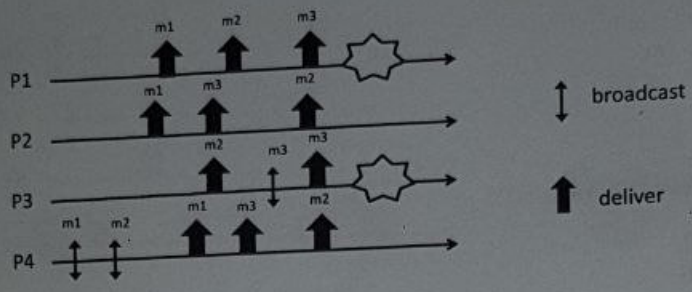
Distributed Systems 08/02/2019

Corso di Laurea Magistrale in Ingegneria Informatica

Family Name_____ Name_____ Student ID_____

Ex 1: Consider the execution depicted in the Figure

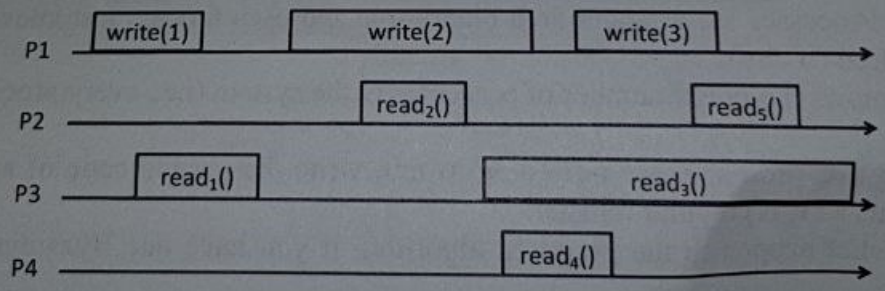


Answer to the following questions:

1. Which is the strongest TO specification satisfied by the proposed run? Motivate your answer.
2. Does the proposed execution satisfy Causal order Broadcast, FIFO Order Broadcast or none of them?
3. Modify the execution in order to satisfy TO(UA, WUTO) but not TO(UA, SUTO).
4. Modify the execution in order to satisfy TO(NUA, WUTO) but not TO(NUA, SUTO).

***NOTE**: In order to solve point 3 and point 4 you can only add messages and/or failures.*

***Ex 2:** Consider the execution depicted in the following figure and answer the questions*



1. Define ALL the values that can be returned by read operations (Rx) assuming the run refers to a regular register.
2. Define ALL the values that can be returned by read operations (Rx) assuming the run refers to an atomic register.
3. Let us assume that values retuned by read operations are as follow: read$_1$() → 0, read$_2$() → 2, read$_3$() → 3, read$_4$() → 3, read$_5$() → 2. Is the run depicted in the Figure linearizable?

Ex 3: Consider the algorithm shown in the Figure

```
upon event ⟨ Init ⟩ do
    correct := Π;
    from[p] := [∅]^N;

upon event ⟨ rb, Broadcast | m ⟩ do
    trigger ⟨ beb, Broadcast | [DATA, self, m] ⟩;

upon event ⟨ beb, Deliver | p, [DATA, s, m] ⟩ do
    if m ∉ from[s] then
        trigger ⟨ rb, Deliver | s, m ⟩;
        from[s] := from[s] ∪ {m};

upon event ⟨◊P,Suspect |p⟩do
    correct := correct \ {p};
    forall m ∈ from[p] do
        trigger ⟨ beb, Broadcast | [DATA, p, m] ⟩;

upon event ⟨◊P,Restore |p⟩do
    correct := correct ∪ {p};
```

Assuming that the algorithm is using a Best Effort Broadcast primitive and an Eventually Perfect Failure Detector ◊P discuss if the following properties are satisfied or not and motivate your answer

- *Validity*: If a correct process p broadcasts a message m, then p eventually delivers m.
- *No duplication*: No message is delivered more than once.
- *No creation*: If a process delivers a message m with sender s, then m was previously broadcast by process s.
- *Agreement*: If a message m is delivered by some correct process, then m is eventually delivered by every correct process.

Ex 4: *Consider a distributed system composed of N processes* p_1, p_2, … p_N, *each having a unique identifier myID. Processes are arranged in a binary tree and each process just knows*[1] *its father (if any) and its children (if any).*
Each process p_i *knows the initial number of processes in the system (i.e., every process* p_i *knows the value of N).*

1. Assuming that processes are not going to fail, write the pseudo-code of an algorithm that implements a (1, N) regular register.
2. Discuss what happen to the proposed algorithm if you have one Byzantine process in the system.

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: _____

[1] Assume that father and children are stored respectively in two local variables FATHER and CHILDREN.