Subject :- DC Experiment / Tutorial / Assignment No. :- 3 Page :- 01

Name: Virendra kalwar
Roll No. 57

Remark	Sign

Q.1 Fill in the blanks:-

(i) The interval between two consecutive transits of the sun is called the solar day.

(ii) In Berkeley Algorithm, the time server is reference time server.

(iii) To synchronize logical clocks, Lamport defined a relation called happen before.

(iv) In Ricart - Agrawala's algorithm, messages per entry/exit required is n - 1

(v) In Tree - Based algorithm, processes are organized in a directed tree structure.

Q.2 Choose Correct Options:

(i) In distributed system logical clock is associated with ____

ans. (c) Each Process

(ii) When a process is executing ~~ins~~ its critical section ____.

ans. (b) No other process can also execute in its critical section.

(iii) If timestamps of two events are same, then the events are ____

ans. (a) Concurrent

(iv) In the token passing approach of distributed systems, processes are organized in a ring structure. ____

ans. (b) Physically

(v) In distributed systems, election algorithms assume that a ____.

ans. (a) unique priority number is associated with each active process in system. Persistents objects.

Subject :- Experiment / Tutorial / Assignment No. :- Page :-

(6) In Lamport's clock, if a is event of sending the message by one process and b is event of receiving same message by other process then a → b. If C(a) is time value of event a then ______

ans. (b) C (a) := C (b)

(7) In case of failure, a new transaction coordinator can be selected by ______

ans. (c) both bully and ring algorithm.

(8) The holding (H) parameter in Raymond's tree based algorithm represents the process /node is ______

ans. (c) Holding information about node having a path to the token node.

(9) Number of messages required in Ricart Agrawala Algorithm is?

ans. (a) 3(n-1)

(10) Suzuki & Kasami is a?

ans. (b) Non-token based algorithm.

Q.3 Answer the following Questions in brief:-

(i)

Ans. To find the maximum clock skew, we need to determine the maximum difference in clock readings between the two machines at any given time.

For the machine that ticks 1000 times per millisecond, it clock advances by 1 tick every 1/1000 millisecond; which means its clock advances by 1 millionsecond every 1000 ticks.

For the machine that ticks 990 times per second; its clock advances by 1 tick every 1/990 millisecond; which means its clock advances by 1 millisecond every 990 ticks.

Now, lets calculate the time taken by both machines to advance by 1 milliseconds.

Subject :- Experiment / Tutorial / Assignment No :- Page :-

For the first machine : $\frac{1}{1000}$ milliseconds per tick ; so 1000 ticks per millisecond.

For the second machine : $\frac{1}{990}$ millionseconds per tick, so $\frac{990}{1000}$ ticks per millionsecond.

The difference in ticks per milliseconds between the two machines is $1000 - \frac{990}{1000} = \frac{10}{1000}$

Now, if we consider that both machines start at the same time & observe the difference after 1 minute (60,000 milliseconds) ; maximum clock skew would be :

Max Clock Skew $= \frac{10}{1000} \times 60000 = 600$ ticks.

Therefore, the maximum clock skew that will occur is 600 ticks.

Subject :- Experiment / Tutorial / Assignment No. :- Page :-

② ans. When two processes simultaneously detect the demise of the Coordinator initiate an election using the bully algorithm, they both send out election messages to other processes to declare themselves as candidates. Since both are unaware of each other's actions, this can lead to a scenario where multiple processes declare themselves as the new Coordinator simultaneously. This results in a race condition, where the process with the highest priority (usually the highest process ID) will eventually be elected as the new coordinator, while the other processes will recognize the newly coordinator and terminate their own election processes.

③ Explain Logical clock with example.

ans. (i) In distributed systems, logical clocks are used to establish a partial ordering of events based on casually rather than physical time. One common logical clock algorithm is Lamport's logical clocks.

(ii) In a distributed system, clocks need not be sychronized absolutely.

(iii) If two processes do not interact, it is not necessary that their clocks be synchronized because the lack of Synchronization would not be observable and thus it does not cause problem.

(iv) It is not important that all processes agree on what the actual time is, but that they agree on the order in which events occur.

(v) Lamport clocks are a simple technique used for determining the order of events in a distributed system.

(vi) Lamport clocks provide a partial ordering of events – Specifically "happened - before" ordering.

(vii) If there is no "happened-before" relationship then the events are considered concurrent.

Subject :- Experiment / Tutorial / Assignment No. :- Page :-

- Rules of Lamport's Logical Clocks

(a) Happened Before Relation

- If a & b are events in the same process and a occurs before b, then $a \to b$.
- If a & b belong to two different processes and a sends message to b, then $a \to b$
- If $a \to b$ and $b \to c$, then $a \to c$.

(b) Logical Clocks Concept:-

- Maintaining a common clock or set of perfectly synchronized clock is difficult.
- So, local timestamp is given for happened before relation.

(4) Explain performance analysis of different mutual exclusion algorithms.

ans.

Parameters	Centralized	Distributed	Token ring
(i) Election	one process is elected as coordinator	Total ordering of all events in the System	Uses token for entering critical section.
(ii) Delay in message times	Delay for messages is 2 messages	Delay for messages is $2(n-1)$	The time varies from 0 to $n-1$ tokens.
(iii) Starvation	No Starvation	No Starvation	No Starvation.
(iv) Complexity	Easy to implement	Complicated process	Implementation is easy
(v) Expenses	Less	More	Less
(vi) Robustness	More	Less	More
(vii) Problems	Entire System can go down due to single point of failure ~~bottleneck~~	N points of failure	Detecting the lost token and regeneration is difficult.

Subject :- Experiment / Tutorial / Assignment No. :- Page :-

5) Explain Suzuki-kasami's broadcast algorithm for mutual exclusion in distributed system.

ans. (i) Suzuki-kasami's broadcast algorithm is a token based algorithm.
(ii) It is used for achieving mutual exclusion in distributed systems.
(iii) This is a modification to Ricart-Agrawala algorithm.
(iv) In this algorithm a REQUEST and REPLY message are used for attaining the critical section.
(v) The process holding the token is the only process able to enter its critical Section.
(vi) The process holding the token is the only process able to enter its critical section.
(vii) Each process maintains an array request.
Req [j] denotes the sequence no of the latest request from process j.
(viii) Additionally, the holder of the token maintains an array last.
Last [j] denotes the sequence number of latest visit to Critical Section for process j.

Algorithm :-
(A) Requesting the Critical Section (CS) :-
(i) If the site does not have token, then it increases its sequence number Req [i] and sends a request (i, sn) message to all other sites (sn = Req_i [i]).
(ii) When a site S_j receives this message, its sets Req_j[i] to Max (Req_j [i] ; sn). If S_j has idle token, then it sends the token to S_i, if Req_j [i] = Last [i] + 1
(B) Releasing the CS :-
(i) When done with the CS, site S_i sets Last [i] = Req_i [i]
(ii) For every site S_j, whose ID is not in the token queue, it appends its ID to the token queue if Req_i[j] = Last[j] + 1.
(iii) If the queue is not empty, it extracts the ID at the head of the queue and sends the token to that site.

Subject :-

Experiment / Tutorial / Assignment No. :- Page :-

Executing the CS

Site S_i executes the CS when it has received the token.

Q⑤ Explain Bully election algorithm with example.

ans. It is the method for dynamically selecting a coordinator or a leader from a group of processes.

The process with the highest process id from amongst non-failed process is selected as a coordinator/Leader.

Assumptions :-

(i) Assume that system is Synchronous:

(ii) The process may fail at any time including the execution of the algorithm.

(iii) There is a failure detector that detects the failed process.

(iv) Message deliever between processes is reliable.

(v) Each process knows its own process ID.

Messages Types

(i) Election Message :- Send to announce the election.

(ii) OK Message :- Process responds to election msg.

Algorithm

(i) Suppose P sends the message to the Coordinator.

(ii) If the Coordinator does not respond to it then it is assumed that the coordinator has failed.

(iii) Now, P sends an election message to every process with a high process ID than itself.

(iv) It waits for the response, if no one responds for the time interval T then P selects itself as a leader/coordinator.

(v) If ~~none~~ one of higher ID responds then it takes over the election.

CamScanner

Subject :- Experiment / Tutorial / Assignment No. :- Page :-

(vii) Then the process gets an election message from one of its lower IDs.

(vii) The receiving process sends an OK Message to the sender to indicate that he is alive and will take over.

(viii) Eventually, all the processes give up a part of one and that one will be a new coordinator.

(ix) The new coordinator announces its victory by sending a coordinator Message.

Example

Step 1 :- Assume 7 processes & processor 4 identifies that the leader is not responding to the processor 4 sends the election message to all the processors which have ID>4.

Step 2 :- As the process 4 sends to message 5,6,7 and in return, process 5,6 sends an OK Message to process 4.

Step 3 :- As the process 5-6, sends the OK Message to the process 4 then process 4 drops from the election and the process 5 sends the Election Message to all the processors which have ID>5.

Step 4 :- After process 5,6 sends Election Message the process 6 replies OK Message to the process 5 and then process 5 drops from the election.

Step 5 :- Finally after ~~the~~ sending the OK Message to process 5 the process 6 declare itself as a Leader and sends the Coordinator Msg to all the processors.

This is all about Bully algorithm and time Complexity is

Worst Case :- Initiator with lowest ID - $O(n^2)$

Best Case :- Initiator with higher ID - $O(n-1)$