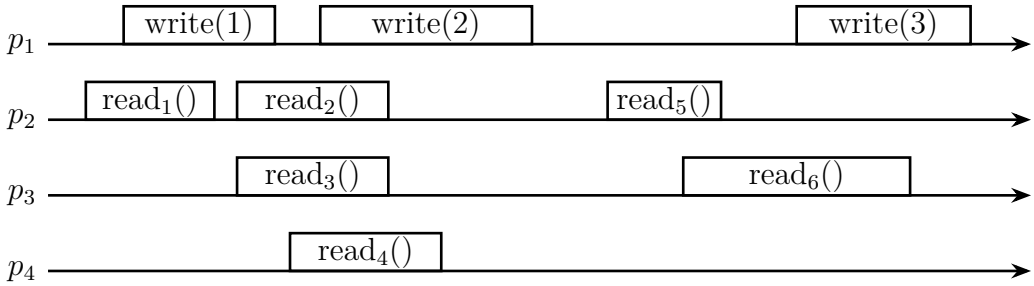
DDS Exam - 07.06.2023

Exercise 1

Discuss blockchain from distributed system point of view, the distinction between public, private, permissioned and premissionless. In addition, explain how PoW (Proof-of-Work) mechanism works and why it creates branches in the blockchain data structure.

Exercise 2

Consider the execution depicted in the following figure.

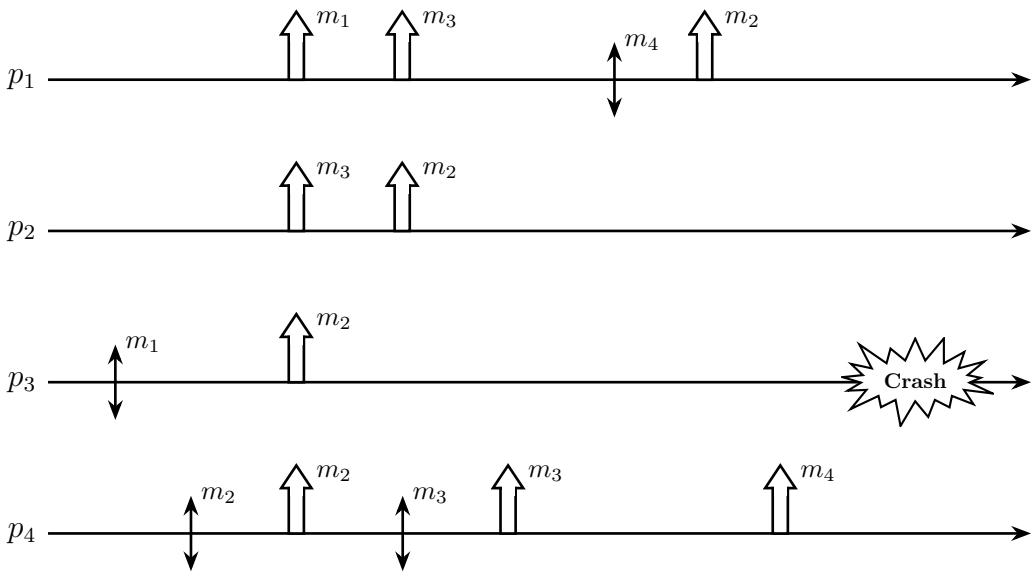


Answer some true/false questions I don't remember, but basically answer the questions:

1. Define ALL values that can be returned by read operations (Rx) assuming the run refers to a regular register
2. Define ALL values that can be returned by read operations (Rx) assuming the run refers to an atomic register
3. Provide a sequence such that the execution is linearizable

Exercise 3

Consider the message pattern shown in the Figure below.



Answer some true/false questions I don't remember, but basically questions on broadcast communications and ordered communications.

Exercise 4

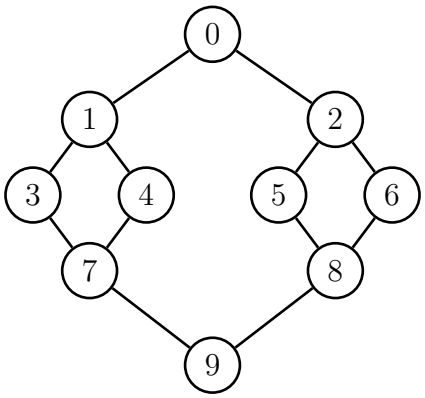
Algorithm: Beb

```
1  upon event Init
2      ID := unique integer identifier;
3      Π := IDs of the neighbors;

4  upon event ⟨ beb, Broadcast | m ⟩
5      trigger ⟨ beb, Deliver | ID, m ⟩;
6      forall k ∈ Π do
7          trigger ⟨ pl, Send | k, ⟨ID, m⟩ ⟩;

8  upon event ⟨ pl, Deliver | j, ⟨i, m⟩ ⟩
9      trigger ⟨ beb, Deliver | i, m ⟩;
10     forall k ∈ Π do
11         if (j < ID ∧ k < i) ∨ (j > ID ∧ k > i) then
12             trigger ⟨ pl, Send | k, ⟨i, m⟩ ⟩

13         if k == 9 then
14             trigger ⟨beb, Deliver | i, m ⟩;
```



The distributed algorithm is executed at every process. In particular, p_0 is the process that starts the Beb broadcast.

Assume that links are perfect and load independent (i.e., $p_i \to p_j$ and $p_j \to p_i$ are independent and do not impact performance of other links). Answer the questions:

1. Assume that p_0 sends a message every 2 seconds, and every process is able to process 1 message per second, and assume that the distribution of arrivals and distribution of services are exponentially distributed, compute the average time to complete the broadcast (i.e., every process has delivered the message).
2. Don't remember :(
3. Don't remember :(

Exercise 5

Consider a system composed by two disjoint set of processes for clients (c_1, c_2, ..., c_n) and for servers (s_1, s_2, ..., s_m). There is a set of resources $R = \{R_1, R_2, R_3\}$ that clients want to request. Clients communicate with servers using perfect point-to-point links. The servers receive the requests from clients and they need to do that, at every time t, the same resource is allocated to one client, and requests that can't be satisfied need to be stored for being satisfied later.

Clients can request multiple resources at the same time, but every resource can be allocated by just one client. Assume that

1. Clients and servers can crash
2. Clients and servers have access to a perfect failure detector
3. Servers can communicate between them using uniform reliable broadcast
4. Clients and servers communicate through perfect point-to-point links

Answer the questions:

1. Provide the implementation of the resource allocation algorithm (at least Client's code).
2. Dont' remember
3. Dont' remember (something on byzantine failure model)

1) A blockchain is a decentralized and distributed ledger used to record transactions across many computers. Every block in a blockchain contains a set of transaction and they are collected by processes. The transactions need to be validated respect to the ledger specification. When a certain number of transactions have been collected a block can be created and attached to the chain.
A blockchain can be classified in:

PUBLIC: open to anyone and anyone can join the network partecipate in consensus process and validate transactions.

PRIVATE: the access is restricted to a specific group; the entities require permission to join and there is more control over the network.

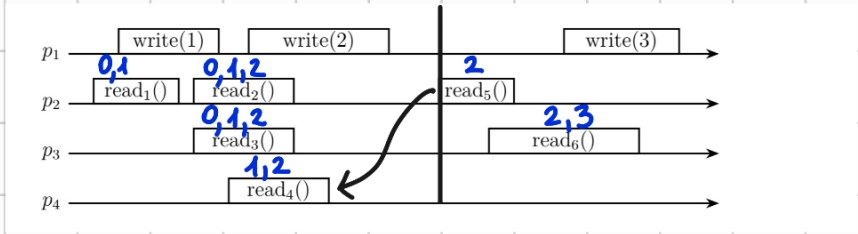
PERMISSIONED: partecipants need permission to join and validate transactions

PERMISSIONLESS: anyone can join and validate transactions without distinction about identities.

The PoW is a consensus mechanism used in blockchain. The idea is to use miners to perform a computationally intensive task to add new blocks to the chain. The miners that "solve" a block can propose it as the next in the chain.
The mining is: find a number s.t. hash(block) < target. The validity is checked by block's receiver that re-compute the hash.

The branches are created when multiple miners solve the blocks and then is a temporary divergence. This leads to the creation of branches, when there are multiple proposes as the next. In order to solve it, the network follows the longest chain rule and have to converge to the longest branch (the valid one). This ensures security and integrity.

2)



Regular: R_1 : 0,1 R_5 : 2

R_2 : 0,1,2 R_6 : 2,3

R_3 : 0,1,2

R_4 : 1,2

Atomic: R_1 : 0,1 R_5 : 2

R_2 : 1,2 or 0,1,2 R_6 : 2,3

R_3 : 1,2 or 0,1,2

R_4 : 1,2

if $R_1 = 0$, then $R_2 = (0,1,2)$

if $R_1 = 1$, then $R_2 = (1,2)$

if $R_1 = 0$, then $R_2 = (0,1,2)$

if $R_1 = 1$, then $R_2 = (1,2)$