Assignment - 4

Submitted by : Ayushi Sarkar
2301010448
B.Tech CSE (Sec-7)

Operating System
(Sem - V)

Part - A

1.) Race Condition with real-world example and mutual exclusion

Race Condition : occurs when multiple entities access shared resources simultaneously, causing inconsistent results.

Example (non-computing) : Two cashiers updating the same ~~class~~ cash register balance at the same time. Both read ₹100, add ₹50 and ₹30 respectively, and write ₹150 and ₹130. The final balance is incorrect.

Mutual Exclusion : ensures only one cashier updates the register at a time, preventing conflicts.

2.) Peterson's Solution vs Semaphores

Feature	Peterson's Solution	Semaphores
→ Implementation complexity	Simple s/w - based solution using flags and turn variable	Requires OS - level support and system calls.
→ Hardware Dependency	works only on single-processor system with sequential consistency	works on multi-core systems, less hardware dependent.

3. Advantage of monitors over Semaphores in multi-core systems

→ Monitors encapsulate synchronization, reducing errors from manual semaphores handling.

→ Advantage : Automatic mutual exclusion avoids race conditions in multi-core systems, simplifying concurrency management.

4. Reader - Writer Starvation & Prevention

Starvation : Writers may starve if readers continuously access the resource.

Prevention : Use a writer - priority protocol or queue - based scheduling to ensure writers eventually get access.

5. Drawback of eliminating "Hold and Wait" in deadlock prevention

Elimination : Processes must request all resources at once.

Drawback : can lead to resource underutilization and longer waiting times, reducing system throughput.

Part - B

6. Distributed Deadlock Detection Simulation

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

(a) Global Wait - for Graph :

P1 → P2 → P5 → P6 → P1

P3 → P4

(b) Deadlock Detection :

Cycle exists : P1 → P2 → P5 → P6 → P1

Deadlock Processes : P1 , P2, P5, P6 .

(c) Distributed Algorithm :

Chandy- Misra - Hass algorithm for distributed deadlock detection.

7. Distributed File System Performance

Given : local access : 5 ms
remote access : 25 ms
remote probability : 0.3

(a) Expected file access time E[T] :

$$E[T] = (0.7 \times 5) + (0.3 \times 25) = 3.5 + 7.5 = 11 ms$$

(b) Caching strategy :

→ Client-side caching - store frequently accessed remote file locally.

→ Justification - reduces repeated remote access latency and network load.

8. Checkpointing in a Concurrent system

Given : Full : 200 ms
Incremental : 50 ms
RPO : 1 s

(a) Optimal Mix : perform 1 full checkpoint every 1 s, followed by incremental checkpoints every 250 ms.

(b) Reasoning : Incremental checkpoints are faster, reducing overhead. Full checkpoints ensure complete recovery. Combination meets RPO without blocking the system.

9. Case Study - Global E-Commerce Platform

(a) Distributed Scheduling Challenges:
Flash sales create sudden load spikes, uneven across regions.
Suitable Algorithm : Weighted Round Robin or Dynamic Load Balancing using Least-Loaded Server.

(b) Fault Tolerance Strategy :
Geo-redundant deployment : replicate services across multiple data centres.

RTO/RPO : Use synchronous replication for critical services (low RPO) and asynchronous replication for less critical services (acceptable RTO).

Result : High availability even if a region fails.