Seventh Semester B. E. (Computer Science and Engineering) Examination

DISTRIBUTED SYSTEMS

Time : 3 Hours] [Max. Marks : 60

Instructions to Candidates :—

(1) All questions carry marks as indicated against them.
(2) Assume suitable data wherever necessary and clearly state your assumptions.
(3) Draw well labeled diagrams wherever required.

1. (a) At 10 : 27 : 540 (hr, min, 1/100 sec.), server B requests time from the time – server A. At 10 : 27 : 610, server B receives a reply from timeserver A with the timestamp of 10 : 27 : 375.

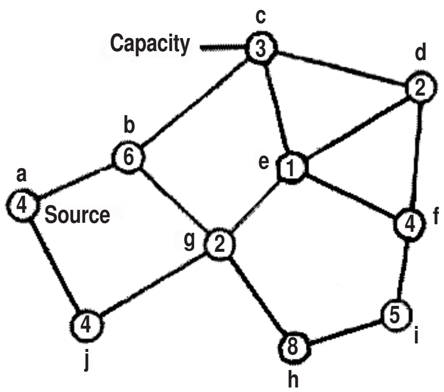
   (i) Find out the drift of B's clock with respect to the time - server A's clock (assume there is no processing time at the time - server for time service).

   (ii) Is B's clock going too fast or too slow ? If the answer is yes, by how much is the clock going too fast or too slow ?

   (iii) How should B adjust its clock ? 3 (CO 1)

   (b) If a site S has to broadcast message M to a set of sites, will the Schiper - Eggli - Sandoz protocol for causal ordering of messages work properly without modification ? If your answer is yes, justify your answer. If your answer is no, then show a counter - example and give the necessary modifications to the algorithm. 3 (CO 1)

   (c) Discuss and trace (for the figure given below) the leader election algorithm for electing a coordinator in a wireless environment.



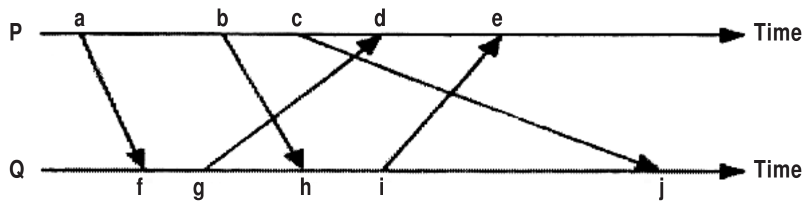
4 (CO 1)

OR

(d) Calculate logical clock values of events **a - j** between two processes P and Q using

(i) Lamport logical clock.

(ii) Vector clock.



4 (CO 1)

2. (a) In Ricart - Agrawala algorithm, the critical section is accessed according to the increasing order of timestamps. Does the same hold true in Maekawa's algorithm ? Prove your answer with the help of an example. 3 (CO 2)

(b) Given a distributed system with three processes P1, P2 and P3 at three different sites. Consider the scenario below :

- Both P1 and P2 are requesting for critical section while P3 has no need at this time.
- The timestamp of P1's request is 47 and that of P2 is 32.

Compare the working of Lamport's and Ricart - Agrawala's algorithm for the above scenario and list in detail all the events that follow until both processes complete their critical sections. 5 (CO 2)

OR

(c) There are three processes, p1, p2 and p3. p1 and p3 seek mutually exclusive access to a shared resource. Trace the working of Suzuki - Kasami's algorithm and show how the contents of RN, LN and Queue are modified at each step. How many messages are needed for CS invocation in this case ? 5 (CO 2)

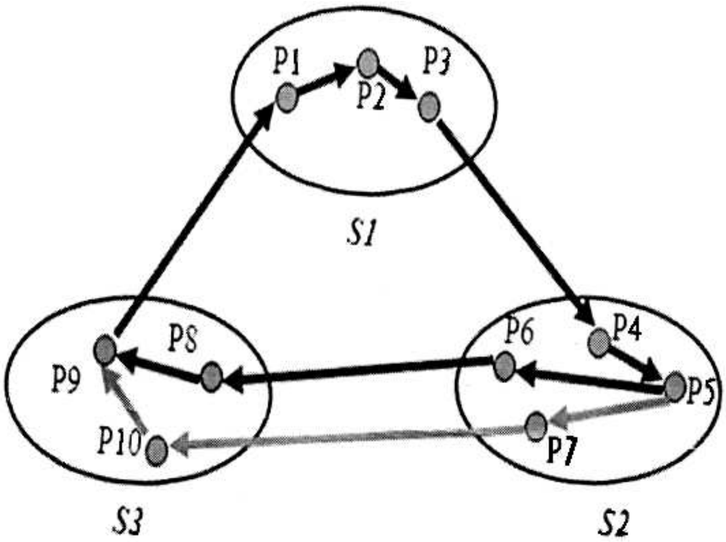
(d) In Raymond's tree algorithm for distributed mutual exclusion, a non - leaf node only forwards a Request message if Request_q is not empty. Explain why it does this. 2 (CO 2)

3. (a) Obermarck's algorithm detects false deadlocks but Chandy Misra Hass algorithm doesn't have this problem for AND - model. Explain using examples. 3 (CO 2)

(b) Is it possible to reach an agreement if $n = 7$ and $m = 2$? Trace the Oral Message algorithm when the source is one of the faulty processors. How many messages are required to reach an agreement ? Draw the recursion tree. 5 (CO 2)

OR

(c) Compare the performance of Mitchell Merritt algorithm with Chandy Misra Hass algorithm for the following scenario :



5 (CO 2)

(d) How is the solution to Byzantine Agreement problem used to solve the Consensus agreement problem ? 2 (CO 2)

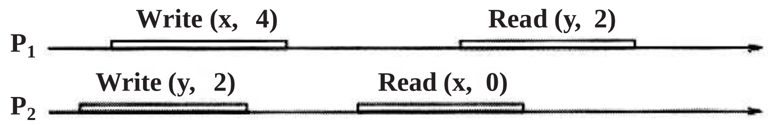
4. (a) Discuss how session semantics can be implemented in distributed file systems. 3 (CO 3)

OR

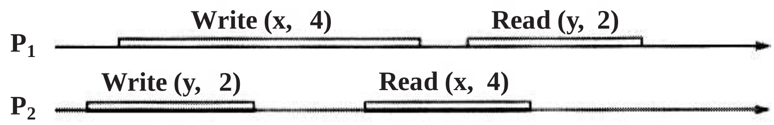
(b) What are the benefits and limitations of spawning multiple threads in a file server to handle file processing activities of different clients ? Describe the synchronization requirements of these threads. 3 (CO 3)

(c) For 2 processes P1 and P2, state in each case whether they are/not sequentially consistent and/or linearizable. Give reasons.

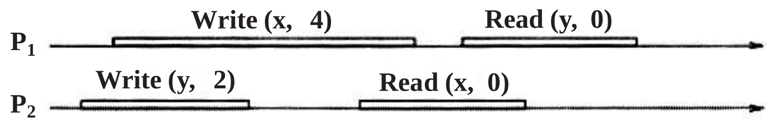
(i)



(ii)



(iii)



3 (CO 3)

(d) Explain why a page is transferred twice when a double fault occurs in IVY's coherence protocol. 4 (CO 3)

5. (a) A sender-initiated distributed scheduling algorithm uses the following protocol to transfer a process one node to another :

(i) A sender polls all other nodes in the system in search of a receiver node.

(ii) It selects a node as a prospective receiver and sends it a "lock yourself for a process transfer" message.

(iii) The recipient of the message sends a no reply if it no longer a receiver. Else it increases the length of its CPU queue by 1 and sends a yes reply.

(iv) The sender transfers a process when it receives a yes reply.

(v) If it receives a no reply, it selects another node and repeats steps (ii) to (v).

Does this protocol avoid instability at high system loads ? Justify your answer. Also construct a simple example to validate your answer. 5 (CO 3)

(b) How does the use of statevector at each node avoid preemptive task transfers in the stable sender initiated algorithm ? How does this algorithm avoid unstability at high load conditions ? Give reasons for your answer. 5 (CO 3)

6. Solve any **Two** :—

(a) Show by example that, in the Koo–Toueg checkpointing algorithm, if processes do not block after taking a tentative checkpoint, then global checkpoint taken by all processes may not be consistent. 5 (CO 4)

(b) Discuss the pros and cons of the tagged and partitioned approaches to implement a capability–based protection system. 5 (CO 4)

(c) In a decentralized variant of the two-phase commit protocol the participants communicate directly with one another instead of indirectly via the coordinator.

In Phase 1, the coordinator sends its vote to all the participants.

In Phase 2, if the coordinator's vote is No, the participants just abort the transaction ; if it is Yes, each participant sends its vote to the coordinator and the other participants, each of which decides on the outcome according to the vote and carries it out.

Calculate the number of messages and the number of rounds it takes. What are its advantages or disadvantages in comparison with the centralized variant ? 5 (CO 4)