Exam Distributed Algorithms

Free University Amsterdam, 23 March 2011, 15:15-18:00

(Answers can be given in English or Dutch. You may use the textbook Introduction to Distributed Algorithms *by Gerard Tel. Use of slides, handouts, laptop is* not *allowed.)*

(The exercises in this exam sum up to 90 points; each student gets 10 points bonus.)

1. Consider an undirected network of N processes $p_0, \ldots, p_{N-1}$, where $p_1, \ldots, p_{N-1}$ form a ring and p_0 has a channel to every other process. Give an execution of the echo algorithm on this network that takes N time units to complete (assuming that a message takes at most one time unit). (12 pts)

2. Run the FireWire leader election protocol on an acyclic undirected network of three processes. Give a scenario in which a leader is elected after an occurrence of root contention. (12 pts)

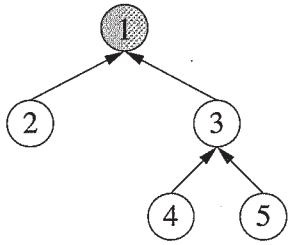
3. Consider the Itai-Rodeh ring size algorithm. Argue (in your own words) why upon message-termination, est_p is the same at all processes p. (10 pts)

4. The requirement of strong accuracy is stronger than the requirement that processes that never crash are never suspected:

$$\forall \tau \; \forall p \notin Crash(F) \; \forall q \notin F(\tau) : \quad p \notin H(q, \tau)$$

Give an example of a failure pattern and a failure detector history that satisfy this property, but which are not allowed in case of a strongly accurate failure detector. (10 pts)

5. Run Raymond's mutual exclusion algorithm on the graph below.



Initially node 1 holds the token. Give a scenario in which first node 5, then node 4, and then node 2 requests the token, but they receive the token in the opposite order. (10 pts)

6. In the t-Byzantine robust synchronizer of Lamport and Melliar-Smith, a correct process p accepts a local clock value of another process q if it differs no more than δ from its own clock value, at the moment of synchronization. Explain in detail why that synchronizer has precision $\frac{3t}{N}\delta$ (versus precision $\frac{2t}{N}\delta$ of the Mahaney-Schneider synchronizer). (14 pts)

7. Given three processes p_0, p_1 and p_2 that are all connected to each other. Let $leader_0 = leader_1 = leader_2 = 3$; $father_0 = 1$, $father_1 = 2$ and $father_2 = 0$; $dist_0 = 1$, $dist_1 = 0$ and $dist_2 = 2$.

   Describe a scenario of the Afek-Kutten-Yung self-stabilizing leader election algorithm, in which eventually p_2 is elected as leader. (12 pts)

8. Consider a processor with one periodic task $(1, 3, 1)$ (starts at time 1, period 3, execution time 1). Suppose sporadic jobs S_1, S_2, S_3 and S_4 arrive at times 0, 1, 3 and 6, with execution times 1, 3, 1 and 2, and with deadlines 1, 12, 7 and 14, respectively. Which of them pass the acceptance test for sporadic jobs? (Explain your answer.) (10 pts)

FireWire

- When a process has one possible parent, it sends a parent request to this neighbor. If the request is accepted, an ack is sent back.
- The last two parentless processes can send parent requests to each other simultaneously. This is called root contention.
- Each of the two processes in root contention randomly decides to either immediately send a parent request again, or to wait some time for a parent request from the other process.

Raymond's algorithm

Each process maintains a FIFO queue, which can contain id's of its children, and its own id. Initially, this queue is empty.

- When a non-root wants to enter the critical section, it adds its id to its own queue.
- When a non-root gets a new head at its (non-empty) queue, it asks its parent for the token.
- When a process receives a request for the token from a child, it adds this child to its queue.

When the root exits the critical section (and its queue is non-empty), it sends the token to the process q at the head of its queue, makes q its parent, and removes q from the head of its queue.

Let a process p get the token from its parent, with process q at the head of its queue.

- if $p \neq q$, then p sends the token to q, and makes q its parent
- if $p = q$, then p becomes the root (i.e., it has no parent, and is privileged)

In both cases, p removes q from the head of its queue.

Acceptance test for sporadic jobs

A sporadic job with deadline d and execution time e is accepted at time t if utilization (of the periodic and accepted sporadic jobs) in the time interval $[t, d]$ is never more than $1-\frac{e}{d-t}$.

If accepted, utilization in $[t, d]$ is increased with $\frac{e}{d-t}$.