Branch : CSE & IT Batch:Hinglish

Subject : Operating System

Topic : Deadlock

DPP-02

[MCQ]

1. Which of the scheme is false regarding deadlock prevention scheme?

(a) request all the necesssary resources that has to be allocated before execution.

(b) never request a resource after releasing resource.

(c) release all resources before requesting a new resource.

(d) none

[MCQ]

2. Consider the following statements:

S_1: In detection and recovery. The CPU periodically checks for deadlock.

S_2: In deadlock ignorance, operating system simply ignores the deadlock.

Which of the following is correct?

(a) S_1 is true S_2 is false.

(b) S_2 is true and S_1 is false.

(c) S_1 and S_2 are true.

(d) Both S_1 and S_2 are false.

[MSQ]

3. Choose the correct statements from the following.

(a) If each resource type has several instances, then a cycle does not necessarily imply that a deadlock has occurred.

(b) We can allow system to enter a deadlock state, detect it and recover.

(c) A cycle in the graph, does not necessarily imply that a deadlock has occurred.

(d) none

[MCQ]

4. Choose the correct statements from the following statements

(a) If a system does not employ either a deadlock prevention (or) a deadlock avoidance algorithm, then a deadlock situation will always occur.

(b) The wait for graph scheme is applicable to resource allocation system with multiple instances of each resource type to detect a cycle in a graph.

(c) To eliminate deadlock using resource preemption, we successively preempt some resources from processes and give these resources to other processes unitl the deadlock is broken.

(d) none

[MCQ]

5. A machine has 6 identical resources and N processes competing for them. Each process can request at most 2 resources. Which one of the following values of N could lead to deadlock?

(a) 4 (b) 5

(c) 6 (d) none

[MCQ]

6. Mutual exclusion condition must hold for

(a) Sharable resources

(b) non-sharable resources

(c) Both (a) and (b)

(d) none

[MCQ]

7. Consider the following statements

S_1: To ensure that the hold and wait condition never occurs in the system, we must guarantee that, whenever a process requests a resource. It does not hold any other resource.

S_2: A deadlock avoidance algorithm dynamically examines the resource allocation state to ensure that a circular wait condition can never exist.

(a) only S_1 is true

(b) only S_2 is correct

(c) both S_1 and S_2 are correct

(d) none

Answer Key

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

Hints and Solutions

1. (b)

(Conceptual) process always can request a resource after releasing resources.

Deadlock avoidance: In deadlock avoidance we try to avoid deadlock from the system the system always makes sure that it is in safe state.

If a system is going into the unsafe state by some resource request. Then system avoids that kind of request and ignores those requests that will put system into unsafe state.

2. (c)

- In deadlock recovery, the CPU does periodically check for the deadlock. The RAG is used to detect the deadlock.
- In deadlock ignorance, as process of deadlock preventions are costly, the OS ignores by assuming that deadlock is never going to happen.

3. (a, b, c)

- We can allow system to enter a deadlock state and detect it through many techniques such as RAG.
- A cycle in the graph does not necessarily implies that deadlock has occurred because, a cycle in graph is a necessary but not sufficient condition for the existence of deadlock.

4. (c)

- If a system does not employ either a deadlock prevention (or) a deadlock avoidance algorithm, then a deadlock situation "may" occur.
- The wait for graph scheme is not applicable to resource allocation system with instances of each resource.

5. (c)

Available identical resources R = 6

Max needs per process = 2

Deadlock will occur if any process gets available resources < needed(requested)

Maximum recourse per process that will lead to deadlock = needed – 1 = 2 – 1 = 1.

For N processes max recourse to be in deadlock = N * 1 = N condition for deadlock

$N \geq R$

$N \geq 6$

6 is the value of N that could lead to deadlock state.

6. (b)

Mutual exclusion condition must hold for non-sharable resources.

Eg: a printer cannot be used/shared simultaneously by several process.

7. (c)

- To ensure that the hold and wait condition never occurs in the system, we must guarantee that whenever a process requests a resource it does not hold any other resources.
- A deadlock avoidance algorithm, dynamically examines that resource allocation state to ensure that a circular wait condition can never exists.