

ECE 6102 Exam 1 (Take-Home Exam)

Due Date: March 9, 2009; 3:00 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.

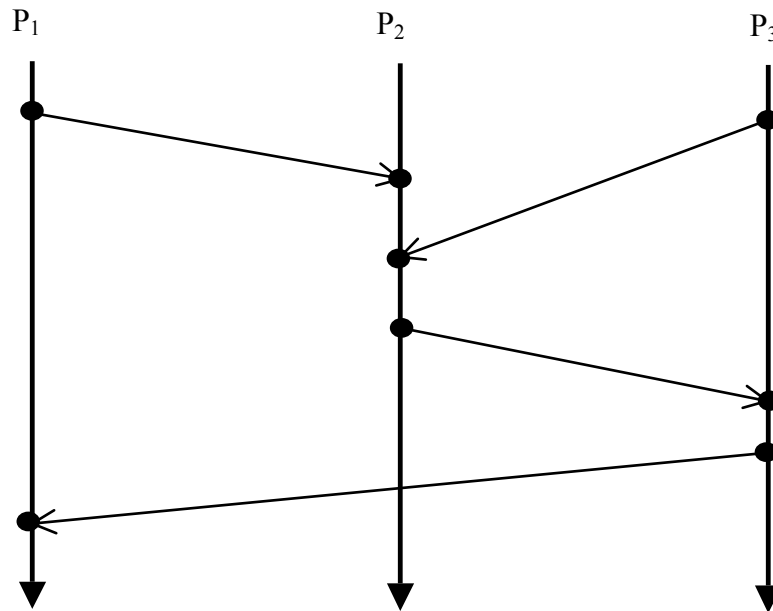
I, _____, hereby state that I did not discuss this exam with any person other than Professor Blough and that I used only the following outside reference materials in preparing my solution:

(Signature)

(Date)

1) 25 points

How many distinct executions are there for the distributed system having 3 processes and events as shown in the figure below? Assume that the only events are sending and receiving of messages and that non-blocking message sending is used. Justify your answer.



2) 25 points

As discussed in lecture, a *global state* of a distributed system is comprised of the states of all processes *and* the states of all channels. Because of inherent delays in sending messages between processes, it is difficult to take a *synchronous* checkpoint of a global state, where all process and channel states are recorded at a single instant. If instead, we allow processes to checkpoint their states and their channels' states *asynchronously*, then an inconsistent state might be recorded. Suppose that process A sends a message *m* to process B at some point in an execution. One inconsistent global state is one in which process A records its state prior to sending *m* and process B records its state after receiving *m* (when these two states are combined into a global state, process A has received a message that was not sent!). Another inconsistent global state is one in which process A records its state immediately after putting the message in the channel and records the channel state as containing *m* and B records its state *after* receiving *m* (when combining these states, the channel state indicates that B has not received *m*, while B's state indicates that it has received *m*).

Design a simple distributed algorithm that allows processes to checkpoint their states and the channels' states asynchronously, but is guaranteed to produce a consistent global state when combining all of the individual states. Specify your algorithm in C-like or Java-like pseudocode and also explain the basic idea of your algorithm in words. The algorithm will be graded both on correctness and simplicity, i.e. try to avoid overly complex solutions.

Hint: You might want to have processes record the states of their *incoming* channels. A process that puts a message on an outgoing channel does not know when that message is received, so it is difficult for the process to ensure that the channel state it records is consistent with the receiving process' state.

3) 25 points

In this problem, you do not need to consider the performance of your solution(s), only correctness. Assume a multiple-ring configuration where only crash faults occur, similar to the situation considered in the Totem paper. Present your solution(s) using pseudo-code similar to the way multicast protocols were described in the lecture.

- a) Describe an *atomic multicast* protocol that works based solely on local sequence numbers and ring IDs (i.e. there are no global timestamps used). Your solution should work for an arbitrary number of rings.
- b) Now, assume that there are only two rings connected by a single gateway. Is it possible to design an algorithm that generates monotonically increasing local sequence numbers in such a way that sequence numbers and ring IDs alone are sufficient to achieve *causal atomic multicast*? If so, sketch an algorithm to achieve it. If not, explain clearly why not. Explain how your result changes, if at all, when there are three or more rings.

4) 25 points

Consider a Pastry network with $b = 2$ (base = $2^b = 4$), $k = |L| = |M| = 4$, and 12-bit node IDs. Assume the proximity metric is number of hops and that there are 10 nodes with the following symmetric proximities to each other:

Node No.	Node ID	Dist. To 1	Dist. To 2	Dist. To 3	Dist. To 4	Dist. To 5	Dist. To 6	Dist. To 7	Dist. To 8	Dist. To 9
0	011332	4	9	2	13	7	5	12	11	1
1	031210		7	3	15	8	2	13	10	4
2	033221			8	8	12	6	6	15	8
3	101213				14	5	5	10	9	3
4	123310					5	14	1	6	13
5	220131						10	5	4	8
6	232110							11	9	3
7	321332								9	12
8	323021									7
9	323202									

- a) Assume that only nodes 1 to 9 are currently in the network and they have perfect state tables. Show the state table of node 9.
- b) Now, assume that node 0 joins the network. Show the route that is used to lookup node 0's ID. Also, show how node 0's state table is initialized at the end of the lookup. How does the initial table differ from the perfect one?