Date

Name :- Ritu Rajput
Roll No. :- 2301010208.

Assignment - 04

OPERATING SYSTEM

Ques6. Distributed Deadlock detection.

Given:-

- S1 : P1 → P2, P3 → P4
- S2 : P2 → P5, P5 → P6
- S3 : P6 → P1

(a) Combine fragments into the global wait for graph:

From all Sites, the combined edges are:

P1 → P2, P3 → P4, P2 → P5, P5 → P6, P6 → P1

So, global wait for graph is

P1 → P2 → P5 → P6 → P1 and P3 → P4

(b.) Detect if a deadlock exists:

Yes

Cycle detected : P1 → P2 → P5 → P6 → P1

Thus, processes P1, P2, P5 & P6 are involved.

Date

(c) Distributed algorithm

Use the chandy - Misra - Haas ((MH) algorithm for distributed deadlock detection.

- Each site sends probe message (initiator, sender, receiver)
- If a probe returns to the initiator, a deadlock is detected
- This algorithm works efficiently without requiring a global co-ordinator.

7. Distributed File system

Given :-

- local access time = 5 ms
- Remote access time = 25 ms.
- Probability of being remote = 0.3

(a) Expected File access time

$E(T) = (1-0.3)(5) + (0.3)(25) = 0.7(5) + 0.3(25) =$

$3.5 + 7.5 = 11$ ms.

(b) Caching strategy :

Use client side caching with least Recently

Used (LRU) replacement

- Frequent accessed remote files are cached locally.
- Reduces remote access frequency, lowering average access time.
- LRU ensures cache freshness and avoid stale data build up.

(8.) Given:-

- Full checkpoint = 200 ms
- Incremental checkpoint = 50 ms
- Must recover within 1 sec. (1000 ms RPO)
- Total operation window = 10 sec.

(a) optimal mix.

To minimize overhead while ensuring quick recovery:

- Take 1 full checkpoint every 10 sec.
- Incremental checkpoints every 2 sec.

That gives:

- 1 full checkpoint = 200 ms
- 4 incremental checkpoint = $4 \times 50 = 200$ ms.
- Total overhead = 400 ms in 10 s (4%)

Date

(b) Explanation :-

Incremental checkpoints save only changed state reducing data volume

Combining full + incremental ensures fast recovery (restore last full + few incremental).

(9) (a) Scheduling challenges & load balancing algorithms

Challenges :-

- Uneven global traffic (eg flash sales in one region)
- Latency due to geographical distribution.
- Dynamic scaling under sudden demand.

(b) Fault tolerance strategy :- Use Active-Active Replication across regions with:

- Data replication (synchronous | asynchronous) for redundacy.
- failover mechanism via global DNS routing
- ~~Meet~~ Regular checkpoints

Ques1-

Ans:- A race condition occurs when two or more entities try to change a shared resource simultaneously, leading to unpredictable results.

Eg:- (Real world): Two people editing the same document at once- one saves changes while the other overwrite them.

Ans 2.

Aspect	Peterson's Sol^{n}	Semaphores
Implementation	Software based algorithm for two process	Abstract data types implemented in OS
Complexity	Simpler logic but limited to 2 process.	More flexible, Supports multiple processes.
Hardware dependancy	Works purely in Software.	Depends on hardware supported atomic operations.

Date

Ans 3.

Advantage: Monitors provide automatic synchronization through mutual exclusion within the monitor. In multi core systems, they are easier to implement and maintain as synchronization is handled at a higher level, reducing the chance of programing errors.

Ans 4.

Starvation :- occurs when writers keep waiting indefinitely because continuous readers hold access to shared data.

Prevention :- Use write-prority- once a writer is waiting, block new readers until the writer finishes.

Ans 5.

Drawback :- Processes must request all resources at once before execution begins, leading to resources underutilization and reduced concurrency since some resources remain idle for long periods.