—

Operating System
Assignment - 4

Harshit Chaurasia
2301010210

Ans1) Race Conditions and Mutual Exclusion

→ Race Condition : Occurs when the result of operations depends on the specific , unpredictible order of multiple processes / thread of ~~multiple~~ accesing a shared resource.

→ Mutual Exclusion : Solves this by ensuring only one process can enter the critical section (the code that manipulates the shared resource) at a time. The first process acquires a lock , performs the complete operation, and releases the lock, forcing the second process to wait.

Ans2) Peterson's Solution vs Semaphores

Feature	Peterson's Solution	Semaphores
→ Implementation Complexity.	Low to Moderate. Pure Software logic using shared variables	Moderate to High. Requires as Kernel support for waiting queues and Systems Calls (wait () / signal ())
→ Hardware Dependency	Low. Primarily dependent on memory consistency / barrieers on modern CPUs	High. Relies on atomic hardware instruction (Pure Test and Set) for Kernel implementation.
→ Scope	Limited to mutual exclusion for two processes	General Synchronization for N processes.

Ans3) Advantage of Monitors in Multi - Core System.

→ Monitors automatically encapsulate the shared data and all necessary locking / Synchronization logic within a single construct.

→ This design ensure that Synchronization is centralized and compiler - managed , making it virtually impossible for a programmer to forgot to aquire or release a lock that could lead to deadlocks or subtle errors accross

c) Suitable Distributed Algorithm.

The Chandy – Misra – Haas (CMH) Algorithm

→ Mechanism : A process waiting for a resource initiates a probe message. If the probe returns to its initiator, a cycle is confirmed.

Ans7) Distributed File System Performance

a) Expected File Access Time

$E[T] = (5ms \times 0.7) + (25ms \times 0.3) = 3.5ms + 7.5ms = 11.0ms$

b) Caching Strategy

→ Suggested strategy : Client - Side Caching with Write - Back Policy.

→ Justification: Remote Access (25ms) is very expensive. Write-Back caching minimizes remote access penalty by satisfying subsequent reads locally and batching writes before sending them to the server.

Ans8) Checkpoint Optimization for RPO

a) Optimal Mix proposal (over 10 seconds)

The RPO is 1 second

Type	Count	Overhead (ms)
Full checkpoint (FC)	2 (at 0s and 10s)	$2 \times 200 = 400$
Incremental (IC)	9 (every 1s in between).	$9 \times 50 = 450$
		850 ms

Total Overhead

b) Explanation of Reasoning

- RPO constraint : An incremental checkpoint (IC) must run every 1 second.
- Minimal overhead : This mix maximizes the use of the low cost ICs (50ms) while using high cost FCs (200ms)

multiple cores.

Ans4) Reader - writer Starvation and Prevention

- How Starvation Occurs: In a Readers - Preference solution, if a continuous stream of new Readers keeps arriving, they are always given priority over a waiting writer. The writer may be indefinitely delayed because the resource is always busy with reading.

→ Prevention Method: Implement a Writer - Preference or Fair Scheduling policy. A writer - Preference scheme prevents any new Readers from starting once a writer is waiting. After the currently active Readers finish, the waiting writer is waiting. After the currently active Readers finish, the waiting.

Ans 5) Drawback of Eliminating "Hold and Wait".

→ Practical Drawback: low Resource Utilization. A process must hold resources from the start, even if it won't use them until the very end. for that entire duration, the resource sits idle, unavailable to other processes that could be using it, leading to poor system throughput and efficiency

Ans6) Distributed deadlock Detection Simulation.

a) Global Wait- for Graph (WFG)

Combine the local fragments ($P_1 \to P_2$, $P_3 \to P_4$, $P_2 \to P_5$, $P_5 \to P_6$, $P_6 \to P_1$):

$P_1 \to P_2 \to P_5 \to P_6 \to P_1$ and $P_3 \to P_4$ ✓

b) Deadlock Detection and Processes Involved

→ Deadlock exists: Yes

→ Reason: There is a cycle in the Global WFG

→ Processes Involved: P_1, P_2, P_5 and P_6.

Ans 9) E-Commerce Case Study

a) Scheduling & load Balancing

→ Challange : Heterogenous Load & and high Synchronization Delay

→ Algorithm: Receiver - Initiated Load sharing

→ Justification : Decentralized approach where underloaded sites ask for work, quickly smoothing flash sale spikes without centralized ~~today~~. delay.

b) Fault Tolerance Strategy

→ Strategy : Active Replication using Active - Active Geo-Redundancy

→ Impact: Low RPO : Synchronous / Near - sync data replication ensures minimal / zero data loss.

· Low RTO : Global Load Balancer immediately routes traffic from the failed region to the healthy, active region, ensuring instant service availability.

Satinder
26/11/25