Operating System - Sample Questions

Multiple Choice	
1.	In what way is an operating system like a government? A) It seldom functions correctly. B) It creates an environment within which other programs can do useful work. C) It performs most useful functions by itself. D) It is always concerned primarily with the individual's needs.
2.	The most common secondary storage device is ____. A) random access memory B) solid-state disks C) tape drives D) magnetic disk
3.	A ____ can be used to prevent a user program from never returning control to the operating system. A) portal B) program counter C) firewall D) timer
4.	Embedded computers typically run on a ____ operating system. A) real-time B) Windows XP C) network D) clustered
5.	What are some other terms for kernel mode? A) supervisor mode B) system mode C) privileged mode D) All of the above
6.	A(n) _____ is the unit of work in a system. A) process B) operating system C) timer D) mode bit
7.	The two separate modes of operating in a system are A) supervisor mode and system mode B) kernel mode and privileged mode C) physical mode and logical mode D) user mode and kernel mode
8.	The ____ of a process contains temporary data such as function parameters, return addresses, and local variables.

	A) text section B) data section C) program counter D) stack
9.	A process control block ____. A) includes information on the process's state B) stores the address of the next instruction to be processed by a different process C) determines which process is to be executed next D) is an example of a process queue
10.	When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process? A) The child process runs concurrently with the parent. B) The child process has a new program loaded into it. C) The child is a duplicate of the parent. D) All of the above
11.	A _____ saves the state of the currently running process and restores the state of the next process to run. A) save-and-restore B) state switch C) context switch D) none of the above
12.	Which of the following statements is true? A) Shared memory is typically faster than message passing. B) Message passing is typically faster than shared memory. C) Message passing is most useful for exchanging large amounts of data. D) Shared memory is far more common in operating systems than message passing.
13.	A race condition ____. A) results when several threads try to access the same data concurrently B) results when several threads try to access and modify the same data concurrently C) will result only if the outcome of execution does not depend on the order in which instructions are executed D) None of the above
14.	A counting semaphore ____. A) is essentially an integer variable B) is accessed through only one standard operation C) can be modified simultaneously by multiple threads D) cannot be used to control access to a thread's critical sections
15.	A mutex lock ____. A) is exactly like a counting semaphore B) is essentially a boolean variable C) is not guaranteed to be atomic D) can be used to eliminate busy waiting

16.	In Peterson's solution, the ____ variable indicates if a process is ready to enter its critical section. A) `turn` B) `lock` C) `flag[i]` D) `turn[i]`
17.	The first readers-writers problem ____. A) requires that, once a writer is ready, that the writer performs its write as soon as possible. B) is not used to test synchronization primitives. C) requires that no reader will be kept waiting unless a writer has already obtained permission to use the shared database. D) requires that no reader will be kept waiting unless a reader has already obtained permission to use the shared database.
18.	What is the correct order of operations for protecting a critical section using mutex locks? A) `release()` followed by `acquire()` B) `acquire()` followed by `release()` C) `wait()` followed by `signal()` D) `signal()` followed by `wait()`
19.	What is the correct order of operations for protecting a critical section using a binary semaphore? A) `release()` followed by `acquire()` B) `acquire()` followed by `release()` C) `wait()` followed by `signal()` D) `signal()` followed by `wait()`
20.	A(n) _____ refers to where a process is accessing/updating shared data. A) critical section B) entry section C) mutex D) test-and-set
21.	Assume an adaptive mutex is used for accessing shared data on a Solaris system with multiprocessing capabilities. Which of the following statements is not true? A) A waiting thread may spin while waiting for the lock to become available. B) A waiting thread may sleep while waiting for the lock to become available. C) The adaptive mutex is only used to protect short segments of code. D) Condition variables and semaphores are never used in place of an adaptive mutex.
22.	What is the purpose of the mutex semaphore in the implementation of the bounded-buffer problem using semaphores? A) It indicates the number of empty slots in the buffer. B) It indicates the number of occupied slots in the buffer. C) It controls access to the shared buffer. D) It ensures mutual exclusion.

23.	How many philosophers may eat simultaneously in the Dining Philosophers problem with 5 philosophers? A) 1 B) 2 C) 3 D) 5
24.	Which of the following statements is true? A) A counting semaphore can never be used as a binary semaphore. B) A binary semaphore can never be used as a counting semaphore. C) Spinlocks can be used to prevent busy waiting in the implementation of the semaphore. D) Counting semaphores can be used to control access to a resource with a finite number of instances.
25.	_____ is/are not a technique for managing critical sections in operating systems. A) Peterson's solution B) Preemptive kernel C) Nonpreemptive kernel D) Semaphores
26.	When using semaphores, a process invokes the `wait()` operation before accessing its critical section, followed by the `signal()` operation upon completion of its critical section. Consider reversing the order of these two operations—first calling `signal()`, then calling `wait()`. What would be a possible outcome of this? A) Starvation is possible. B) Several processes could be active in their critical sections at the same time. C) Mutual exclusion is still assured. D) Deadlock is possible.
True/False	
27.	The operating system kernel consists of all system and application programs in a computer.
28.	A system call is triggered by hardware.
29.	A dual-core system requires each core has its own cache memory.
30.	The `exec()` system call creates a new process.
31.	For a single-processor system, there will never be more than one process in the Running state.
32.	A traditional (or heavyweight) process has a single thread of control.
33.	A thread is composed of a thread ID, program counter, register set, and heap.
34.	Linux distinguishes between processes and threads.

35.	Each thread has its own register set and stack.
36.	The value of a counting semaphore can range only between 0 and 1.
37.	A deadlock-free solution eliminates the possibility of starvation.
38.	The local variables of a monitor can be accessed by only the local procedures.
39.	Monitors are a theoretical concept and are not practised in modern programming languages
40.	A thread will immediately acquire a dispatcher lock that is the signalled state.
41.	Mutex locks and counting semaphores are essentially the same things.
42.	Mutex locks and binary semaphores are essentially the same things.

Long Answer

43.	What is a bootstrap program, and where is it stored?
44.	Describe the differences between physical, virtual, and logical memory.
45.	Distinguish between system and application programs.
46.	Describe why direct memory access (DMA) is considered an efficient mechanism for performing I/O.
47.	Describe why multi-core processing is more efficient than placing each processor on its own chip.
48.	Name and describe the different states that a process can exist in at any given time.
49.	Explain the main differences between a short-term and long-term scheduler.
50.	Explain the difference between an I/O-bound process and a CPU-bound process.
51.	Explain the concept of a context switch.
52.	List the four major categories of the benefits of multithreaded programming. Briefly explain each.
53.	Distinguish between parallelism and concurrency.
54.	What three conditions must be satisfied in order to solve the critical section problem?
55.	Describe the dining-philosophers problem and how it relates to operating systems.