**ECE 6102 Take-Home Exam 1**

Due Date: March 3, 2014; 3:05 PM

**Instructions:** This exam is to be completed on your own without discussing it with any one other than the Professor. You may use any reference materials you like in completing the exam but you must list the materials that you consulted below. It is not necessary to list class notes nor the papers distributed to the class as reference materials.

I, ____________________________, hereby state that I did not discuss this exam with any person other than Professor Blough and that I used only the following reference materials in preparing my solution:

__________________________
(Signature)

__________________________
(Date)
1) 25 points
Consider a distributed system having 2 processes, P0 and P1, with 7 sequential events on each process. All pairs of events, where one event comes from each process, are concurrent, i.e. events are ordered only within one process but can be arbitrarily ordered between processes.

   a) How many distinct global states are there in this system, beginning from the initial state where no events have occurred and ending with the final state where all 14 events have occurred? Two states reached through the same sequence of events, but in different orders, are not considered distinct states. For example, the state reached by event 0 on P0 followed by event 0 on P1 is the same state reached by event 0 on P1 followed by event 0 on P0.

   b) Generalize your answer to the case where the number of sequential events on each of the two processes is any value k, i.e. come up with a general formula for your answer for any number of events.

For both parts, make sure to show how you calculated your solutions – answers without sufficient work will not receive credit.

2) 25 points
Consider the Paxos algorithm and assume it is run without a distinguished proposer. This means that multiple proposers are free to initiate proposals at any time. Furthermore, assume that the algorithm is run in a synchronous system.

   a) The synchronous assumption is not enough to guarantee that the Paxos algorithm (exactly as described in the paper but with multiple proposers) will make progress. Why not?

   b) Modify the Paxos algorithm to guarantee progress with multiple proposers in a synchronous system. You must keep the basic idea of proposers that make proposals, acceptors that consider different proposals and accept one, and a value that is chosen when a majority of acceptors have accepted it. However, you can change anything you want about how the proposers make their proposals and how the acceptors decide when and how to accept a proposal.

   c) Compare your algorithm to the Paxos algorithm with a distinguished proposer. Which approach do you think is simpler and more efficient? Why?

3) 25 points
Describe a different causal broadcast algorithm than the one presented in class. When sending a message, the algorithm presented in class sends the full content of all other messages that the current message causally depends on at the same time. As pointed out during discussion, this will require a lot of network bandwidth since the size of each message could potentially become quite large. The algorithm you design should not send the full contents of the other messages with the dependent message. It is permissible to include information about the other messages but not their payload (data portion). Your algorithm should still deliver messages in causal order. Make the algorithm as efficient as you can, meaning it should maintain as little state as possible in order to determine when it is safe to causally deliver a particular message. Describe the algorithm in pseudo-code. You can also include a paragraph describing in words the basic operation of the algorithm and the data structures it uses.
4) 25 points

Design and write pseudo-code for a RESTful implementation of Paxos. Your design should include clients containing values that they submit to be considered for proposing, a distinguished proposer that chooses a value to propose at each round of the algorithm and executes the two-phase proposal process described in the paper, acceptors that follow the behavior of acceptors described in the paper, and a distinguished learner that collects accepted values and decides when a value is chosen. You do not need to include multiple learners, only one distinguished learner. You do not need to deal with failures of the distinguished proposer and learner. However, the protocol should not assume that there is only one proposed value under consideration, i.e. the full propose and accept operation as described in the paper must be included. As a RESTful design, all services that you include must be stateless, and clients and services must interact with each other only using shared resources.

Show a block diagram of your design clearly showing which components are clients, which are services, and which are shared resources. The block diagram should also indicate the interactions between different components. Also write a paragraph describing in words the basic organization and operation of your design.

Finally, write pseudo-code for each component (clients, services). As a RESTful design, make sure that your code only uses HTTP get, put, post, and delete operations for all interactions between components. Again, make sure that services are stateless and that they only record information in shared resources.