Name - Gurleen Kaur
class - Btech CSE
RollNo - 230101020 7

Date
DELTA Pg No.

Assignment - 04
Operating System

Que1. Race condition + real-World example + mutual exclusion.

A race condition occurs when two activi-tes access a shared resource at the same time and the final results depends on the order of execution

Real-world example: Two people writing on the Same whiteboard at the sametime may overwrite each other's content.

Mutual exclusion: Ensures only one person access the shared resource at a time, preventing conflicts.

Que2. Peterson's solution vs Semphores

Aspect	PeterSon's Solution.
• Implementation complexity	More complex, uses busy-waiting
• Hardware dependency	Pure software, no hard-ware support needed.

Que3. Advantage of monitors in multicores system.

Monitors automatically handle mutual exclusion using condition variables and locks, making them safer and easier to use in multicore systems compared to Semaphores, which require manual signaling

Ques4. Reader-Writer problem: Starvation + preventation
Starvation: If writes keep arriving, readers may never get access.
Preventation: Use fair Schedulling such as readers-writers with ~~familiar~~ fairness, where requests are ~~ser~~ served in arrival order.

Ques5. Deadlock preventation: drawback of eliminating "Hold & wait".
If Processes must request all resources at once many resources must remain unused for long periods, causing poor resource utilisation and system inefficiency.

Part - B.

Ques6. Distributed Deadlock Detection
Given fragements.
S1: P1 → P2, P3 → P4
S2: P2 → P5, P5 → P6
S3: P6 → P1

a) Global wait-for graph.
P1 → P2 → P5 → P6 → P1 (Graph)
P3 → P4 (no cycle)

b) Deadlock detection.
A deadlock exists in the cycle:
Processes involved = {P1, P2, P5, P6}

c) Algorithm.

Use the Chandy-Misra-Haas distributed deadlock detection algorithm; which detects cycles by passing probe messages.

Que 7. Distributed file System Performance.

Local access = 5 ms

Remote access = 25 ms.

Probability remote = 0.3

a) Expected file access time:

$E = 0.7 \times 5 + 0.3 \times 25 = 3.5 + 7.5 = 11$ ms.

b) Caching strategy.

Use client-side caching with LRU or TTL-based invalidation.

Justification: Reduces remote access frequency and improves responsiveness.

Que 8. Checkpointing in a concurrent Systems.

Full checkpoint: 200ms.

Incremental checkpoint = 50ms.

RPO requirement = must cover within 1 second.

a) Optimal mix.

Over a 10-second period, use one full checkpoint + several incremental checkpoint

Eg:

- 1 full checkpoint every 10s.
- Incremental checkpoints, every 1s.

This keeps recovery time < 1 seconds.

b) Reasoning

Full checkpoint Captures the entire state infrequently, and incremental checkpoints keep recent changes recorded often enough to statisfy the RPO.

Que9. Case Study - Global E-Commerce Platform

a) Distributed schedulling challenges & Suitable algorithm.

Challenges:

- Sudden flash sale spikes.
- Uneven user load across regions.
- Server overload and delays.

Algorithms:

Use dynamic load balancing with work stealing or a distributed load balancer.

b) Fault tolerence Strategy (with RTO & PRO)

- Use Geo-redundant data centers.
- Replicate data accross regions.
- Use automated failover to meet RTO.