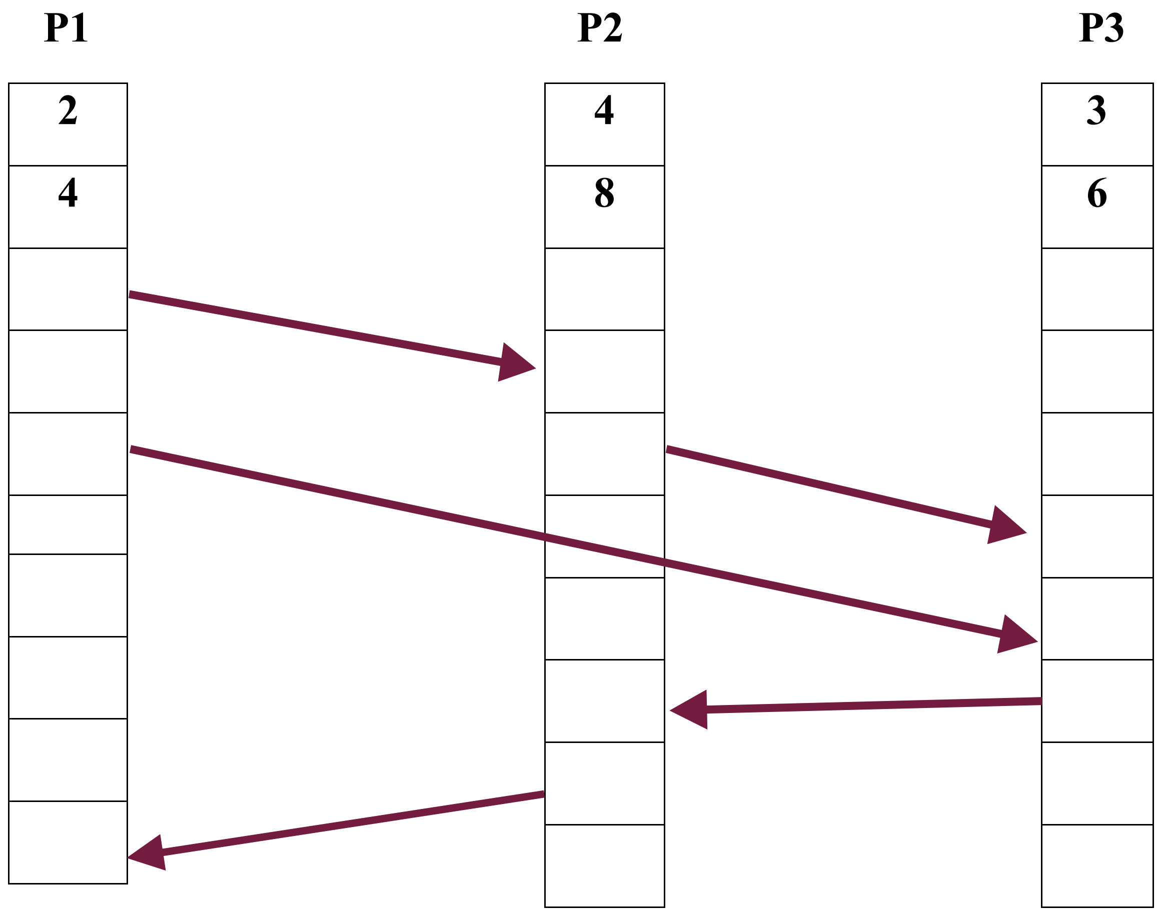
CS131 - Homework 1

Deadline: 05/27/21 (Thursday) at 11:55PM

Problem 1 – Logical Clocks – Part 1(10 pts)

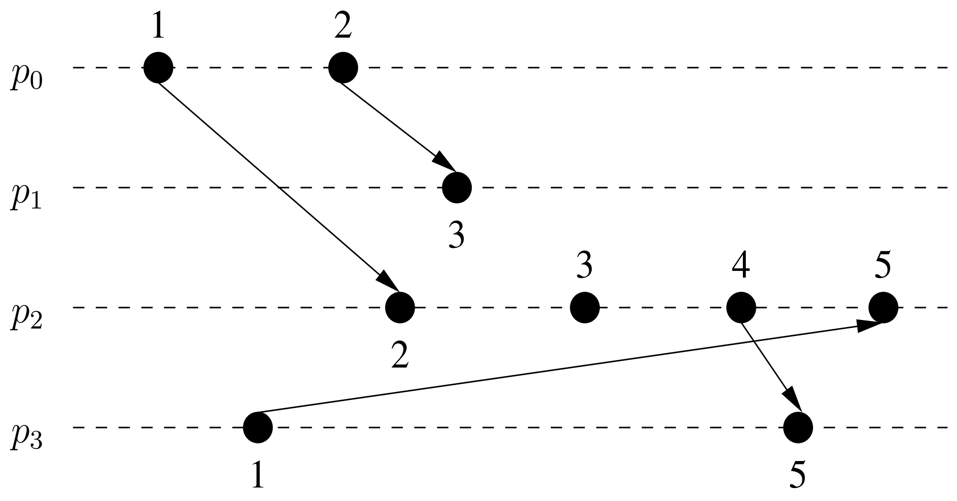
Fill in the time slots in columns for each process using Lamport's logical clock update mechanism.

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. Arrows indicate messages being sent/received.



Problem 2 – Logical Clocks – Part 2 (15 pts)

The diagram below shows logical clock values for 4 processes with a number of events. Change the event times to vector clocks in the diagram below.



Problem 3 - Clock Synchronization (25 pts)

A distributed system with N nodes uses a ring-based interconnect. The ring is unidirectional and the communication time between “adjacent” nodes is 10 msec. A node can send a “broadcast” message along the ring and every other node “receives” the broadcast as it goes by and can add it’s local time to the message. Design a clock synchronization algorithm for this system, assuming node 0 acts a time daemon of the **Berkeley algorithm**. Account for message communication time (ignoring any conflicts in the ring).

Problem 4 - Election Algorithms (25 pts)

A system has 6 processes (1-6), but process 4 is not active. Process 5 and Process 6 crash simultaneously. Process 2 notices this and initiates the election. Process 4 becomes active right after that and will receive the election message. Show all election messages and replies in this election when using the **Bully algorithm**. (Hint: processes know the ids of all other processes in the system.)

Problem 5 - Mutual Exclusion (25 pts)

A system with 4 processes (IDs 1 to 4) uses the distributed mutual exclusion algorithm (p. 324 in the textbook) with Lamport's logical clocks. The logical clocks at these processes have values 2, 4, 3 and 1 (2 at process 1, etc).

Processes 2 and 3 request access to a resource at this time. Show the sequence of messages exchanged and their logical clocks until the first process gains access to the resource.