

Name - Virshat Yadav

Roll No - 2301010258

### Assignment - 03

#### Operating system

Ans 1

A race condition occurs when two or more entities try to change a shared resource simultaneously leading to unpredictable results

eg - (real world) - Two people editing the same document at once - one saves changes while the other overwrites them

Ans 2

Aspect	Peterson's Sol <sup>n</sup>	Semaphores
Implementation	Software based algorithm for two processes	Abstract data types implemented in as
complexity	Simpler logic but led to 2 process	More flexible, supports multiple process
Hardware dependancy	works purely in software	Depends on hardware supported atomic operations

Ans 3 Advantage -

monitors provide automatic synchronization through mutual exclusion. Within the monitor, in multi core system, they are easier to implement & maintain as synchronization is handled at a higher level, reducing the chance of programming errors.

Ans 4

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

Prevention - use write priority - once a writer is waiting, block new readers until the writer finishes.

Ans 5

Drawback - Process must request all resources at once before execution begins, leading to resources underutilization & reduced concurrency since some resources remain idle for long periods.

## Case Study - Air Traffic Control System

### a Critical sections

- Radar data acquisition
- Flight path calculation
- Communication Channel updates

IPC mechanism: use message queue for real-time data synchronization and minimal latency

### b Deadlock handling

If a deadlock occurs b/w data acquisition and path calculation.

## Ans Dining Philosophers problem

using semaphores

- Each philosopher has one chopstick
- To each, philosopher needs both left & right chopstick

Deadlock scenario - All philosophers pick up their left chopstick & wait for the right one.

Soln - use one semaphore mutex to limit maximum philosophers eating to  $(n-1)$  units.

Ans. Available = Total -  $\Sigma$  Allocation

$$(10, 5, 7) - (7, 2, 5) = (3, 3, 2)$$

Now check safe sequence using Banker's algorithm

Safe sequence =  $P_1 \rightarrow P_3 \rightarrow P_4 \rightarrow P_0 \rightarrow P_2$

c) If  $P_1$  requests (1, 0, 2)  
 New need for  $P_1 = (1, 2, 2) - (1, 0, 2) = (0, 2, 0)$   
 Available  $(3, 3, 2) - (1, 0, 2) = (2, 3, 0)$

check if safe  $\rightarrow$  Sequence still possible — Yes

Ans Given

Total Instances  $A=10, B=5, C=7$

Allocation & Max table

a) Need Matrix = Max - Allocation

Process	Need (A, B, C)
$P_0$	$(7-0, 5-1, 3-0)$
$P_1$	$(3-2, 2-0, 2-0)$
$P_2$	$(9-3, 0-0, 2-0)$
$P_3$	$(4-2, 2-1, 2-1)$
$P_4$	$(5-0, 3-0, 3-2)$

b) Available = Total -  $\Sigma$  Allocation

$$(10, 5, 7) - (7, 2, 5) = (3, 3, 2)$$