Worksheet 4

Co-ordination and Transaction Processing

Group discussion questions

1. Give an example execution of the ring-based mutual exclusion algorithm to show that processes are not necessarily granted entry to the critical section in causality order.

2. A reliable failure detector can always accurately detect a process' failure. It gives one of two answers: *Unsuspected* (i.e. has recently received evidence that the process has not failed, but it may have failed since) or *Failed* (i.e. it has determined that the process has failed). An unreliable detector can only return *Suspected* or *Unsuspected*, which are only hints about the process healthiness.

(i) Suggest how to implement an unreliable failure detector.

(ii) Is it possible to implement either a reliable or an unreliable (process) failure detector using an unreliable communication channel? *Hint*: consider different distributed system models.

3. Explain how the two-phase commit protocol for nested transactions ensures that, if the top-level transaction commits, the correct descendants are committed or aborted.

Problem solving questions

4. Consider a ring with 5 processes, each with unique id, arranged in the following order: 5, 9, 2, 4, 3. Assume that links are reliable and there are no process crashes.

(i) The process with id 2 starts a new election. Describe the steps that the ring-based election algorithm of Chang & Roberts takes to elect the leader.

(ii) What happens if two processes, say 5 and 2, start election simultaneously?

(iii) Assume that links are reliable but processes may crash. To ensure that the ring topology is not affected, when a crash occurs the process is still able to receive and forward messages unchanged but can no longer perform processing of the ids. What problems arise for the Chang & Roberts algorithm?

(iv) Assume the links in the system are reliable and there are no process crashes. The election algorithm is run to elect a co-ordinator that will issue a token if one gets lost. At most one token can be circulating at any time. A process may start an election whenever it thinks that a token is lost, but this cannot be established with certainty, so it is possible for the old token to be still circulating. How would you modify the behaviour of the Chang & Roberts algorithm to deal with this situation?

5. Adapt the central server algorithm for mutual exclusion to handle the crash failure of any client (in any state), assuming that the server is correct and you are given a reliable failure detector. Discuss what would happen if a token-holding client:

(i) crashes

(ii) is wrongly suspected to have failed?

6. Consider the synchronous consensus algorithm based on multicast between three processes, P, Q and R, who attempt to agree on integer values from the set {1..5}. The initial values are 1 for P, 3 for Q and 4 for R. R crashes after sending the first message. Give the steps taken by the algorithm to reach consensus, assuming tolerance of one crash. Would you recommend using the mean as the decision value?

7. A three-phase commit protocol has the following parts:

Phase 1: is the same as for two-phase commit.

Phase 2: the co-ordinator collects the votes and makes a decision; if it is *No*, it *aborts* and informs participants that voted *Yes*; if the decision is *Yes*, it sends a *preCommit* request to all the participants. Participants that voted *Yes* wait for a *preCommit* or *doAbort* request. They acknowledge *preCommit* requests and carry out *doAbort* requests.

Phase 3: the coordinator collects the acknowledgments. When all are received, it *Commits* and sends *doCommit* to the participants. Participants wait for a *doCommit* request. When it arrives they *Commit*.

Explain how this protocol avoids delay to participants during their ‘uncertain’ period due to the failure of the co-ordinator or other participants. Assume that communication does not fail.