Distributed Systems
08/01/2020

Family Name_____Name_____Student ID_____

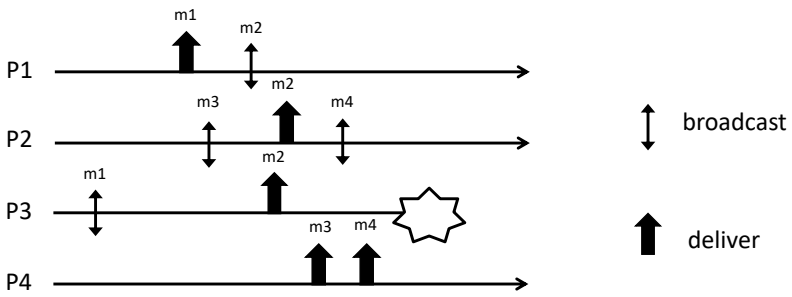
Please, tick the appropriate option:

☐ Master of Science in Engineering in Computer Science ☐ Erasmus

☐ Master of Science in Artificial Intelligence and Robotics ☐ Other

Ex 1: Let us consider a replicated object X. Discuss how crash failures are managed in the active replication and in the primary-backup approach.

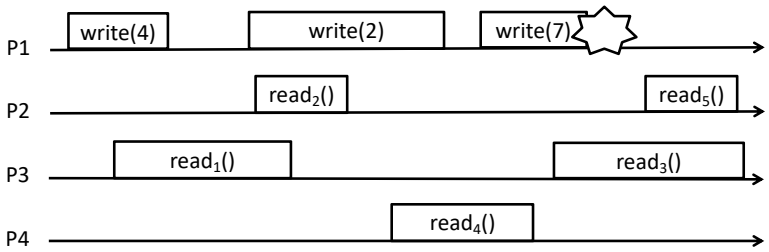
Ex 2: Consider the partial execution depicted in the Figure



Answer to the following questions:

1. Complete the execution in order to have a run satisfying Uniform Reliable Broadcast
2. Complete the execution in order to have a run satisfying Regular Reliable Broadcast but not Uniform Reliable Broadcast
3. Complete the execution in order to have a run satisfying Best Effort Broadcast but not Regular Reliable Broadcast
4. Considering the runs you provided as answer to points 1-3, discuss if they also satisfy any of the ordering properties discussed during the course.

Ex 3: Consider the partial execution depicted in the Figure



Answer to the following questions:

1. Define ALL the values that can be returned by read operations (Rx) assuming that the run refers to a regular register.
2. Define ALL the values that can be returned by read operations (Rx) assuming that the run refers to an atomic register.

3. Assign to each read operations (Rx) a return value that makes the execution linearizable.

Ex 4: Let us consider the specification of the Byzantine Consistent Broadcast

- ***Validity***: If a correct process p broadcasts a message m, then every correct process eventually delivers m.
- ***No duplication:*** Every correct process delivers at most one message.
- ***Integrity:*** If some correct process delivers a message m with sender p and process p is correct, then m was previously broadcast by p.
- ***Consistency:*** If some correct process delivers a message m and another correct process delivers a message m′, then m = m′.

And let us consider the algorithm in the following Figure

```
upon event ⟨ bcb, Init ⟩ do
    sentecho := FALSE;
    delivered := FALSE;
    echos := [⊥]^N;

upon event ⟨ bcb, Broadcast | m ⟩ do                          // only process s
    forall q ∈ Π do
        trigger ⟨ al, Send | q, [SEND, m] ⟩;

upon event ⟨ al, Deliver | p, [SEND, m] ⟩ such that p = s and sentecho = FALSE do
    sentecho := TRUE;
    forall q ∈ Π do
        trigger ⟨ al, Send | q, [ECHO, m] ⟩;

upon event ⟨ al, Deliver | p, [ECHO, m] ⟩ do
    if echos[p] = ⊥ then
        echos[p] := m;

upon exists m ≠ ⊥ such that #({p ∈ Π | echos[p] = m}) > (N+f)/2
        and delivered = FALSE do
    delivered := TRUE;
    trigger ⟨ bcb, Deliver | s, m ⟩;
```

Discuss what happen to each property in the following cases:

1. $2f < N \leq 3f$ and links are authenticated.
2. $N > 3f$ and links are perfect.

Ex 5: Consider a distributed system constituted by *n* processes $\prod=\{p_1, p_2 \ldots p_n\}$ with unique identifiers that exchange messages through FIFO perfect point-to-point links and are structured through a line (i.e., each process p_i can exchange messages only with processes p_{i-1} and p_{i+1} when they exists). Processes may crash and each process is equipped with a perfect oracle (having the interface *new_right(p)* and *new_left(p)*) reporting a new neighbor when the previous one is failing. Write the pseudo-code of an algorithm implementing a Total Order Broadcast communication primitive.

According to the Italian law 675 of the 31/12/96, I authorize the instructor of the course to publish on the web site of the course results of the exams.

Signature: _____