[CSC3101]

NEWCASTLE UNIVERSITY

SEMESTER 1 2010/11

DISTRIBUTED SYSTEMS

Time allowed – 1½ Hours

Instructions to candidates:

Answer any TWO questions

Marks shown for subsections are indicative only

[Turn over

Question 1.

a) Describe the "at least once" semantics of remote procedure calls (RPCs). Identify an application context where such RPCs would be useful. [3 marks]

b) Assuming that any crashed node eventually recovers, describe a scheme that guarantees "at most once" RPC semantics when no communication failures can occur and only the server node is prone to crashing. [7 marks]

c) Using the principles of the two-phase commit protocol, enhance the above scheme to provide "exactly once" RPC semantics when no communication failures can occur and both the server node and the client node are prone to crashing [12 marks]

d) Define the possible RPC semantics that can be obtained in the presence of server crashes and lost messages. How can the at most once semantics be obtained using call sequence numbers? [3 marks]

Question 2.

a) Consider the following form of anomalous behaviour. Process p sends a message m to process q after stamping m with its clock reading at the time of sending. Let the timestamp of m be denoted as T_m. Process q receives m when it clock reads less than T_m.

   i) Argue that this anomalous behaviour cannot occur if clocks of p and q had been logical. [5 marks]

   ii) Given that the p and q refer to physical clocks that are synchronised within some non-zero ε and the minimum transmission time for m can be μ, construct a scenario to illustrate the occurrence of this anomalous behaviour when $\mu < \varepsilon$. [5 marks]

b) It is desired that a total order message delivery service is to be built for a system of distributed processes. The following assumptions should be made:

   A1) Processes have access to perfectly synchronised physical clocks; i.e., the synchronisation error is zero;

   A2) The underlying communication subsystem guarantees that a sent message is eventually delivered to the destination, but guarantees no bound on transmission delays;

A3) For any two processes p and q, messages sent by p to q are received in the sent order;

A4) Sending of a given message to one or more destinations is a single event within the sending process and a message is stamped with the sending time according to the sending process' clock; and,

A5) Processes never crash and their identifiers are uniquely ranked; this ranking is known to all processes.

The desired message delivery service should satisfy the following three conditions in delivering the received messages to processes:

C1) Suppose that messages m and m' are sent to a process, say p. If sending of m *happens before* sending of m', then m is delivered to p before m' is delivered.

C2) When multiple messages (say, m and m') are sent to a common set of destination processes, these messages are delivered in the same order to all common destination processes.

C3) Any message sent to a process is eventually delivered to that process.

i) Describe a design for such a service; your description should explain how the conditions C1, C2 and C3 are met. [8 marks]

ii) Suppose that assumption A2 is revised as follows. The communication subsystem guarantees a known upper bound, denoted as D, on message transmission delays. That is, if a process sends m at time t_1 as per its local clock, m will be received at some time t_2 as per the destination process' local clock and $t_2 \leq t_1 + D$. Explain how your design for i) can be modified to be more message efficient while at the same time meeting all three conditions C1-C3? [7 marks]

 [Turn over

Question 3.

a) Let p and q be two processes in a distributed system where a sent message is eventually received by the destination process(es). The following predicates are true at some point in time:

P1 a message sent by p is in transit;

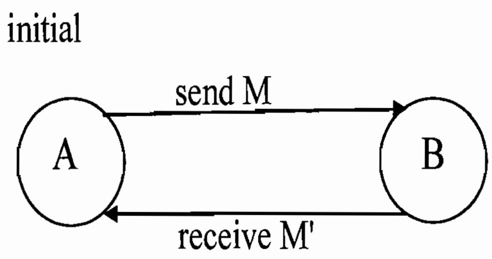
P2 a message sent to q is received by q; and,

P3 p's logical clock has a value smaller than the value of q's logical clock.

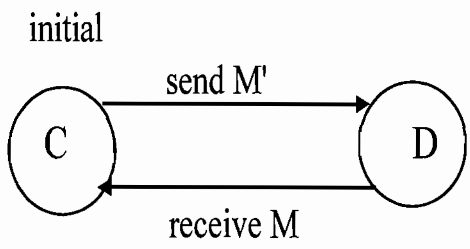
Assess which of these predicates can be stable properties. [6 marks]

b) Describe the marker sending and receiving rules to be followed by a process so that the global state of a distributed system can be constructed from the recordings of individual processes in the system. [6 marks]

c) Consider a system of two distributed processes p and q. Process p is initially in state A where it can only send message M to q and this sending makes p go to state B. Process q is initially in state C where it can only send message M' to p and this sending takes q to state D. Processes p and q can only receive a message when they are in states B and D respectively, and message reception changes their states to A and C respectively. The state transition diagrams are given below:



State Transition Diagram for p.



State Transition Diagram for q.

Assuming that the (unidirectional) channels connecting the processes are initially empty and that the recording terminates, determine the recorded global state if the recording activities interleave with the computational events e0, e1, …, e6, in the following manner:

e0: q sends M'; e1: p sends M; e2: q receives M;

// q records its state and acts

e3: q sends M'; e4: p receives M'; e5: p sends M;

// p receives q's marker and acts

e6: q receives M;

// q receives p's marker and acts.

If the recorded global state did not actually occur during the computation {e0, e1, …, e6}, then determine an equivalent computation in which the recorded state does occur after the recording has commenced. [13 marks]