CS 5800 Distributed OS - Spring 2017
Homework 3

Josh Herman
Dalton Cole
Neil Blood

March 7, 2017

Problem 1 How to define causality in a distributed system?

Causality helps identify sequential and concurrent events without using physical clocks. If **a**, **b** are two events in a single process **P**, and the time of **a** is less than the time of **b** then $\mathbf{a} \prec \mathbf{b}$. If **a** = sending a message, and **b** = receipt of that message, then $\mathbf{a} \prec \mathbf{b}$. $\mathbf{a} \prec \mathbf{b} \wedge \mathbf{b} \prec \mathbf{c} \Rightarrow \mathbf{a} \prec \mathbf{c}$.

Problem 2 Can the simple counter based logical clock be used to detect causality? Why or why not?

No because the logical clocks might not be in sync so even if process b happened before process a, the logical clock of b could be larger than a. Even if they are in sync process a could be unrelated to process b making them having concurrency, absence of causal order. Therefore $LC(a) < LC(b)$ does not imply $a \prec b$.

Problem 3 According to Figure 1, there are three processes and their local events are denoted by $a, b, ..., m$. Provide the vector clock values for these events.

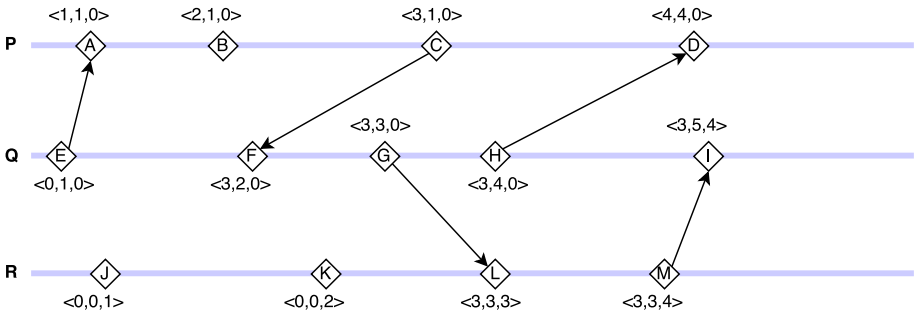

Figure 1: Events related to three processes: P, Q and R

Problem 4 Regarding the Lamport and Melliar-Smith's internal clock synchronization algorithm, using induction, prove that the maximum difference between the averages computed by two non-faulty modes is $\frac{3t\delta}{n}$, where t is the number of faulty clocks and n is the total number of clocks in the system. You can assume $t < n - 1$.

Problem 5 Refer to slide #5-6 of "lec6.pdf", what is the fairness property? Does the fairness property hold? Why or why not?

The fairness property in this scenario would be that processes with a lower time stamp will be able to enter the critical section first.
The fairness property does NOT hold because the queue adds requests as they come in with no priority on them. A possible solution would be to implement a priority queue instead of a normal queue that was sorted based on smallest time stamp.