Date 6/04/22
Page 43

TUTORIAL 11

Q1> Compare the following MUTUAL EXCLUSION. Consider the following parameter to compare various mutual exclusion algorithms. You have to check and validate that these comparisions are correct.

① Message Exchanged : messages per entry/exit of critical section
- Centralized 3
- Ring $1 \rightarrow \infty$
- Multicast $2 \times (n-1)$

② Delay : Delay before entering critical section
- Centralized 2
- Ring $0 \rightarrow n-1$
- Multicast $2 \times (n-1)$

Q2> Compare and Contrast centralisation and distributed mutual exclusion.

③ Reliability :
- Centralized : co-ordinator crashes
- Ring : lost token, processor crashes
- Multicast : any process crashes.

Answers

1.> 1> Message Exchanged { message per entry/exit of cs }

a) Centralized – 3 – (Right)

Reason – It requires only three messages to enter and leave a critical region
① A Request [~~Grant to enter~~]
② Reply { Grant to enter }
③ Release to exit

Date 6 Apr
Page 4/1

(1) (b) Ring - $1 \rightarrow \infty$ = True

Reason - If every process constantly want to enter critical region, then each token passed will result in one entry and exit, for average of one message per critical region entered.
At other extreme, token may sometimes circulate for hours without anyone being interested in it. In this case, the number of messages per entry into critical section is Unbounded.

(1) (c) Multicase (Distributed) - $2*(n-1)$

Reason - ① The distributed algorithm requires $(n-1)$ request messages, one to each of other process, and additional $(n-1)$ grant messages, making total of $2\times(n-1)$.

② Delay (time before entering critical section)

(2) (a) Centralized - 2 - (Right)

Reason - It requires only two messages to enter the critical section - Request & Reply.

(2) (b) Ring - 0 to $(n-1)$ ∴ No delay

Reason - Time varied from 0 (token just arrived) & $(n-1)$ (token just departed)
∴ It needs to travel remaining $(n-1)$ nodes before reaching the current node.

(2) (c) Distributed - $2\times(n-1)$

∴ Messages are sent ~~after~~ one after the other
$(n-1)$ request + $(n-1)$ grant messages = $[2\times(n-1)]$

2.) Parameters	Centralized Algorithm	Distributed Algorithm
⇒ Election	① One process is elected as co-ordinator	① Total ordering of all events in the system
⇒ Message per entry/exit	② Requires 3 messages (Request Reply Release) to enter and exit critical section	② Requires $2*(n-1)$ messages
⇒ Delay in message time	③ 2 messages	③ $2*(n-1)$ messages
⇒ Mutual Exclusion	④ Guarantees mutual exclusion	④ Mutual exclusion guaranteed without deadlock
⇒ Starvation	⑤ No starvation	⑤ No starvation
⇒ Complexity	⑥ Easy to implement	⑥ Complicated process.
⇒ Used for	⑦ General allocation	⑦ used for small group processes that do not change group membership.
⇒ Problems	⑧ Entire system can go down due to single point of failure, Bottleneck	⑧ N points of failure
⇒ Expense	⑨ Less Expensive.	⑨ More expensive
⇒ Robustness (cope up with errors)	⑩ More Robust	⑩ Less Robust
⇒ Applications	⑪ - Application development (central server + many clients Express & django server) - Data Analysis - all data at one place.	⑪ - Cluster computing - many computers work together to acheive global goals - Grid computing (All resources are pooled for sharing)
⇒ Use case	⑫ centralized databases ✱ single player games - NFS, GTA Test servers	⑫ Service oreinted Architecture systems ✱ Multiplayer online games
⇒ Organizations using it	National Informatics centre (India) IBM	[Apple, Google, Meta]