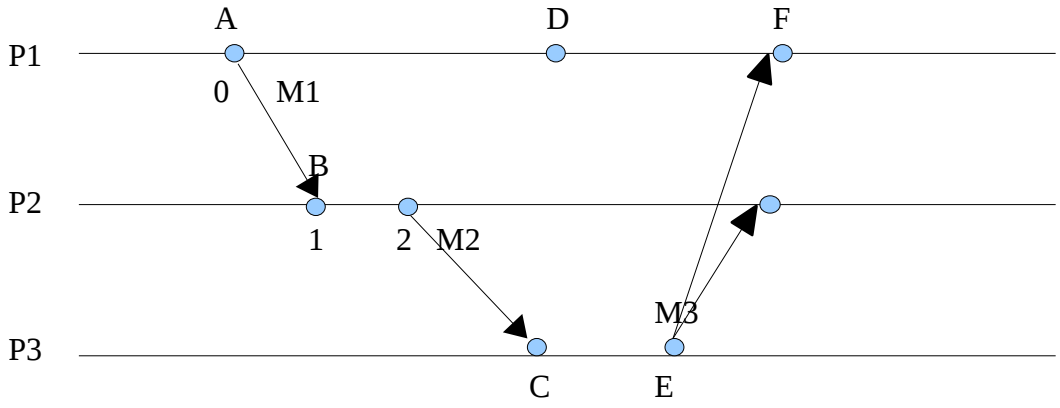
15-446 Distributed Systems

Assignment 3 – Logical Time, Replication
Due: At the midterm review (or 5:00pm 3/2, if you can't make it to the midterm review session.)

1. **Lamport "happens before"**

Process P1, P2, P3 uses Lamport's clock. Arrows represent messages sent and received by processes.

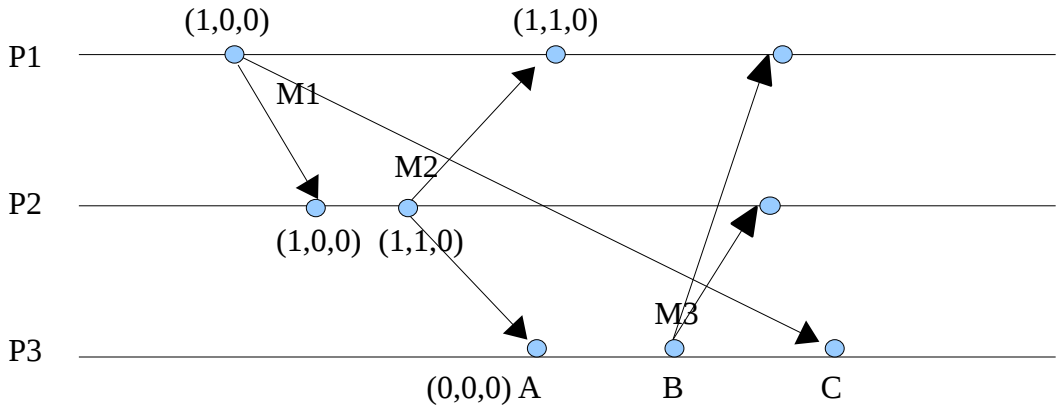


1) What's the logical time for C, E and F?

2) Mark true (T) or false (F) on the following statements. (→ denotes Lamport's "happens-before" relation.)
   a. A → E.
   b. C → D.
   c. B → E
   d. A → F

2. Causally-ordered multicast

Using vector clocks, it is possible to ensure that a message is delivered only if all messages that causally precede it have also been received. To enable such a scheme, we will assume that messages are multicast within a group of processes. Note that this causally-ordered multicasting is weaker than the totally-ordered multicasting. Specifically, if two messages are not in any way related to each other, we do not care in which order they are delivered to applications. They may even be delivered in different order at different locations.



Process P1, P2, P3 send the multicast messages M1, M2, and M3 according to the figure. P1, P2, P3 ensures the multicast messages are only delivered in causal order. The delivery of the message to the applications layer will then be delayed until the causal ordering is met.

Assume, clocks start from (0,0,0) and may only be adjusted when sending or receiving messages. Specifically, local clock is incremented when sending a message, and just before a message delivery to the application. (Assume message is delivered only one at a time.)

1) What's the causal relationship between M1 and M2?

2) At time A, P3's vector clock is (0,0,0). Does P3 delivers M2 to the application?

3) What is the vector clock timestamp that M3 carries?

4) What is the vector clock at C?

3. Replication and client centric consistency model.

Consider a nonblocking primary-backup protocol used to guarantee sequential consistency in a distributed data store. With nonblocking primary-backup, the updating process gets the acknowledgement right after the update is reflected on the primary, but before it has been reflected to any replica. The primary writes to remote replicas afterwards. Does such a data store always provide read-your-writes consistency? Explain your answer.

4. Sequential Consistency

Processes P1, P2, P3 are being executed concurrently, and they are sharing three integer variables (x,y,z) stored in a shared sequentially consistent data store. Each variable is initialized to 0.

Process P1	Process P2	Process P3
x ← 1;	y ← 1;	z ← 1;
print(y,z);	print(x,z);	print(x,y);

Is “001110” a legal output? Explain your answer.