

Instructions

For the main question, be ready to:

- Describe the architecture of your solution and its operations.
- Define all its components and their interactions.
- Motivate your design decisions.
- Make diagrams whenever necessary.
- **State clearly your choices and assumptions.**

Question 1.

You are tasked with the design of a key-value store that is accessible to a set of clients. Every process is able to read and write to the shared key-value storage concurrently. Make sure the transactions are linearizable. The storage should be persistent against at least a single failure until the termination of the algorithm. Assume a fail-stop setting.

- What is linearizability?
- Do the termination and validity properties hold?
- Are the transactions atomic?
- Can we improve performance (read throughput) if we drop the linearizability property, but instead only desire sequential consistency? What is sequential consistency?

Tip: don't overthink or overengineer your solution, be lazy-efficient.

Question 2. (theory)

- Can we reach consensus in asynchronous systems?
- What is the main intuition behind the FLP result?
- The FLP result exists, so why does Paxos work?
- Can you come up with a (simple) scenario where Paxos does not terminate?