Assignment - 4

Ans-1) A shared Family notebook.

~~Imagine~~ The Race condition:

1) Person A checks list and sees "milk, eggs"
2) At the same time, Person B checks list and also sees "milk, eggs".
3) Person A adds "bread" write down "milk, eggs, bread."
4) Person B adds "butter" and, working from their original mental copy, overwrites the list with "milk, eggs, butter."
5) Result: The item "bread" is lost. The final list is wrong because both people were updating the shared resource w/o coordination.

How Mutual Exclusion addresses it:

The solution: A lock on the notebook.

classmate
Date
Page

A lock is placed on the notebook. The rule is! you must have the lock to read or write the list.

Ans - 2)

Feature	Peterson's Solution	Semaphores
Implementation complexity	High. The programmer must manually and correctly code the entire algorithm for each critical section	Low. The OS provides ready-to-use wait() and signal() operation
Hardware Dependency	High. Relies on atomic hardware instructions for correctness and is vulnerable to modern CPU memory reordering	Low (for the user). The hardware dependent complexity is hidden inside.

Answer-3) One advantage of using monitors is that they integrate the lock and condition variable, making the code less error-prone.

- with semaphores, a programmer can easily cause deadlock by incorrectly ordering the wait operations

- with Monitors, the lock is automatically acquired on entry and released during condition waits. This atomic "check-and-wait" operation prevents the deadlock scen comm in smphore-based solution, leading to more robust code on multi-core systems where concurrent access is the norm.

Ans-4) How starvation occurs
In the common "Readers-preference" solution, a continous stream of new readers can enter the critical section as long as at least one readre is inside. A waiting writer must wait for all readers to finish, which may never happen if new readers keep arraving.

One method to Prevent it:
Implement a "fair" sett" using a turnstile (like a semaphore).

- Both readers and writers must wait on this turnstile semaphore.
- A writer who acquires it will block all new readers from starting until the writer has finished.
- This ensures a first-come, first-served order, preventing a stream of readers from indefinitely starving a writer.

Ans-5) Practical Drawback: Severe Reduction in Resource utilization
The common method to eliminate "Hold and wait" is to require a process to request all needed resources at once before it starts.

The drawback is that resources allocated so a process may sit idle for longer period. For example, a process that needs a scanner now and a plotter later must lock both at the start. The plotter is unavailable to other processes while the process uses the scanner, leading to ~~door~~ poor overall system performance and throughput.

Ans-6) a) The global graph is formed by connecting the local fragments.

- From S1: P1->P2 and P3->P4
- From S2: P2->P5 and P5->P6
- From S3: P6->P1

P1->P2 -> P5 -> P6->P1

b) Yes, a deadlock exists. The process involved in the circular wait are P1, P2, P5 and P6.

c) A suitable algorithm is the Path-Pushing Algorithm. In this method, a site that suspects a deadlock initiates a probe message. This probe travels along the edges of the wait-for graph. If the probe ever returns to the

Ans 7) a) Expected Time = (Probability Local × Time Local) + (Probability × Time Remote)

$= (0.7 \times 5ms) + (0.3 \times 25ms) = 3.5 + 7.5 = 10ms.$

b) Strategy: Client-side caching

Justification: After a file is accessed remotely once, subsequent reads are served from the local cache. This reduces the effective probability of a remote access, dramatically lowering the average access time from 25ms to 5ms for cached files

classmate
Date
Page

Ans-9) a) Total overhead : 200ms + (9 x 50 ms) = 650ms

b) The 1-second RPO requires a checkpoint every second using one full checkpoint as a base and nine faster incrementals is the cheapest valid way to create this continuous recovery chain

Ans-9) a) 1 full + 9 incremental checkpoints
Overhead = 200 + (9 x 50) = 650ms.

b) • 1-second RPO requires a checkpoint every second
• 1 full checkpoint creates a valid recovery base
• 9 incremental checkpoints maintain the RPO at minimum cost
• This is the cheapest valid strategy meeting the RPO requirement.