Date

Assignment - 03

OPERATING SYSTEM

1. A race Condition occurs when two or more entities try to change a shared resource simutaneously, leading to unpredictable results.
eg: Two people editing the same document at once - one saves changes while the other one write

2.

Aspect	Peterson's Solⁿ	Semaphores
Implentation	Software based algorithm for two process	Abract data types implemented in OS.
Complexity	Simple logic but limited to 2 process	More flexible, Supports multiple process.
Hardware dependency	Works purely in software	Depends on hardware Supports atomic operations

3. Advantage : Monitors provide automactic Synchronization through mutual exclusion within the mointer. In multicore systems, they are easier to implement and maintain as synchronization is handled at a higher ~~to~~ level, reducing the chance of programming errors.

Date

4. Starvation: occurs when writers keep waiting indefinitely because continuous readers hold access to shared data.

Preventation: Use write, priority once a write is waiting, blocks new readers until the writers finishes.

5. Drawback: Process must request all resources at once before execution begins, leading to resources underutilization and reduced concurency since some resources remain idle for long periods.

6. Given:
Total instances: A=10, B=5, C=7
Allocation & Max Table;

Process	Allocation (A, B, C)	Max (A, B, C)
P0	0, 1, 0	7, 5, 3
P1	2, 0, 0	3, 2, 2
P2	3, 0 2	9, 0, 2
P3	2, 1, 1	4, 2, 2
P4	0, 0, 2	5, 3, 3

(a) Need Matrix = Max − allocation

Process	Need (A, B, C)
P0	(7−0, 5−1, 3−0)
P1	(3−2, 2−0, 2−0)
P2	(9−3, 0−0, 2−0)

Date

P-3	(4-2 , 2-1 , 2-1)	
P4	(5-0 , 3-0 , 3-2)	

(b) Available = Total − Σ Allocation
(10, 7, 5) − (7, 2, 5) = (3, 3, 2)

Now check Safe Sequence using Banker's algorithm.
Safe sequency: P1 → P3 → P4 → P0 → P2

(c) If P1 requests (1, 0, 2);
New need for P1 = (1, 2, 2) − (1, 0, 2) = (0, 2, 0)

Available = (3, 3, 2) − (1, 0, 2) = (2, 3, 0)
check if safe → sequence still possible → Yes.

7. Dinning Philosophers Problem: using semaphores.

- Each Philosopher has once chopstick
- To each, philosopher need both left & right chopstick.

Deadlock Scenario: All philosopher pick-up their left chopstick & wait for the right one

Soln: Use one Semaphore mutex to limit maximum philosopheres eating to 4 (n-1) rule.

Date

8. I/O System Analysis:

Given:

- Interupt handling time: 5 μs
- Data transfer rate: 500 KB/s
  : 500,000 bytes/s
- Data block per interupt = 100 bytes

(a) CPU time spent handling interupts:

interupts per second = 500,000 / 100 = 5000.

CPU time = 5000 x 5 μs = 25000 μs = 0.025 s

2.5% of CPU time per second.

(b) Improvement:

Increase data block size per interupt

eg: 1KB instead of 100 bytes

9. Case Study - Air traffic control system

(a) Critical Sections:

- Radar data acquition:
- Flight Path Calculation
- Communication Channel updates

IPC mechanism: Use message queue for real time data synchronization and minimal latency.

(b) Deadlock Handling:

- If a deadlocks occurs. b/w data acquition / Path Calculation