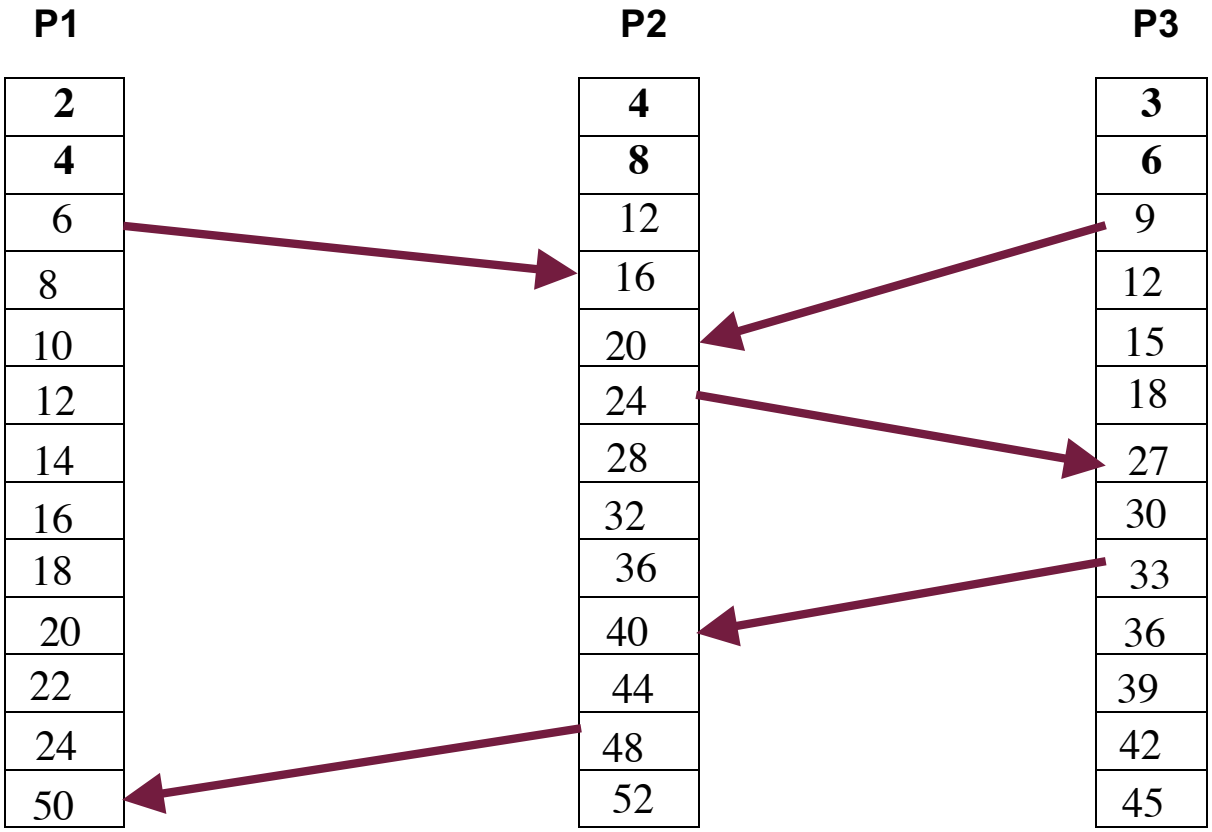
Homework 1: Synchronization

Due: 02/27/17 (Monday) at 11:55PM

Problem 1 – Logical Clocks (20 pts)

Fill in 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's local clock increment is equal to 2, process P2's clock increment is equal to 4, and process P3's clock increment is equal to 3.



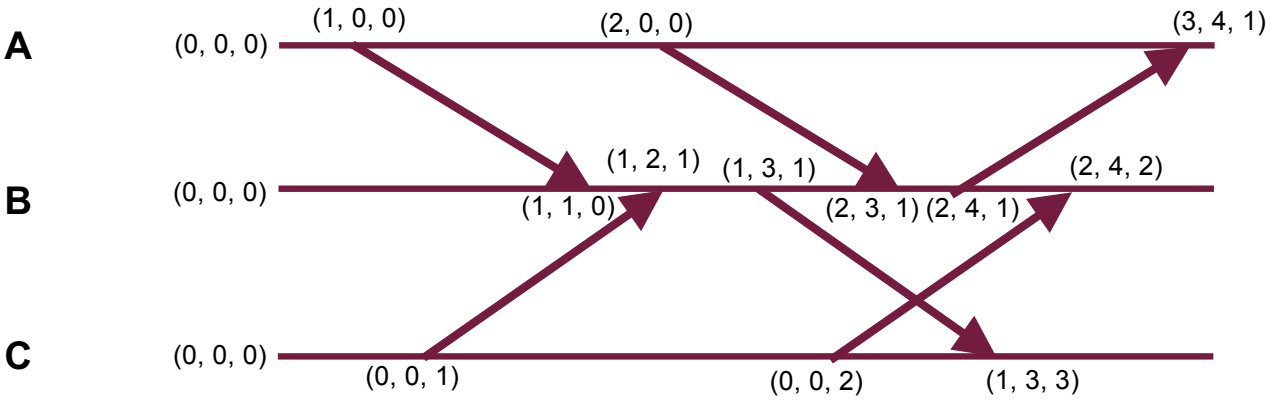
Problem 2 - Election Algorithms (8 points)

A system with 4 processes uses the 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 runs a local logical clock that starts at 0. The only events in the system are sending and receiving of messages.



Problem 4 - Mutual exclusion (5+15 points)

1. A system with two processes (A and B) uses a central server (process C) for mutual exclusion. Each node runs a logical clock, which is initially set to 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. Label all the events (e.g. Req, Gnt, Rel) in the figure below.
2. Now use a distributed algorithm (Sec.6.3.4 in the book) and logical clocks. Assume the time in the critical section by process “i” is “2*i” (i =1 for A, 2 for B and 3 for C). The logical clocks are initially 37 for A, 21 for B, and 11 for C. Use only “Req” events in the figure and show the details of mutual exclusion accesses until all are completed, including logical clocks at each event.