Assignment - 4

Name :- Srishti Mishra Roll no :- 2301410005

Course : Btech CSE (Cyber Security)

Ques1 :- Explain race conditions with a real-world example outside of computing, and show how mutual exclusion addresses it.

Ans :- Imagine two people trying to withdraw money from the same bank account at the same ATM at the same time. Both see the balance as 500. Each withdraws ₹500, and the system allows both, causing inconsistency. The final outcome depends on "who finishes first" not on correct logic → This is a race condition.

→ How mutual exclusion solves it :

A mutex would allow only one user to access/update the account at a time. When one withdrawal is in Progress, the other must wait. This prevents conflicting updates and ensures consistent balance.

Ques2 :- Compare peterson's solution and semaphores in terms of implementation Complexity and hardware depedency.

Ans :- Peterson's solutions :-

- Pure software-level algorithm ; Simple but error-prone ; limited to 2 processes.
- Requires strict assumptions about atomic read/write ordering → not reliable on modern multi-core CPUs.

→ Semaphores :-

- More general :- easier for programmers ; widely supported.
- Depends on hardware atomic operations like test-and-set ; explicitly supported by processors.

Ques 3. The producer-consumer problem can be solved using either semaphores or monitors. Identify one advantage of using monitors in a multi-core system.

Ans:- Monitors automatically combine mutual exclusion with condition variables, so synchronization is handled implicitly by the language / runtime.

Advantage:- They prevent low-level race conditions automatically, allowing threads on multiple cores to safely access shared data without manually managing semaphores, reducing bugs and complexity.

Ques 4:- For the Reader-writer problem, explain how starvation can occur and describe one method to prevent it.

Ans:- Starvation :- If the system gives priority to readers, a continuous stream of readers can keep arriving, causing a waiting writer to never get access → writer starvation.

Prevention Method :- use writer preference : once a writer requests the lock, no new readers are allowed to enter. Existing readers finish, then the writer is served. This ensures no indefinite waiting.

Ques 5:- In deadlock prevention, the "Hold and wait" condition can be eliminated. Explain one practical drawback of doing so in an OS.

Ans:- To prevent deadlock, processes must request all resources at once before starting.

→ Practical drawback :-

This leads to low resource utilization. A process may hold several resources it is not currently using, while others wait unnecessarily. This increases waiting time and reduces system throughput.

Ques 6:- Distributed Deadlock Detection Simulation:

Three sites S1, S_2 and S_3 have the following wait-for graph fragments:

- S1 : $P_1 \rightarrow P_2$, $P_3 \rightarrow P_4$
- S_2 : $P_2 \rightarrow P_5$, $P_5 \rightarrow P_6$
- S_3 : $P_6 \rightarrow P_1$

a) Combine these fragments to form the global wait-for graph.
b) Detect if a deadlock exists, and list the processes involved.
c) Suggest one distributed algorithm that could be applied to detect it.

Ans. (a) Combining all edges from the three edg fragments.

Graphically:

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

and seperately $P_3 \rightarrow P_4$

(b) Detect deadlock and list processes involved

→ A deadlock exist if there is a cycle in the wait-for-graph

→ we have the cycle $P_1 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \rightarrow P_1$ Process involved in the deadlock → P_1, P_2, P_5, P_6

(c) Suggest one distributed deadlock detection.

→ Chand - Misra - Hass (Prode) algorithm is stared choice for distruction deadlock detection

How it work :-

- A process that suspect it is blocked (or a site periodically) initates probe messages of from.

Ques 7 :- Distributed file system performance:

A DFS has the following characterstics:

- Average local files access time : 5ms
- Average remote file access time : 25ms
- Probability of file being remote : 0.3

a) Calculate the expected file access time.

b) suggest one caching strategy to improve performance, and justify your choice.

Ans. Given :

- local access time T_L = 5ms
- Remote access time T_R = 25ms
- Probability of file is remote p → 0.3

a) Expected time $E[T]$ is :

$$E[T] = P \cdot T_R + (1-P) \cdot T_L$$

Plug numbers:

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

b) Caching strategy suggestion.

→ Suggested strategy :- client site LRU caching with write-back for read-heavy file and validation TTL.

justification:-

- LRU (least Recently used) :- works well in practice because file access pattern often how temporal locality or recently used file are likely to be reused.

Ques 8:- In a concurrent system, a full checkpoint takes 200 ms, while an incremental checkpoint takes 50 ms. The system must maintain recovery capability within a 1-second recovery point objective (RPO).

a) Propose an optimal mix of checkpointing methods over a 10-second period to meet the RPO with minimal overhead.

b) Explain your reasoning.

Ans:- a) Proposed optimal mix

Steps :-

→ Take one full checkpoint every 10s.

→ Take incremental checkpoint every 1s between full checkpoints.

Total Checkpoint overhead periods :-

- Fulls : 1 X 200 ms = 200ms
- incrementals : 9 X 50ms = 450 ms

Total = 650 ms per 10 s

→ average overhead :- 65 m/s

b.) Reasoning :-

- RPO constraint : The system must be prepared to restore a state no older than 1s ; Therefore an incremental Checkpoint must be taken at least once per second.
- Full checkpoint are expensive (200 ms). Doing them less frequently reduces heavy overhead full checkpoint reduce recovery time because only one full.
- Trade off :- Frequent incremental add modest overhead but keep RPO tight Periodic fulls keeps recovery efficient.

Ques9:- Case study – Global E-Commerce Platform:

A global e-commerce company runs its services using a distributed operating system deployed across multiple continents.

Ans.

a.) Key challenges:-

1.) Sudden bursty traffic :- Requests spike order of magnitude within seconds.

2.) Geographic latency & data locality:- User should be served from the nearest region when possible.

3.) Hot-spots / skewed load:- Certain products / regions get disproportionate attention.

4.) Fairness & SLA gurantees:- Mission-critical request (Payment) must get prioritized.

b.) Fault-tolerance strategy :-

1. Use multi-region Deployment.
2. Replicate data based on RPO Requirments.
3. Enable Automated failover to meet RTO.
4. Maintain Cross-Region Backups & Regular DR Drills.