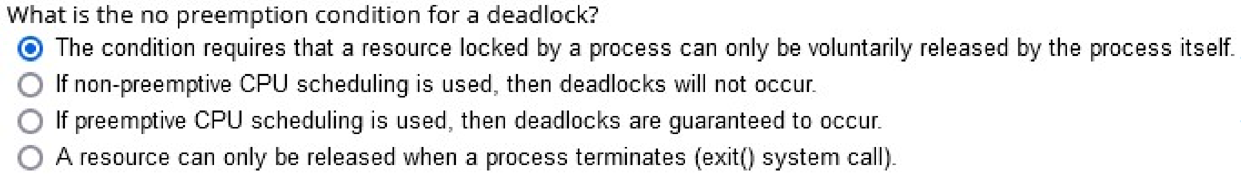
Quiz 04

Thursday, 22 September 2022 11:32

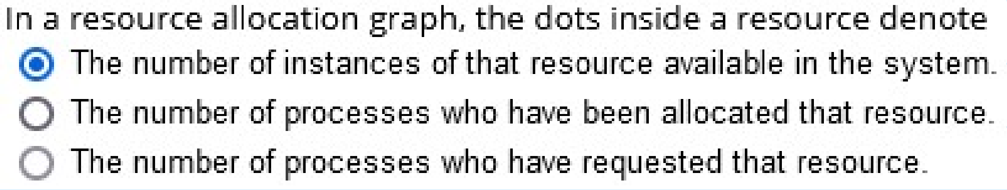
What is the no preemption condition for a deadlock?



What is the no preemption condition for a deadlock?

- ● The condition requires that a resource locked by a process can only be voluntarily released by the process itself.
- ○ If non-preemptive CPU scheduling is used, then deadlocks will not occur.
- ○ If preemptive CPU scheduling is used, then deadlocks are guaranteed to occur.
- ○ A resource can only be released when a process terminates (exit() system call).

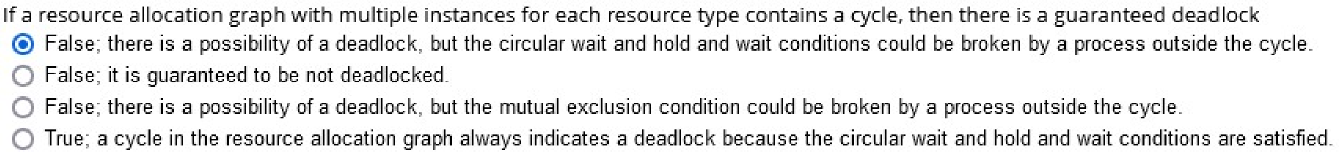
In a resource allocation graph, the dots inside a resource denote



In a resource allocation graph, the dots inside a resource denote

- ● The number of instances of that resource available in the system.
- ○ The number of processes who have been allocated that resource.
- ○ The number of processes who have requested that resource.

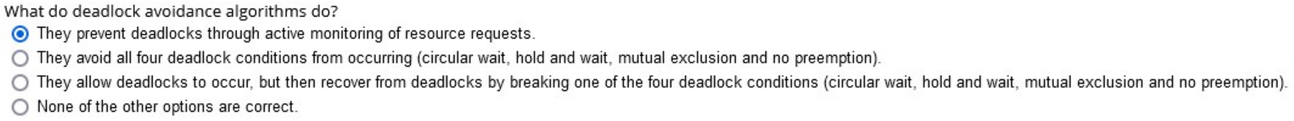
If a resource allocation graph with multiple instances for each resource type contains a cycle, then there is a guaranteed deadlock



If a resource allocation graph with multiple instances for each resource type contains a cycle, then there is a guaranteed deadlock

- ● False; there is a possibility of a deadlock, but the circular wait and hold and wait conditions could be broken by a process outside the cycle.
- ○ False; it is guaranteed to be not deadlocked.
- ○ False; there is a possibility of a deadlock, but the mutual exclusion condition could be broken by a process outside the cycle.
- ○ True; a cycle in the resource allocation graph always indicates a deadlock because the circular wait and hold and wait conditions are satisfied.

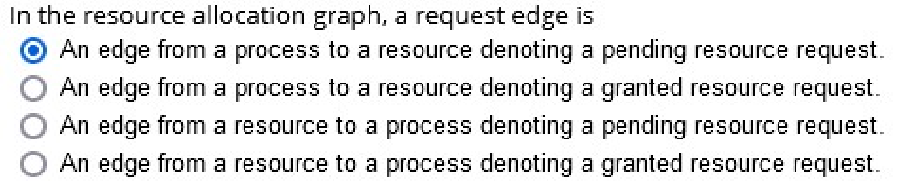
What do deadlock avoidance algorithms do?



What do deadlock avoidance algorithms do?

- ● They prevent deadlocks through active monitoring of resource requests.
- ○ They avoid all four deadlock conditions from occurring (circular wait, hold and wait, mutual exclusion and no preemption).
- ○ They allow deadlocks to occur, but then recover from deadlocks by breaking one of the four deadlock conditions (circular wait, hold and wait, mutual exclusion and no preemption).
- ○ None of the other options are correct.

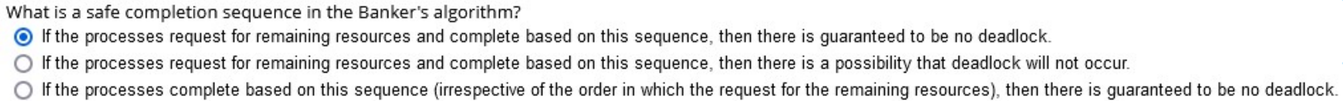
In the resource allocation graph, a request edge is



In the resource allocation graph, a request edge is

- ● An edge from a process to a resource denoting a pending resource request.
- ○ An edge from a process to a resource denoting a granted resource request.
- ○ An edge from a resource to a process denoting a pending resource request.
- ○ An edge from a resource to a process denoting a granted resource request.

What is a safe completion sequence in the Banker's algorithm?



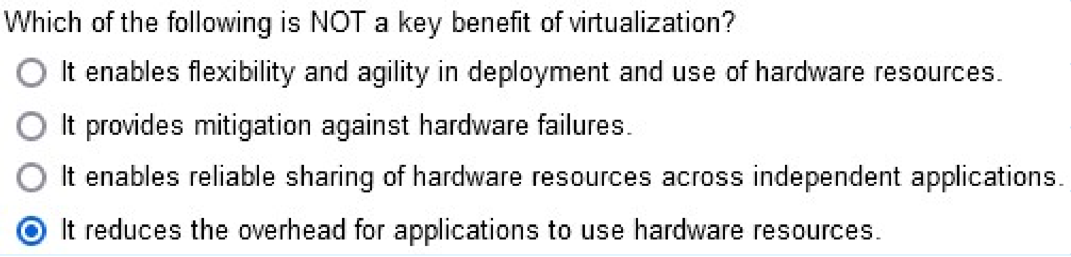
What is a safe completion sequence in the Banker's algorithm?

- ● If the processes request for remaining resources and complete based on this sequence, then there is guaranteed to be no deadlock.
- ○ If the processes request for remaining resources and complete based on this sequence, then there is a possibility that deadlock will not occur.
- ○ If the processes complete based on this sequence (irrespective of the order in which the request for the remaining resources), then there is guaranteed to be no deadlock.

Quiz 04

Thursday, 22 September 2022 11:32

Which of the following is NOT a key benefit of virtualization?



Which of the following is NOT a key benefit of virtualization?

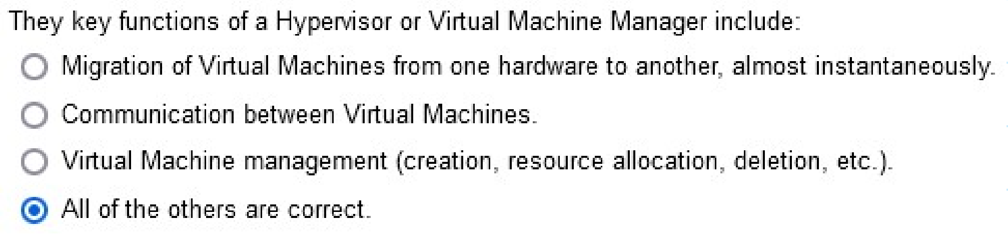
○ It enables flexibility and agility in deployment and use of hardware resources.

○ It provides mitigation against hardware failures.

○ It enables reliable sharing of hardware resources across independent applications.

● It reduces the overhead for applications to use hardware resources.

The key functions of a Hypervisor or Virtual Machine Manager include:



They key functions of a Hypervisor or Virtual Machine Manager include:

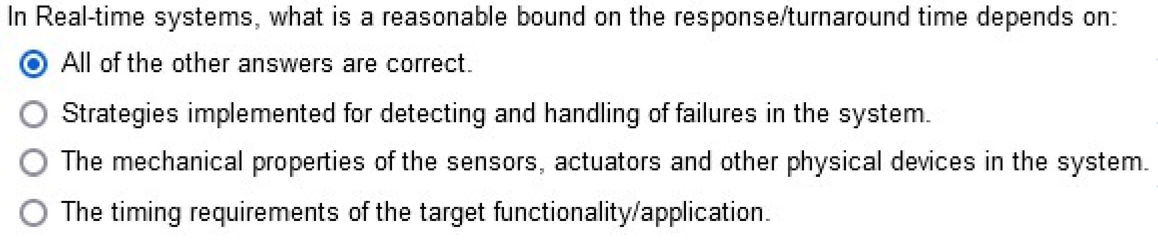
○ Migration of Virtual Machines from one hardware to another, almost instantaneously.

○ Communication between Virtual Machines.

○ Virtual Machine management (creation, resource allocation, deletion, etc.).

● All of the others are correct.

In Real-time systems, what is a reasonable bound on the response/turnaround time depends on:



In Real-time systems, what is a reasonable bound on the response/turnaround time depends on:

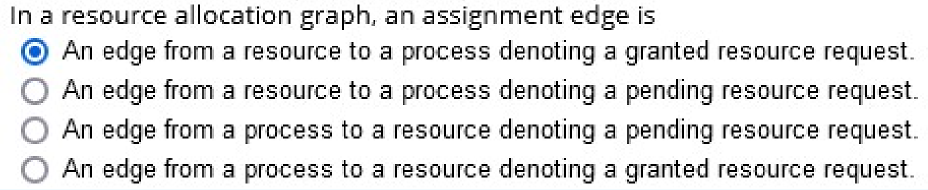
● All of the other answers are correct.

○ Strategies implemented for detecting and handling of failures in the system.

○ The mechanical properties of the sensors, actuators and other physical devices in the system.

○ The timing requirements of the target functionality/application.

In a resource allocation graph, an assignment edge is



In a resource allocation graph, an assignment edge is

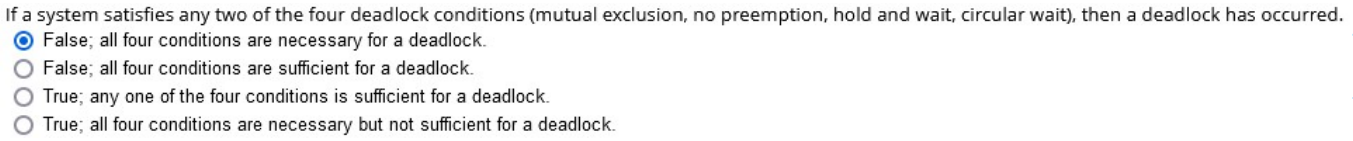
● An edge from a resource to a process denoting a granted resource request.

○ An edge from a resource to a process denoting a pending resource request.

○ An edge from a process to a resource denoting a pending resource request.

○ An edge from a process to a resource denoting a granted resource request.

If a system satisfies any two of the four deadlock conditions (mutual exclusion, no preemption, hold and wait, circular wait), then a deadlock has occurred.



If a system satisfies any two of the four deadlock conditions (mutual exclusion, no preemption, hold and wait, circular wait), then a deadlock has occurred.

● False; all four conditions are necessary for a deadlock.

○ False; all four conditions are sufficient for a deadlock.

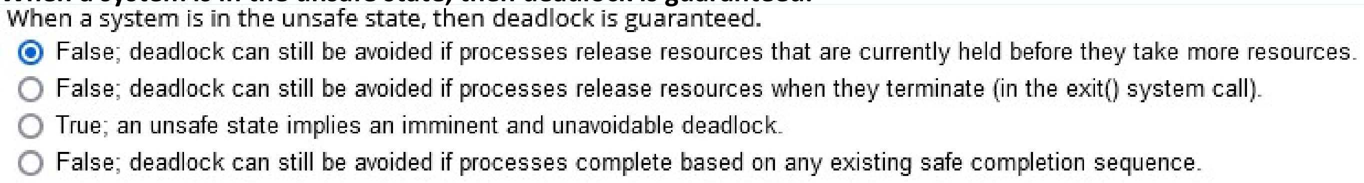
○ True; any one of the four conditions is sufficient for a deadlock.

○ True; all four conditions are necessary but not sufficient for a deadlock.

Quiz 04

Thursday, 22 September 2022 11:32

When a system is in the unsafe state, then deadlock is guaranteed.



When a system is in the unsafe state, then deadlock is guaranteed.

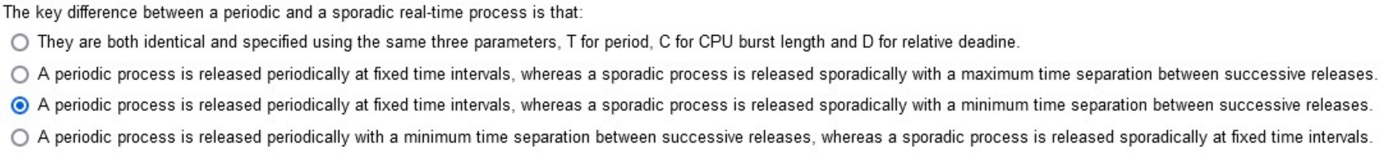
● False; deadlock can still be avoided if processes release resources that are currently held before they take more resources.

○ False; deadlock can still be avoided if processes release resources when they terminate (in the exit() system call).

○ True; an unsafe state implies an imminent and unavoidable deadlock.

○ False; deadlock can still be avoided if processes complete based on any existing safe completion sequence.

The key difference between a periodic and a sporadic real-time process is that:



The key difference between a periodic and a sporadic real-time process is that:

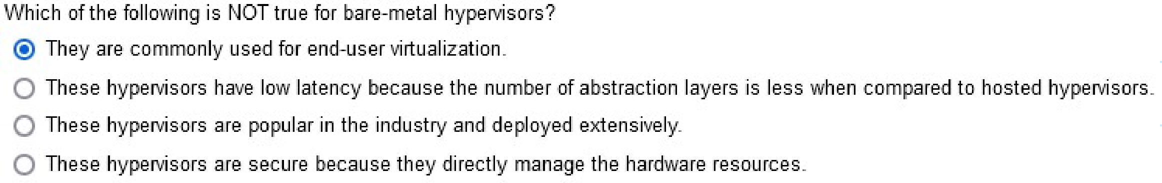
○ They are both identical and specified using the same three parameters, T for period, C for CPU burst length and D for relative deadine.

○ A periodic process is released periodically at fixed time intervals, whereas a sporadic process is released sporadically with a maximum time separation between successive releases.

● A periodic process is released periodically at fixed time intervals, whereas a sporadic process is released sporadically with a minimum time separation between successive releases.

○ A periodic process is released periodically with a minimum time separation between successive releases, whereas a sporadic process is released sporadically at fixed time intervals.

Which of the following is NOT true for bare-metal hypervisors?



Which of the following is NOT true for bare-metal hypervisors?

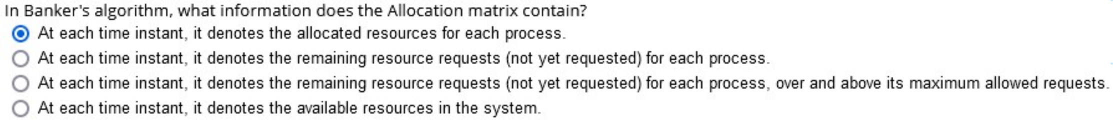
● They are commonly used for end-user virtualization.

○ These hypervisors have low latency because the number of abstraction layers is less when compared to hosted hypervisors.

○ These hypervisors are popular in the industry and deployed extensively.

○ These hypervisors are secure because they directly manage the hardware resources.

In Banker's algorithm, what information does the Allocation matrix contain?



In Banker's algorithm, what information does the Allocation matrix contain?

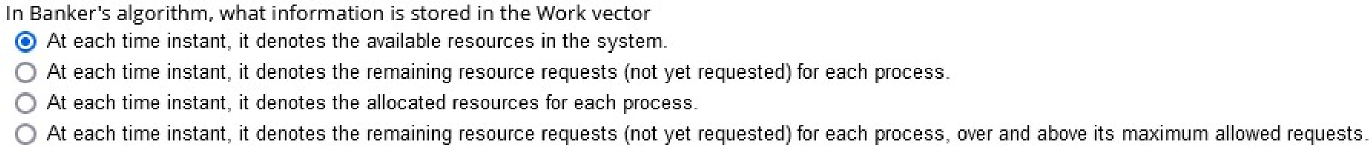
● At each time instant, it denotes the allocated resources for each process.

○ At each time instant, it denotes the remaining resource requests (not yet requested) for each process.

○ At each time instant, it denotes the remaining resource requests (not yet requested) for each process, over and above its maximum allowed requests.

○ At each time instant, it denotes the available resources in the system.

In Banker's algorithm, what information is stored in the Work vector



In Banker's algorithm, what information is stored in the Work vector

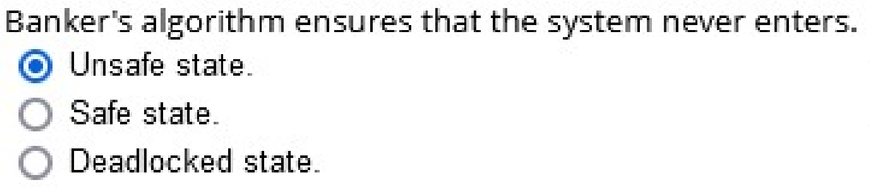
● At each time instant, it denotes the available resources in the system.

○ At each time instant, it denotes the remaining resource requests (not yet requested) for each process.

○ At each time instant, it denotes the allocated resources for each process.

○ At each time instant, it denotes the remaining resource requests (not yet requested) for each process, over and above its maximum allowed requests.

Banker's algorithm ensures that the system never enters.



Banker's algorithm ensures that the system never enters.

● Unsafe state.

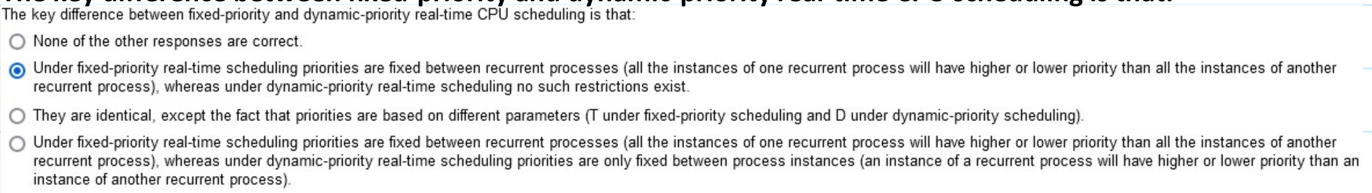
○ Safe state.

○ Deadlocked state.

Quiz 04

Thursday, 22 September 2022 11:32

The key difference between fixed-priority and dynamic-priority real-time CPU scheduling is that:



The key difference between fixed-priority and dynamic-priority real-time CPU scheduling is that:

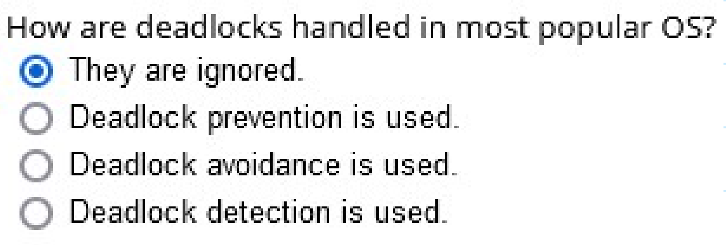
○ None of the other responses are correct.

● Under fixed-priority real-time scheduling priorities are fixed between recurrent processes (all the instances of one recurrent process will have higher or lower priority than all the instances of another recurrent process), whereas under dynamic-priority real-time scheduling no such restrictions exist.

○ They are identical, except the fact that priorities are based on different parameters (T under fixed-priority scheduling and D under dynamic-priority scheduling).

○ Under fixed-priority real-time scheduling priorities are fixed between recurrent processes (all the instances of one recurrent process will have higher or lower priority than all the instances of another recurrent process), whereas under dynamic-priority real-time scheduling priorities are only fixed between process instances (an instance of a recurrent process will have higher or lower priority than an instance of another recurrent process).

How are deadlocks handled in most popular OS?



How are deadlocks handled in most popular OS?

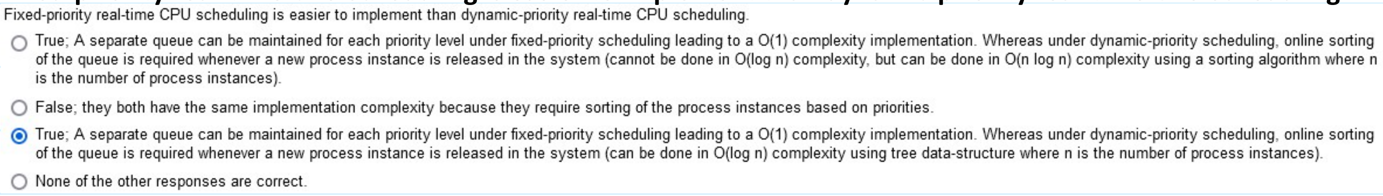
● They are ignored.

○ Deadlock prevention is used.

○ Deadlock avoidance is used.

○ Deadlock detection is used.

Fixed-priority real-time CPU scheduling is easier to implement than dynamic-priority real-time CPU scheduling.



Fixed-priority real-time CPU scheduling is easier to implement than dynamic-priority real-time CPU scheduling.

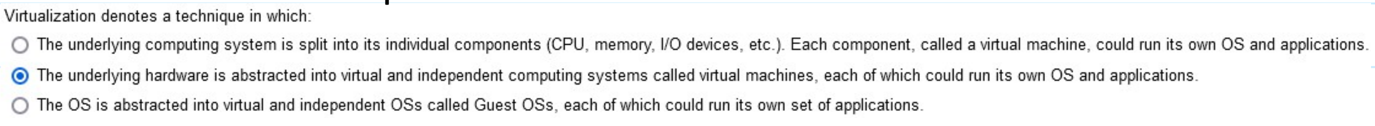
○ True; A separate queue can be maintained for each priority level under fixed-priority scheduling leading to a O(1) complexity implementation. Whereas under dynamic-priority scheduling, online sorting of the queue is required whenever a new process instance is released in the system (cannot be done in O(log n) complexity, but can be done in O(n log n) complexity using a sorting algorithm where n is the number of process instances).

○ False; they both have the same implementation complexity because they require sorting of the process instances based on priorities.

● True; A separate queue can be maintained for each priority level under fixed-priority scheduling leading to a O(1) complexity implementation. Whereas under dynamic-priority scheduling, online sorting of the queue is required whenever a new process instance is released in the system (can be done in O(log n) complexity using tree data-structure where n is the number of process instances).

○ None of the other responses are correct.

Virtualization denotes a technique in which:



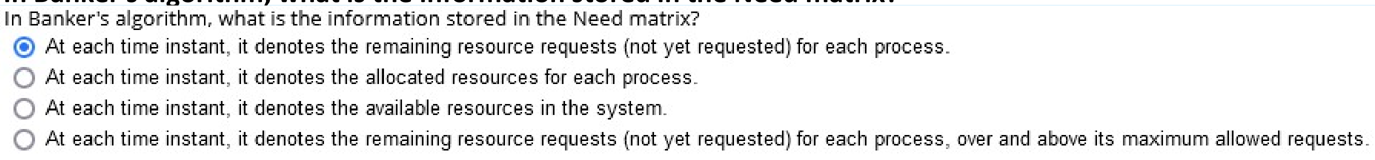
Virtualization denotes a technique in which:

○ The underlying computing system is split into its individual components (CPU, memory, I/O devices, etc.). Each component, called a virtual machine, could run its own OS and applications.

● The underlying hardware is abstracted into virtual and independent computing systems called virtual machines, each of which could run its own OS and applications.

○ The OS is abstracted into virtual and independent OSs called Guest OSs, each of which could run its own set of applications.

In Banker's algorithm, what is the information stored in the Need matrix?



In Banker's algorithm, what is the information stored in the Need matrix?

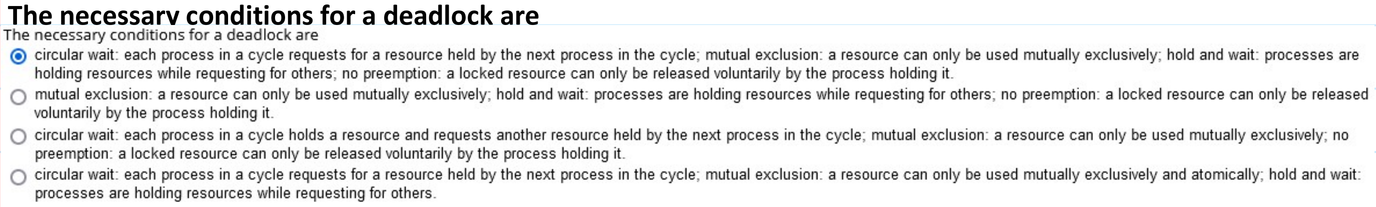
● At each time instant, it denotes the remaining resource requests (not yet requested) for each process.

○ At each time instant, it denotes the allocated resources for each process.

○ At each time instant, it denotes the available resources in the system.

○ At each time instant, it denotes the remaining resource requests (not yet requested) for each process, over and above its maximum allowed requests.

The necessarv conditions for a deadlock are



The necessary conditions for a deadlock are

● circular wait: each process in a cycle requests for a resource held by the next process in the cycle; mutual exclusion: a resource can only be used mutually exclusively; hold and wait: processes are holding resources while requesting for others; no preemption: a locked resource can only be released voluntarily by the process holding it.

○ mutual exclusion: a resource can only be used mutually exclusively; hold and wait: processes are holding resources while requesting for others; no preemption: a locked resource can only be released voluntarily by the process holding it.

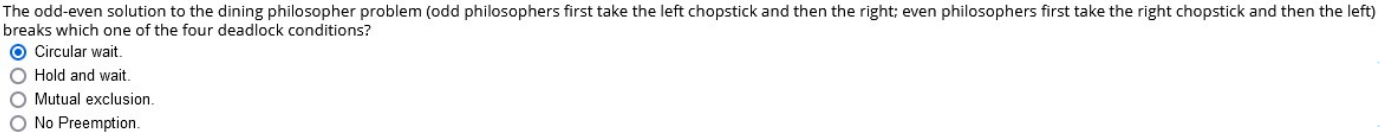
○ circular wait: each process in a cycle holds a resource and requests another resource held by the next process in the cycle; mutual exclusion: a resource can only be used mutually exclusively; no preemption: a locked resource can only be released voluntarily by the process holding it.

○ circular wait: each process in a cycle requests for a resource held by the next process in the cycle; mutual exclusion: a resource can only be used mutually exclusively and atomically; hold and wait: processes are holding resources while requesting for others.

Quiz 04

Thursday, 22 September 2022 11:32

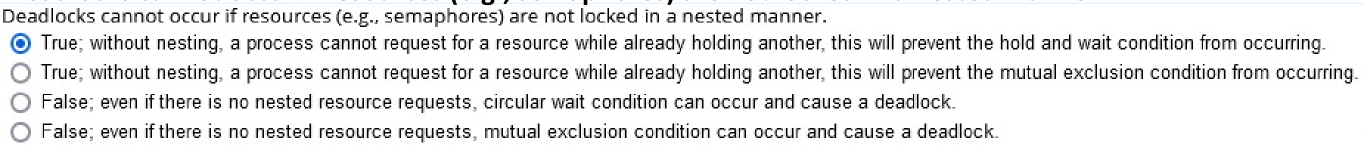
The odd-even solution to the dining philosopher problem (odd philosophers first take the left chopstick and then the right; even philosophers first take the right chopstick and then the left) breaks which one of the four deadlock conditions?



The odd-even solution to the dining philosopher problem (odd philosophers first take the left chopstick and then the right; even philosophers first take the right chopstick and then the left) breaks which one of the four deadlock conditions?

- ● Circular wait.
- ○ Hold and wait.
- ○ Mutual exclusion.
- ○ No Preemption.

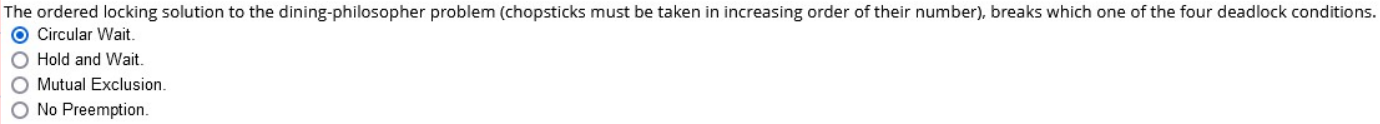
Deadlocks cannot occur if resources (e.g., semaphores) are not locked in a nested manner.



Deadlocks cannot occur if resources (e.g., semaphores) are not locked in a nested manner.

- ● True; without nesting, a process cannot request for a resource while already holding another, this will prevent the hold and wait condition from occurring.
- ○ True; without nesting, a process cannot request for a resource while already holding another, this will prevent the mutual exclusion condition from occurring.
- ○ False; even if there is no nested resource requests, circular wait condition can occur and cause a deadlock.
- ○ False; even if there is no nested resource requests, mutual exclusion condition can occur and cause a deadlock.

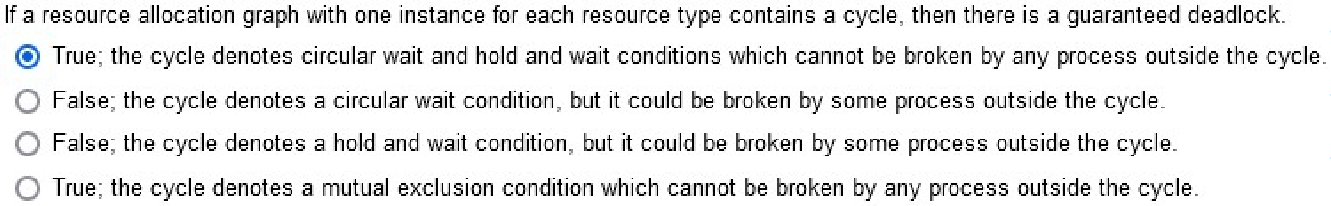
The ordered locking solution to the dining-philosopher problem (chopsticks must be taken in increasing order of their number), breaks which one of the four deadlock conditions.



The ordered locking solution to the dining-philosopher problem (chopsticks must be taken in increasing order of their number), breaks which one of the four deadlock conditions.

- ● Circular Wait.
- ○ Hold and Wait.
- ○ Mutual Exclusion.
- ○ No Preemption.

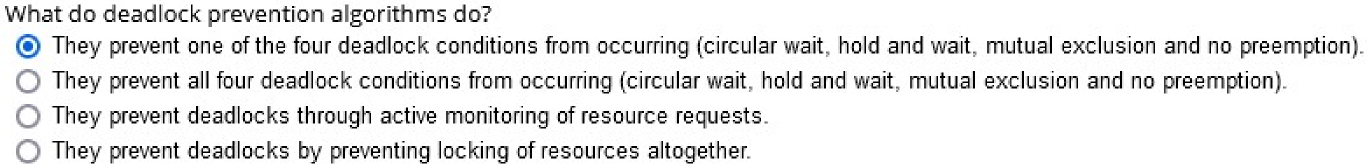
If a resource allocation graph with one instance for each resource type contains a cycle, then there is a guaranteed deadlock.



If a resource allocation graph with one instance for each resource type contains a cycle, then there is a guaranteed deadlock.

- ● True; the cycle denotes circular wait and hold and wait conditions which cannot be broken by any process outside the cycle.
- ○ False; the cycle denotes a circular wait condition, but it could be broken by some process outside the cycle.
- ○ False; the cycle denotes a hold and wait condition, but it could be broken by some process outside the cycle.
- ○ True; the cycle denotes a mutual exclusion condition which cannot be broken by any process outside the cycle.

What do deadlock prevention algorithms do?



What do deadlock prevention algorithms do?

- ● They prevent one of the four deadlock conditions from occurring (circular wait, hold and wait, mutual exclusion and no preemption).
- ○ They prevent all four deadlock conditions from occurring (circular wait, hold and wait, mutual exclusion and no preemption).
- ○ They prevent deadlocks through active monitoring of resource requests.
- ○ They prevent deadlocks by preventing locking of resources altogether.