Operating System

Process Synchronization Part - 4

DPP-04

[MCQ]

1. A counting semaphore is initialized to 12. Then 6P (wait) operations and 4V (Signal) operation were completed on this semaphore. The resulting value of the semaphore is?

(a) 10 (b) 11
(c) 12 (d) 13

[MCQ]

2. A deadlock free solution to the dining philosopher's problem

(a) eliminates any possibility of any other type of problem.
(b) possibility of starvation is eliminated.
(c) does not necessarily eliminates the possibility of occurrence of starvation.
(d) none of the above.

[MCQ]

3. The term "Busy waiting" used in process synchronization when a process executes critical region and other processes are waiting for their turn to enter into critical region.
Busy waiting is also known as ____.

(a) Busy processing
(b) Spin lock
(c) Bounded waiting
(d) none

[NAT]

4. Consider the two function P_i and P_j that share a variable Q with an initial value '3' execute concurrently:

P_i()	P_j()

{
 R = Q * 2;
 Q = R;
 }
	{
	 S = Q + 1;
	 Q = S
	 }

What are the possible different value for variable Q at the end of execution of both process P_i and P_j?

[MCQ]

5. Assume a counting semaphore variable 'P', following are the semaphore operations performed 20P, 5V, 9V, 4P, 8V. What will be the largest initial value of 'P' to keep 1 process in suspended list?

(a) 1 (b) 2
(c) 3 (d) none

[MCQ]

6. The two atomic operations allowed on semaphores are _______ and _______.

(a) hold, signal (b) wait, hold
(c) wait, stop (d) wait signal

[MCQ]

7. Which of the following is not an optimization benchmarks in the design of a CPU scheduling policy?

(a) Minimum waiting time
(b) Maximum throughput
(c) Minimum CPU utilization
(d) Minimum turnaround time

Answer Key

1. **(a)**
2. **(c)**
3. **(b)**
4. **(4)**
5. **(a)**
6. **(d)**
7. **(c)**

Hints and Solutions

1. (a)

6P ⇒ decrements the semaphore 6 times hence the value becomes 6.

4P ⇒ increments the semaphore 4 times hence the value becomes 10.

Note: The positive value of counting semaphore means that those many down(P) operations can be done successfully . The negative value of counting semaphore indicates that the number of blocked processes.

2. (c)

Does not necessarily eliminate the possibility of occurrence of starvation.

3. (b)

Spinlock is also called as Busy waiting. When a process executes critical region and other processes are waiting for their turn to execute or enter into critical region, then it is known as spinlock.

4. (4)

I. It is given that process $P_i()$ and process $P_j()$ is executing concurrently. So, assign the unique number to operations of both the processes.

$P_i()$: $I_1 \Rightarrow R = Q * 2$; $I_2 \Rightarrow Q = R$

$P_j()$: $I_3 \Rightarrow S = Q + 1$; $I_4 \Rightarrow Q = S$

II. Now, perform the operation to find the distinct values of Q

1. $I_1, I_2, I_3, I_4 = 7$
2. $I_1, I_3, I_2, I_4 = 4$
3. $I_3, I_1, I_4, I_2 = 6$
4. $I_3, I_4, I_1, I_2 = 8$
5. $I_3, I_1, I_2, I_4 = 4$
6. $I_1, I_3, I_4, I_2 = 6$

Therefore, we have total 4 distinct values of Q: {7, 4, 6, 8}

5. (a)

Binary semaphore: the integer value in this type of semaphore can take values only 0's and 1's.

Counting semaphore: The range of integer value in this type of semaphore can be more than 0's and 1's.

There are two types of operations that are atomic and mutually exclusive in nature.

(1) wait(S) (or) P

(2) signal(S) (or)V

Calculations

Let us assume initial value of counting semaphore to be 'a'.

One process is in suspended list, which makes the final value to be –1.

$-1 = a - 20 + 5 + 9 - 4 + 8$

$-1 = a - 24 + 22$

$a = 1.$

6. (d)

- **Wait:** wait operation decrements the value of its argument 'S' if it is positive. If 'S' is negative or zero, then no operation is done or performed.
- **Signal:** Signal operation increments the value of its argument 'S'.
- Semaphores are integer variables which is used to solve critical section problems by using two atomic operations mentioned above.

7. (c)

Maximum throughput, maximum CPU utilization, minimum turnaround time and minimum waiting time are optimization benchmarks in the design of a CPU scheduling policy.