

1. Deadlock may occur when:

A wait() is followed by a wait() instead of a signal()

A programmer reverses the order of wait() and signal()

2. Select all the necessary and sufficient conditions to cause deadlock:

- **Circular wait (A process must be waiting for a resource which is being held by another process, which in turn is waiting for the first process to release a resource)**
- **Hold-and-wait (A process currently holding at least 1 resource is requesting additional resources held by other processes)**
- **No Preemption (A resource can be released only voluntarily by the process holding it)**
- **Mutual Exclusion (A resource is held in a non-shareable mode)**

3. Match the following terms:

Deadlock Avoidance:

For every request, the system determines if it would be in an unsafe state by granting that request. And thereby, will not grant it.

Deadlock Detection:

Deadlocks are allowed to occur and system will examine itself periodically to detect detection and correct it.

Deadlock Prevention:

Preventing one of the four necessary and sufficient conditions from occurring.

4. Match the word with the definition:

Mutual Exclusion:

The idea of which we want to prevent one process from accessing a resource while the current process is modifying said shared resource.

Race Condition:

When two or more threads try to change the shared data at the same time.

Critical Section:

Area in code where processes access shared resources.

5. Dining philosophers problem is a classic synchronization problem. What problem are we specifically concerned about?

Deadlock

6. What problem can arise when implementing synchronization with Semaphores?

Deadlock

7. A binary semaphore provides the same functionality as a mutex.

False

8. A system in an unsafe state is one in deadlock.

False

9. A mutex would be preferred over a semaphore if the order of execution of critical sections for different processes was desired.

False

10. A _____ occurs when two or more threads try to access the same shared data and try to change them at the same time.

Race Condition