KTH Computer Science and Communication

Hand-in 3
DD2451 + FDD3008, Parallel and Distributed Computing Fall 2011

Posted Dec 2 16.00. Due Dec 9 16.00. Answers can be mailed to mfd@kth.se as pdf or dropped in Mads' intray, located on level 4, Lindstedtsvägen 3 (enter through the doors with the label "NADA", after going a small distance to the left proceed straight and you find the department mail boxes on your right). Make sure answers are clearly marked with name and mail account on each sheet. The general rules on homework solutions available at the course home-page apply. In particular, discussions of ideas in groups of up to at most two people are allowed but solutions should be written down individually, and you should note the name of your discussion partner. Some of the problems below are "classical" and hence their solutions are probably posted on the Internet. It is *not* allowed to use such solutions *in any way*. The order of the problems is "random" and hence do not expect that the lowest numbered problems are the easiest. Any corrections or clarifications on this problem set will be posted under "Exercises and hand-ins" on the course home page (google DD2451).

1. (14p) Consider the randomized algorithm in the slides for lecture 7. Suppose only crash failures happen.
   - 1a (7p) How many crash failures can the algorithm handle? Explain.
   - 1b (7p) Modify the algorithm to handle as many crash failures as you can. Explain how many crash failures the modified algorithm can handle.

2. (15p) Devise an algorithm for leader election in asynchronous connected network graphs. You may assume that nodes have linearly ordered node identifiers and that nodes do not fail. Leader election may at any time be initiated by one or more nodes independently of each other. What is the message complexity of your algorithm?

3. (15p) The following synchronous algorithm assumes unique node identifiers.

```
algorithm foo:
variable b: boolean ;
Each round every node with identifier id executes:
  do atomically
    send (id,b) to all neighbours ;
    receive a message from each neighbour ;
    if a received message is of form (id',true) and id < id'
    then b := false
    else b := true
  end do
end round ;
```

   Describe which graph problem algorithm `foo` solves. *(Hint: After stabilization the graph with nodes labelled by flags b has a certain property. Which?)* Prove that `foo` is correct, i.e. that after stabilization the property you have identified indeed does hold. Prove that the algorithm is self-stabilizing (that is, when started in any initial state, the algorithm converges to a stable state in a finite amount of rounds). How many rounds are needed for stabilization in the worst case? Explain your reasoning.

4. (12p) Consider the 3PC protocol. Explain how the affected parties act in the following situations:
   - 3a (2p) Coordinator and 1 cohort fail in phase 1
   - 3b (2p) Coordinator and 1 cohort fail in phase 2
   - 3c (2p) Coordinator and 1 cohort fail in phase 3
   - 3d (2p) The coordinator times out in phase 1
   - 3e (2p) The coordinator times out in phase 2
   - 3f (2p) The coordinator times out in phase 3

5. (14p) Suppose Paxos is changed in the following way: The conditional

```
if m = 0 then
  Send propose(x,n) to the same set of acceptors
else
  Send propose(y,n) to the same set of acceptors
```

   in lines 6-11 of the pseudocode for Proposer in slide set 9 is replaced by the statement

```
Send propose(x,n) to the same set of acceptors
```

   Does the agreement property still hold? Explain carefully why or why not.

6. (15p) A risk in Byzantine fault recovery procedures is that one or several Byzantine replicas conspire to eliminate a non-faulty primary. Explain how this is prevented in PBFT and in the Zyzzyva system.

7. (15p) In this exercise you may start from the binary tree DHT presented in the slides for lecture 10. Describe three attacks a Byzantine adversary might mount against a DHT and how you would protect against them, by suitable modifying the algorithms for search, join, and leave. A concise textual description is sufficient, you do not need to provide pseudocode, for instance.

Good luck!