CMSC 621: Project 2

Clocks, Multicast, and COMMIT

Sakshi Umesh Reddy – YO19530

I have written make file to compile all the codes and delete all the object files.

Assignment 1:

In this assignment, I have written 2 programs. First berkserver.cpp and second berkclient.cpp. Steps to execute the Berkeley's Algorithm are as following:

1. Open two windows Terminal-1 and Terminal-2 for Server and Client respectively
2. Compile the code on Terminal-1 by using command: make
3. Run the server on Terminal-1 by using: ./server <portnumber>
4. Enter the number of machines to be connected
5. On Terminal-2 run the client: ./client <portnumber>
6. Client program can be executed from multiple terminals simultanueously to have more than one client participating in the algorithm. **(Need to mention the correct number of clients to the server beforehand)**

Here the server is the time coordinator, and it collects time difference of other machines(clients) and send them the required adjustment to add to their logical clock.

Assignment 2:

For this assignment, there are 2 programs written - for Causal Ordered Multicast messages and for Non Causal Ordered Multicast messages. Steps to execute these Algorithms are as following:

1. Open 4 terminals to execute this program.
2. Compile the code on Terminal-1 by using command: make
3. Run the program on Terminal-1,2,3,4 by using: ./causal <portnumber1,2,3,4> and give the process number as 1,2,3,4 respectively.
4. **Give the input in order -** first enter the port number of 4th process on process 1 terminal, then on the process 2 terminal, at last on process 3.

5. Similarly, after the 4th process's port number is given for all other processes. Give port number of 3rd process on process 1 terminal, then on the process 2 terminal. (in the same ordered as mentioned)
6. After step 5, enter port number of 2nd process on process 1 terminal.
7. Enter 1 to multicast the messages at each process terminal respectively.

Same steps can be used to execute the Non Causal Ordered Program by executing: ./noncaual <portnumber1,2,3,4>

Limitation of these programs is that it only works for 4 processes.

Assignment 3:

In this assignment, there are 2 programs.

1. server_bonus.cpp – It acts as a coordinator who allows clients to access the critical section file
2. client_bonus.cpp – It is a process requesting to the server for access to increment the counter present in critical section file.

To execute this assignment, follow the given steps:

1. Open two windows Terminal-1 and Terminal-2 for Server and Client respectively
2. Compile the code on Terminal-1 by using command: make
3. Run the server on Terminal-1 by using: ./server1 <portnumber>
4. Enter the number of processes requesting for access
5. On Terminal-2 run the client: ./client1 <portnumber>
6. Client program can be executed from multiple terminals simultanueously to have more than one client participating in the algorithm. **(Need to mention the correct number of clients to the server beforehand)**

To delete all the object files, execute the command: make clean