Name : Shalini N
Reg. NO : 20509998

Distributed System

Lamport's distributed mutex algorithm

p1 → p3 || p2 || p1 → p2 || p1

1	p1 □□□	p2 □□□	p3 □□□
2	(24 p1) (2 u p2) Req (1, p1, {p2, p3})	(2 u p1)(2 u p2) Req (1, p1, p2)	(2 u p3)(2 u p2) Req (1, p2, p3)
3	add to queue	Rep (3, p2, p1)	Rep (p1, 3 p3, p1) add
4	Rel (2, p1, {p1, p2}) remove	Rel (6, p1, p2) remove	Rel (6, p1, p3) remo
5	Req (8, p1, {p2, p3}) add	Req (8, p2, {p1, p3}) add	Req (8, p3, {p1, p2}
6	Req (8, p2, p1) add	Req (8, p1, p2) add	Req (8, p1, p3) add
7	Req (8 p3, p1) add	Req (8, p3, p2) add	Req (8, p2, p3) add
8		Rep (11, p2, p1)	Rep (11, p3, p1)
9			
10	Rep (11, p2, p1)		
11	Rep (11, p3, p1)		
12			
13	CS		
14	Rep (151, p1, {p2, p3}) remove	Rel (51, p1, p2) remove	Rel (15, p1, p3) remove.
15			
16			CS
17		Req (p8, p2 {p1, p3}) remove.	
18	Rel (1, p2, p1) remove.		Rel (8, p2, p1) remov
19			
20			
21	Rel (21, p3, p2) remove	Rep (21, p3, p2) (remove)	
22	Rel (24, p1 {p2, p34}) add		
23		Req (24. p2 {p1, p34} remove	Req (24, p1, p3) add
24	Req (24, p2, p1) add		
25		Req (24, p1, p2) add	Req (21, p2, p3)
26	Rep (26, p2, p1)		
27	Repl (26, p3, p1)	Rep (26, p2, p1).	Rep (26, p3, p1)

	P1	P2	P3
28	CS		
29	Rel(29, p1 {p2, p3} remove.		
30		Rel(19, p1, p2) remove.	Rel (29, p1, p3) remove
31			
32		CS	
33	Rel (32, p2, p1) remove.	Rel(32, p2, {p1, p3}) remove.	
34			Rel (32 p2, p3) remove.
35			
36			
37			
38.			

Lamport's algorithm achivers mutual excusion by sending req. rep and rel message accordingly.

Ricart agrawala algorithm

p1 → p3 || p2 || p1 → p2 || p1

	P1 (000)	P2 (000)	P3 (000)
1	Req (1, p1, p2, p3)		
2		Req (1, p11, p2)	Req (1, p1, p3)
3		Rep (3, p2, p1)	Req (3, p3, p1)
4	Rep (3, p2, p1)		
5	Rep (3, p3, p1		
6	CS		
7	Req (7, p1, {p2, p3})	Req (7, P2, {p1, p2})	Req (7, p3 {p1, p2})
8	Req (7, p2, p1) 010	Req (7, p1, p2)	Req (7, p1, p3)
9	Req (7, p3, p1) 011	Req (7, P3, p2)	Req (7, p2, P3)
10		Rep (10, P3, p2)	Rep (10, P3, {p1, p2} (000)
11	Rep (10, p2, p1)		
12	Rep (10, P3, p1		

13	CS		
14	Rep(11. p1 {p2, p3})	Rep (p1, p1, P2)	Rep (11, P1, p1)
15		CS	
16		Rep(18. P2. P3)	
17			Rep(18. P2, P3)
18			CS
19			
20			
21	Rel(22. P1 {P2, P3})	Req (22, P2, {P1, P3})	Req (22, P1, P3)
22	Req (22. P2, P1) (010)	Req (22, P1, P2)	Req (22, P2, P3)
23		Rep (22. P2, P1)	Req (21, P3 {P1, P2})
24			
25	Rep(21, P2, P1)		
26	Rep (21. P3, P1)		
27	CS		
28	Rep(28. P1. P2) (000)	Rep(28. P1. P2)	
29			
30		CS	

Ricart agarwala algorithm reduces the number of message sent by checking which replies have been deferred.