~~Name~~ →

Date

Page

Theory Assignment - 2

Name → Himanshu Gupta

Roll No. → BT17CSE033

Course → DCS

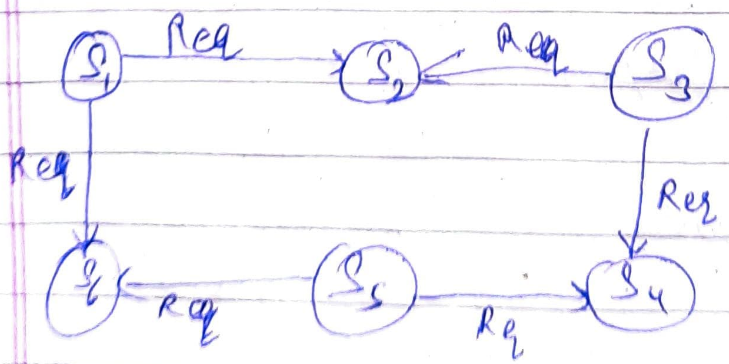
Semester → 7th sem, CSE

Q1. Maekawa's mutual exclusion algorithm

Since Maekawa's Algorithm is quorum based approach in which a site requests permission from a subset of sites which is called quorum.

A-a) Yes, it can deadlock because it is a request queue based mechanism in which requests are not prioritized by their timestamps. and until reply come, a site is exclusively locked by other sites.

For Example :→



Here, sites S_1, S_3, and S_5 simultaneously invoke mutual exclusion.

So,

So, request set ~~for each~~ be like

$R_1 = \{S_1, S_2, S_3\}$

$R_2 = \{S_2, S_3, S_4\}$

$R_5 = \{S_4, S_5, S_6\}$, $R_6 = \{S_5, S_6, S_1\}$.

From the figure, following case ~~may~~ happened

(i) S_2 has been locked by S_1, which is forcing S_3 to wait at S_2

(ii) S_4 has been locked by S_3, which is forcing S_5 to wait at S_4

(iii) S_6 has been locked by S_5, which is forcing S_1 to wait at S_6.

(b) So, it is a type of Resource deadlock. because a site waits until it receives the REPLY for its request.

Ans Solving of Deadlock Problem

Maekawa's Algorithm solves deadlocks by prioritized the request by their timestamp and applying a lock on a site if the timestamp of its request is larger than the timestamp of some other request waiting ~~of~~ for same lock.

Deadlock handling requires three types of messages :-

(i) **Failed** :- A failed massage from site S_i to site S_j indicates that S_i can not grant S_j's request because it has currently granted permission to higher priority request.

(ii) **Inquire** :- An INQUIRE message from S_i to S_j to check if it has succeeded in locking all the sites in its request set.

(iii) **YIELD** :- A YIELD message to check if S_i is returning the permission to S_j

Handling Process :-

→ If request (ts, i) from site N_i gets blocked at site N_j, then a FAILED(j) message is sent to N_i if N_i's request has lower priority. Otherwise S_j sends an INQUIRE(j) message to site N_k.

→ Now, if N_k has either got a FAILED message from atleast one of its quorum set, or if N_k has not received a Grant message from the site where it sent the YIELD message.

Now, N_k passes a YIELD(3) message to N_j as a reply to the INQUIRE(2) message.

→ In response to a YIELD(k) message from site N_k, N_j assumes as if it has been released by N_k, places the request of S_k at appropriate location in the request queue, and sends a GRANT(j) to top request's site in the Queue. In this way, Maekawa's algorithm requires extra message to handle deadlocks.

Ans 3 Given the no. of nodes in the system as 10, Now constructing of Request set.

→ Since, for each ~~Nod~~ site S_i, there is a seperate Request set R_i

So, we need total 10 request sets

i.e. $R_1, R_2, R_3 \ldots R_{10}$

Conditions to be followed:-

(i) $\forall i \,\forall j$; $i \neq j$, $1 \leq i, j \leq N$; $R_i \cap R_j \neq \phi$

(ii) $\forall i$; $1 \leq i \leq N$; $S_i \in R_i$

(iii) $\forall i$; $1 \leq i \leq N$; $|R_i| = K$

(iv) Any site S_i is present in exactly K request sets., $1 \leq i \leq N$

From the Algorithm,

$N = K(K-1)+1$, N = No. of Nodes
K = Size of Request set.

$\Rightarrow K = O(\sqrt{N})$

For our problem, $N = 10$

So, $10 = K(K-1)+1$

$\Rightarrow 10 = K^2 - K + 1$

$\Rightarrow K^2 - K - 9 = 0 \Rightarrow K = \frac{1 \pm \sqrt{1+36}}{2}$

$\Rightarrow K = \frac{1+\sqrt{37}}{2}$ or $\frac{1-\sqrt{37}}{2}$ → can not negative i.e. $k \cong 3.54$

Taking ceiling of k, we get

$k \cong \lceil 3.5 \rceil \cong 4$

So, each request set need to contain 4 sites following all conditions.

So, from finite projection plane theorem possible Request sets are.

$R_1 = \{N_1, N_2, N_4, N_6\}$

$R_2 = \{N_2, N_3, N_5, N_7\}$

$R_3 = \{N_3, N_4, N_6, N_8\}$

$R_4 = \{N_4, N_5, N_7, N_9\}$.

$R_5 = \{N_5, N_6, N_8, N_{10}\}$

$R_6 = \{N_1, N_6, N_7, N_9\}$

$R_7 = \{N_2, N_7, N_8, N_{10}\}$

$R_8 = \{N_1, N_3, N_8, N_9\}$

$R_9 = \{N_2, N_4, N_9, N_{10}\}$

$R_{10} = \{N_3, N_1, N_5, N_{10}\}$

~~By Utsav~~