring.

If a peer wants to contact a server, the user includes the –new option. If a peer wants to contact a peer of an existing ring, the user does not include the –new option.

Server User Input:

The server does not take any input from the user other than the command line arguments described above. Once the server is started, the server will do all of the work and terminate by its self.

Peer User Input:

After the ring has been initiated, the user will be prompted to choose between four options: write, read, list, or exit. The prompt will look like:

****

Enter w to write to the bulletin board

Enter r to read an entry from the bulletin board

Enter l to list all entries from the bulletin board

Enter e to exit the bulletin board ring

The message format is as follows:

<message n=sequence Number

<body>

</message>

****

Please enter one of the options above:

The user will enter one of the options and will be asked additional information if it is needed for the option. The write option will ask the user to enter the message that they would like to write to the bulletin board file. The read entry will ask the user which message the user would like to read from the bulletin board file. The list and exit options do not ask for any additional information.

NOTE: The server and all of the peers in the ring must be on the same host in the same directory. This project does not handle multi-host communication.

Running Program Captures:

       The following screen captures show the Token Ring Bulletin Board project working properly. The tests were done on the UWF's SSH server.

       Figure 1 shows the peers contacting the server to create the ring. The server, in the top left window, is started with the port number of 60001 and requiring 3 peers to join before the ring can be created. Peers with the port numbers 60002 and 60003, in the top right and bottom left respectively, send the server a message. The server receives these messages and displays them, still waiting for the third peer to contact the server.

Figure 1. Peers contacting the Server

Figure 2 shows the third peer contacting the server, allowing the server to create the ring. The server sends each peer its next peer in the ring then shuts down and the peers enter in the initiation process.

The peer with the port number 60002, in the top right window, has the smallest port number and is determined to be the initiator of the ring. The initiator then sends out a notification to its next peer to inform the peer that the initiator has been found then creates the bulletin board file and sends off the first token.

The peer with the port number 60003, in the bottom left window, receives the notification that the initiator has been found and forwards the notification to the next peer. The peer with the port number 60004, in the bottom right window, then receives the notification that the peer with port number 60003 sent. Thus, all of the peers in the ring know that the initiator has been found and the initiation process ends.
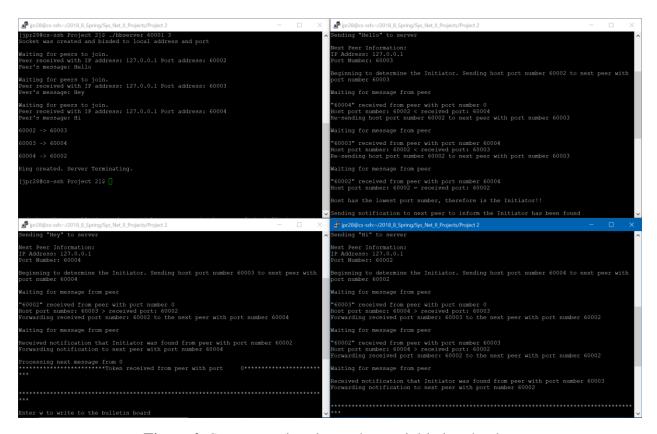

Figure 2. Server creating ring and peers initiating the ring

Figure 3 shows the peer in the bottom left window writing to the bulletin board file. The user enters the command to write to the bulletin board file, enters the message they would like to write to the file, and waits to receive the token, then writes the message to the bulletin board file.
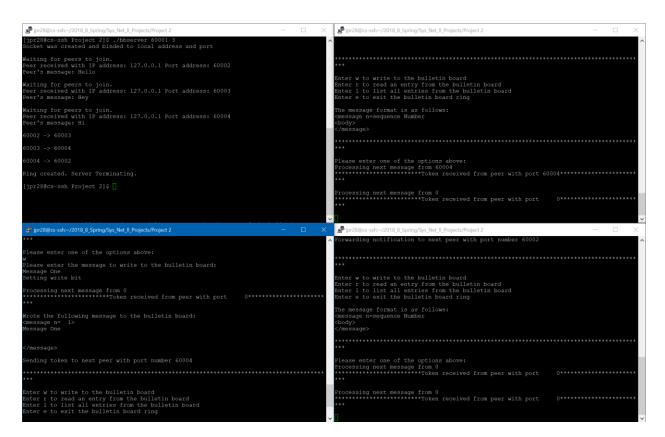

Figure 3. Peer in bottom left window writing to bulletin board file

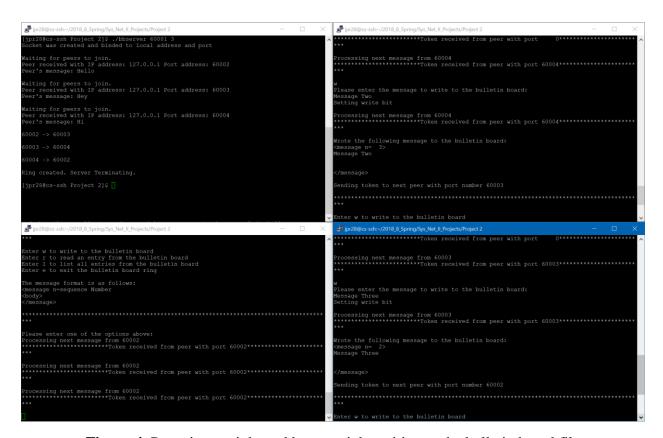Figure 4 shows the peers in the top right and bottom right going through the process of writing more messages to the bulletin board file.

Figure 4. Peers in top right and bottom right writing to the bulletin board file

Figure 5 shows the peer in the bottom left window requesting to list all of the messages in the bulletin board file. The peer sets the list bit then waits to receive the token to read from the file.
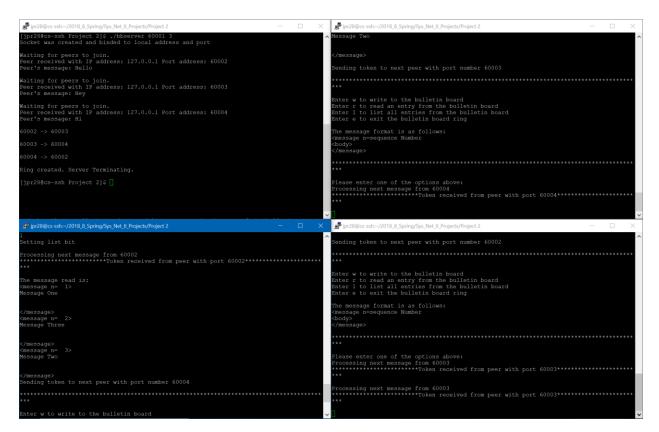

Figure 5. Peer in bottom left requesting to list all of the messages in the bulletin board file

Figure 6 shows the peer in the bottom left window requesting to read the second message in the bulletin board file. The peer waits to receive the token, then reads the second message from the bulletin board file.
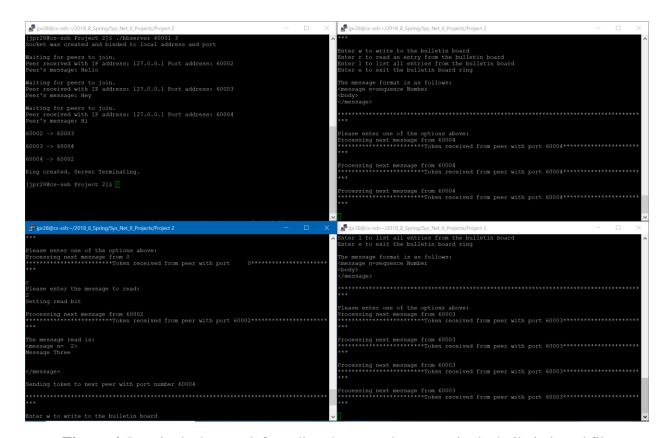

Figure 6. Peer in the bottom left reading the second message in the bulletin board file

Figure 7 and Figure 8 show the process of the peers in the bottom left and bottom right wanting to exit simultaneously. Since the peer in the bottom left window comes before the peer in the bottom right window, the peer in the bottom left window gets to exit before the other because the peer in the bottom left window receives the token before the other.

Figure 7 shows the peer in the bottom left sending an exit notification with its port number and the port number of its next peer to its next peer. The peer in the bottom right window, already having declared wanting to exit, receives the exit notification of the peer in the bottom left window. Since the peer in the bottom right has not received the token yet, it must wait to exit the ring and forwards the exit notification to its next peer. The peer in the top right window receives the forward exit notification for the peer in the bottom left and forwards the exit notification to the peer in the bottom left window.

The peer in bottom left window then sends the peer in the bottom right the token and exits the ring and terminates. The peer in the bottom right window receives the token and starts its exiting process as shown in Figure 7 and Figure 8. The peer in the bottom right window follows the same process as above and exits the ring leaving the peer in the top right to be the only peer in the ring.
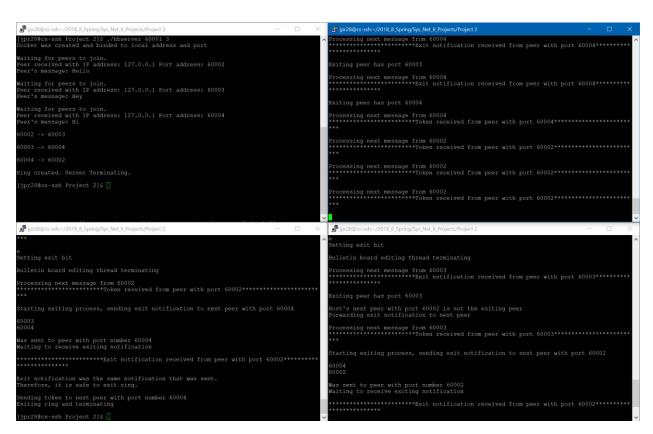

Figure 7. Peers in the bottom left and bottom right wanting to exit simultaneously
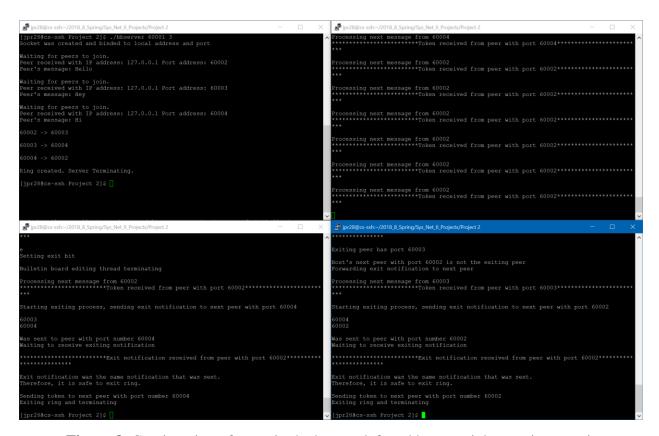
Figure 8. Continuation of peers in the bottom left and bottom right wanting to exit simultaneously

Figure 9 and Figure 10 show the process of a peer joining an existing ring. The existing ring consists of only one peer, the peer in the top right window. Figure 9 shows the peer in the bottom right window contacting the peer in the top right window with a join request. The peer in the top right receives this join request and sends the joining peer the socket address of its self. The joining peer receives this socket address and joins the ring.

Figure 10 shows the same process, but the peer in the bottom left window is joining this time.
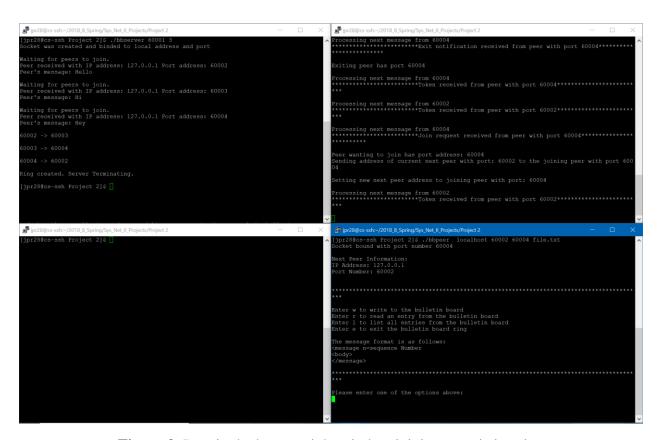

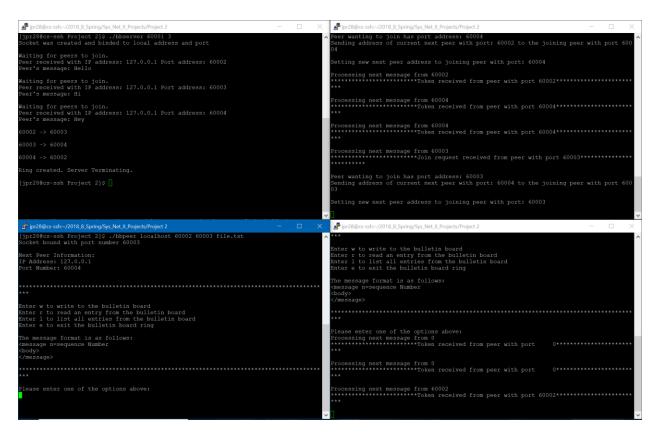**Figure 9.** Peer in the bottom right window joining an existing ring

Figure 10. Peer in the bottom left window joining an existing ring