Chalmers Techniska Högskola
Data- och Informationsteknik

CTH TDA297
GU DIT290

Distribuerade system fk

Tentamen 2023-08-18

Dag, *Tid, Sal*: August 18th 2023, 14:00-18:00, SBMultisal

Kursansvarig: Philippas Tsigas (Tel: 772 5409)

Totalt *Poängtal*: 60

Betygsgränser:

CTH: 3:a 30 p, 4:a 38 p, 5:a 48 p

GU: Godkänd 30p, Väl godkänd 48 p

Instructions

- Please answer in English, if possible.
  If you have very big difficulty with that, though, you may answer in Swedish.
- **Do not forget to write your personal number and if you are a GU or CTH student and at which "linje".**
- Please start answering each assignment on a new page; number the pages and use only one side of each sheet of paper.
- Please write in a tidy manner and explain (Clearly) your answers.

LYCKA TILL !!!!

1. (10 points) Consider the dinning philosophers problem, n philosophers $P_1, \ldots, P_n$ in a bidirectional ring. Moa wants to help them eat and not starve. She assigns n priorities to them in the following way: P_1 has the lowest priority, P_2, has the second lowest, and so on up to P_n who has the highest priority. Moa instructs each philosopher to seek forks in parallel and to use their priorities to resolve conflicts, i.e. the philosopher with the highest priority gets the respective fork when hungry and competing with a neighbor. Moa is not sure whether her protocol is correct and cannot find a way to calculate its complexity.

    1. Can you help Moa to prove or disprove the correctness of her protocol by providing a proof or a counterexample?
    2. If you prove that the protocol is correct please provide a complexity analysis of the protocol.
    3. If you prove that the protocol is not correct, can you extend the protocol, by wrapping it around with another protocol leyer, without modifying Moa's part (you do not want to hurt Moa's feelings :)), to make it correct? What is the complexity of the algorithm?
    4. Can you design another algorithm from scratch that solves the dinning philosophers problem that has better time complexity?

2. (10 points) Moa wants to implement a replicated shared counter object that support two operations `inc` and `read`, where `read` returns the number of previous `inc` operations using a quorum based approach. She expects that her system is going to have significantly more `read` operations compared to `inc` operations; because of that she wants to make sure that the replication system provides as high availability as possible to the `read` operations. Can you design an algorithm for that? What is the consistency of your algorithm (linearizability, sequential consistency, eventual consistency); provide a proof?

3. (15 points) Consider a synchronous network, in every round every process may send and receive a message to/from each of its neighbors. The network is organized as a 2-dimensional grid. The width of the grid is w, the height is h. Width and height are defined in terms of edges. All processes start at the same time. At the beginning we assume no process or link failures.

    A Assume that all processes know the grid dimensions. Design a protocol that reaches consensus.

    B Can you modify the protocol to tolerate process crashes, same assumptions, no partitions.

    C Assume that none of the processes knows the grid dimensions, no process failures. Design a protocol that reaches consensus. How many rounds does your algorithm require?

    D Assume a system with crash failures; i.e, nodes can crash at any time during the execution, no partitions. Does your algorithm from C work in these system settings? If your answer is yes calculate the number of rounds it takes. If your answer is no provide an execution that does not.

4. (10 points) A processor P is part of an asynchronous connected network $G(V, E)$. P believes that processor Q is also connected to the network. Describe a protocol that P together with the other processors of the network can use in order to find Q or to find that Q is not part of the network. Prove the time complexity of the algorithm.

5. (5 points) Construct a solution to reliable, totally ordered multicast in a synchronous system, using a reliable multicast and a solution to the consensus problem.

6. (10 points) Each statement is either true or false. A correct answer gives 1 point, a wrong answer gives -1 point, no answer gives 0 points. Overall you cannot get less than 0 points for this question.

   1. In an asynchronous system, where only one processor might crash, consensus is solvable.
      A. True B. False
   2. Leader election can be solved with a consensus algorithm.
      A. True B. False
   3. The Gossip architecture for replication was designed to provide highly available service.
      A. True B. False
   4. The 3 Phase Commit protocol was introduced to improve the latency of the 2 Phase Commit protocol in executions where no faults take place.
      A. True B. False
   5. In an asynchronous system, where messages can be lost undetectably, consensus is solvable.
      A. True B. False
   6. In an synchronous system, where messages can be lost undetectably, consensus is solvable.
      A. True B. False
   7. Bitcoin is linearizable.
      A. True B. False
   8. Bitcoin is strongly eventual consistent.
      A. True B. False
   9. A system can be sequential consistent but not linerizable.
      A. True B. False
   10. Any transactional service is linearizable if each request is executed at most once.
       A. True B. False