Assignment - 4

Submitted by: Navneet kumar
2301010479
BTech CSE (G)

Operating systems
Sem - V

Part - (A)

1. Race condition with real world example & mutual exclusion.

Race condition: occurs when multiple entities access shared resources simulataneously, causing inconsistent results.

Example (non-computing): Two cashier updating the same cash register balance at the same time. Both read ₹100, add ₹50 & ₹30 respectively. & write ₹150 & ₹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 Solutions	Semaphores
→ Implementation complexity	Simple s/w - based solution using flags & then variable	Requires OS - level support & 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 & Wait" in deadlock prevention.

Elimination: Processes must request all resources at once.

Drawback: can lead to resource under utilization & longer waiting times, reducing system throughout.

Part-B

6) Distributed Deadlock Detection simulation

Given: S1 : $P1 \rightarrow P_2$, $P_3 \rightarrow P_4$

S2 : $P_2 \rightarrow P_5$, $P_5 \rightarrow P_6$

S3 : $P_6 \rightarrow P_1$

(a) Global wait for graph:

$P_1 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \rightarrow P_1$

$P_3 \rightarrow P_4$

(b) Deadlock Detection:

Cycle exists: $P_1 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \rightarrow P_1$

Deadlock Processes: P_1, P_2, P_5, P_6

(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 & network load.

8) Checkpointing in a concurrent system

Given: Full: 200 ms

Incremental: 50 ms

RPO: 1s

(a) Optimal mix: perform 1 full checkpoint energy 1s, 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 schedulling challenges:

Flash sales create sudden load spikes uneven accross regions.

Suitable Algorithm: Weighted Round Robin or dynamic load balancing using least-loaded server.