CS/CE/TE 6378: Advanced Operating Systems
Section 0U1
Project 1

Instructor: Neeraj Mittal

Assigned on: Tuesday, June 10, 2014
Due date: Thursday, July 3, 2014

This is an individual project. *Code sharing among students is strictly prohibited and will result in disciplinary action being taken.*

You can do this project in C, C++ or Java. Each student is expected to demonstrate the operation of this project to the instructor or the TA. Since the project involves socket programming, you can only use machines netXX.utdallas.edu, where XX $\in$ {01, 02, .., 45}, for running the program. Although you may develop the project on any platform, the demonstration has to be on netXX machines; otherwise, you will be assessed a penalty of 20%.

1 Project Description

Implement a totally ordered multicast service. Your service should provide two calls to the application: m-send and m-receive. The first call m-send allows an application to send a message to a subset of processes. The second call m-receive allows an application to receive a multicast message; it is blocking and returns only when a multicast message is available for delivery.

Implementation Details: Design your program so that each process consists of two separate modules–one module that implements the application (sends and delivers messages) and one module that implements the totally ordered multicast service (guarantees that delivered messages satisfy total ordering condition). Intuitively, the two modules interact using m-send and m-receive function calls. Each module in turn may be implemented using one or more threads.

Testing: Design a mechanism to test the correctness of your service, that is, to ascertain that the total message ordering condition is satisfied. Your mechanism should be as *automated* as possible and should require minimal human intervention. *For example, visual inspection of log files to verify the correctness of the execution will not be acceptable.* You will be graded on how accurate and automated your testing mechanism is. Bonus points may be given for comprehensive and well thought out testing approach.

2 Submission Information

All the submission will be through eLearning. The submission details are as follows:

- **Thursday, June 19, 2014:** Submit a design document that describes the architecture of your program: what modules it will have and how they will interact with each other. Also submit details of your plan for testing the correctness of your program.
- **Thursday, July 3, 2014:** Submit all the source files necessary to compile the program and run it. Also, submit a README file that contains instructions to compile and run your program.