

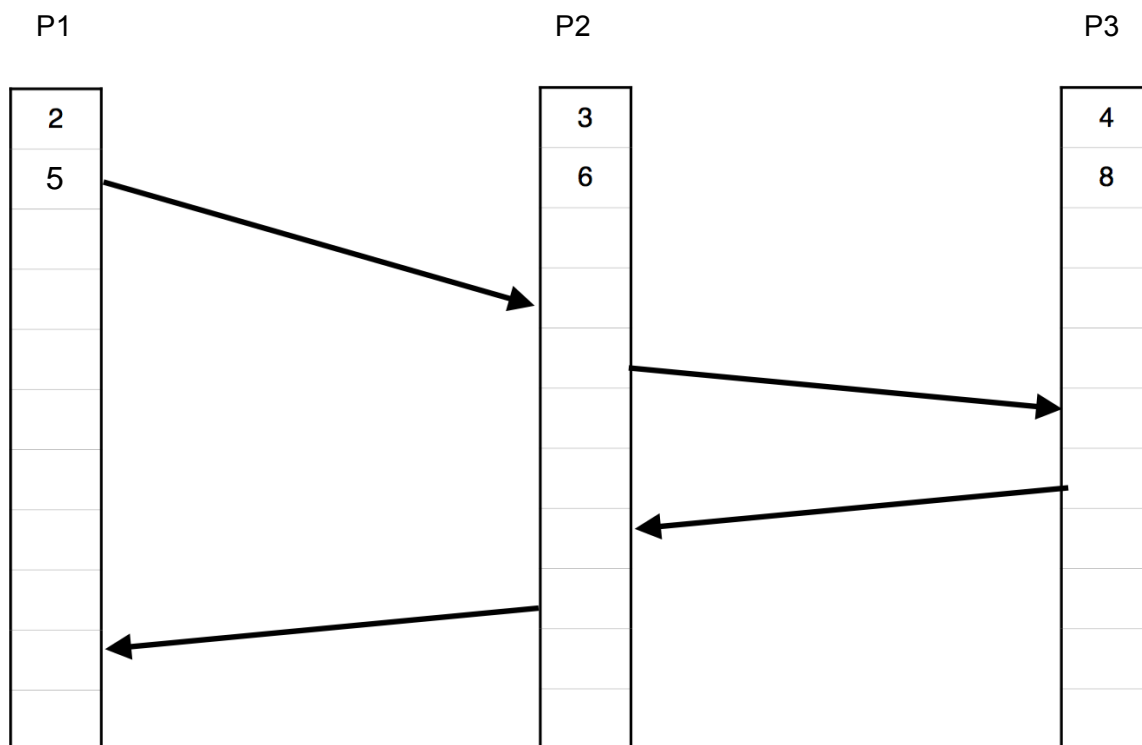
HW1: SYNCHRONIZATION

100 points

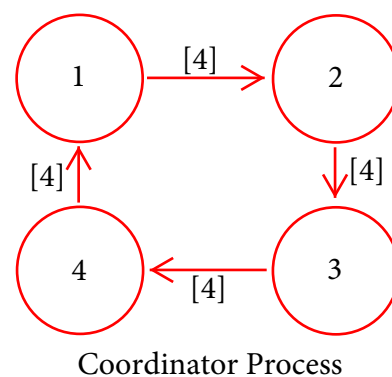
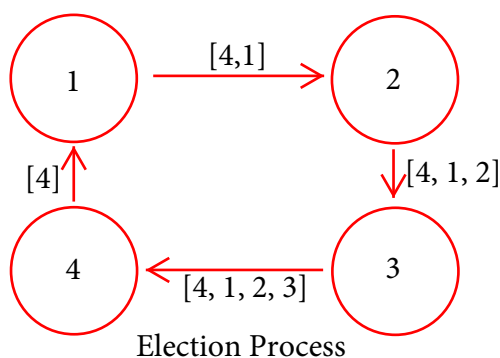
Out: 5/12/15

Due: 5/22/15 at 11:55 pm

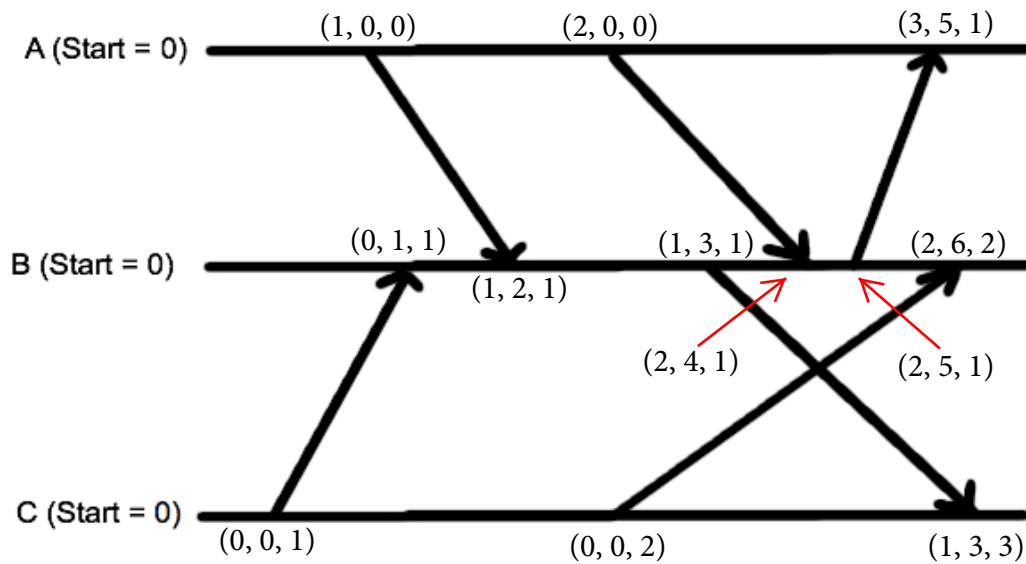
Problem 1 - Logical Clocks (20 points): Fill the time slots in columns for each process using Lamport's logical clock update mechanism (see section 6.2.1 of the textbook). Process P1 local clock increment is equal to 2, process P2 clock increment is equal to 3, and process P3 clock increment is equal to 4.



Problem 2 - Election Algorithms (8 points): A system with 4 processes uses the token ring election algorithm (see section 6.5 of the textbook). Assume that process 4 has just come online and that it initiates a new election. Draw the election process, all the messages that it generates, including the messages that the new "coordinator" sends to communicate to the other processes that it is now in charge.



Problem 3 - Vector clocks (36 points): Produce vector timestamps for all events in the figure below (see section 6.2.2 of the textbook). The three processes: A, B, and C, each run a local logical clock that starts at 0. The only events in the system are sending and receiving of messages.



Problem 4 - Mutual exclusion (36 points): A system with 2 processes (A and B) uses the central server (process C) for mutual exclusion. Each node runs a logical clock, which is initially 3 on all of them. Processes A and B send a request (Req) to process C for exclusive access to resource R. Process C sends a grant request (Gnt). The process exiting exclusive access mode sends a release message (Rel) to the central server. Draw all the events (e.g. Req, Gnt, Rel) and all the logical time stamps. Use the standard logical clock update mechanism for managing message times. Use time="2*i" for process "i" to perform computation (i=1 for A, 2 for B and 3 for C). Computation includes request processing.

