QUESTION 1

1. Banker's algorithm ensures that the system never enters.

- (•) Unsafe state.
- ( ) Safe state.
- ( ) Deadlocked state.

0.15 points

QUESTION 2

1. 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.

0.15 points

QUESTION 3

1. 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.

0.15 points

QUESTION 4

1. 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.

0.15 points

QUESTION 5

1. What is the no preemption condition for a deadlock?

- (selected) 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).

0.15 points

QUESTION 6

1. In the resource allocation graph, a request edge is

- (selected) 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.

0.15 points

QUESTION 7

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

- (selected) 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.

0.15 points

QUESTION 8

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

- 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).
- (selected) 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.
- None of the other responses are correct.

0.15 points

QUESTION 9

1. They key functions of a Hypervisor or Virtual Machine Manager include:
   - ○ Migration of Virtual Machines from one hardware to another, almost instantaneously.
   - ◉ All of the others are correct.
   - ○ Virtual Machine management (creation, resource allocation, deletion, etc.).
   - ○ Communication between Virtual Machines.

0.15 points

QUESTION 10

1. In Real-time systems, what is a reasonable bound on the response/turnaround time depends on:
   - ◉ All of the other answers are correct.
   - ○ The mechanical properties of the sensors, actuators and other physical devices in the system.
   - ○ The timing requirements of the target functionality/application.
   - ○ Strategies implemented for detecting and handling of failures in the system.

QUESTION 5

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

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

QUESTION 6

1. 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.

QUESTION 7

1. 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.

QUESTION 1

1. How are deadlocks handled in most popular OS?

- (•) They are ignored.
- ( ) Deadlock prevention is used.
- ( ) Deadlock avoidance is used.
- ( ) Deadlock detection is used.

0.15 points

QUESTION 2

1. 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.

QUESTION 5

1. 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.

0.15 points

QUESTION 6

1. 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.

0.15 points

QUESTION 7

1. 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.

0.15 points

QUESTION 8

1. Which of the following is NOT true for bare-metal hypervisors?
   - 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.
   - They are commonly used for end-user virtualization.