1.	**Describe one strategy for dealing with deadlocks?**	1) we can use a protocol to prevent or avoid deadlocks, ensuring that the system will never enter a deadlocked state *2) We can allow the system to enter a deadlocked state, detect it, and recover *3) We can ignore the problem altogether and pretend that deadlocks never occur in the system	9.	**True or False? A spinlock is a type of mutex lock.**	True
2.	**How many writers may concurrently share the database with the readers-writers problem?**	one writer	10.	**True or False? Linux uses spinlocks for both single and multiple processor systems.**	false
3.	**If the current value of counter = 5, what are its possible values if the producer and consumer processes run concurrently?**	4,5, or 6	11.	**True or False? Semaphores can provide the same functionality as mutex locks.**	True
4.	**Name at least one modern programming language that has incorporated the idea of a monitor.**	1) Java *2) Mesa *3) C#	12.	**True or False? There are no guarantees Peterson's solution works correctly on modern computer architectures.**	True
5.	**Provide at least one alternative to mutex locks, semaphores, readerwriter locks, and monitors that provide support for concurrent programming.**	1) Transactional Memory *2) OpenMP *3) Functional Programming Languages	13.	**True or False? The system model for deadlocks first requires a process request a resource, then use the resource, and finally release the resource.**	...
6.	**True or False? A binary semaphore is functionally equivalent to a mutex lock.**	True	14.	**What are the four necessary conditions for characterizing deadlock?**	1) mutual exclusion- at least one resource miust be held in a nonsharable mode; that is,only one process at a time can use the resource 2) Hold and wait-a process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes 3) No preemption-resources cannot be preempted;that is, a resource can be released only voluntarily by the process holding it, after that process has completed its task 4) Circular wait- a set of waiting processes must exist such that P0 is waiting for a resource held by P1, P1 is waiting for a resource held by P2 etc
7.	**True or False? All solutions to the critical section problem are based on the premise of locking.**	True			
8.	**True or False? A nonpreemptive kernel is essentially free from race conditions.**	True			

15.	**What are the names of the two processes associated with the boundedbuffer problem?**	1) Producer 2) Consumer
16.	**What are the Pthreads operations for locking an unlocking a mutex lock?**	1) pthread_mutex_lock() 2) pthread_mutex_unlock()
17.	**What are the three requirements a solution to the critical-section problem must satisfy?**	1)mutual exclusion 2)progress 3)bounded-waiting
18.	**What are the two functions used with mutex locks?**	1) aquire() 2) release()
19.	**What are the two general hardware instructions that can be performed atomically?**	1) TestAndSet() 2)Swap()
20.	**What are the two operations that can be performed on a condition variable?**	1) x.wait() (wait()) 2) x.signal() (signal())
21.	**What are the two operations that can be performed on a semaphore?**	1)signal() 2)wait()
22.	**What are the two states of a Windows dispatcher object?**	1)signaled-state (object-avaliable) 2)non-signaled state
23.	**What is available in Linux for updating an integer variable without having to use locks?**	atomic integers
24.	**What is the only reasonable condition that can be used to prevent deadlocks from occurring?**	Circular-wait Impose a total ordering of all resource types, and require that each process requests resources in an increasing order of enumeration.
25.	**What is the problem if all philosophers simultaneously pick up their left fork?**	Deadlock- each philosopher is waiting infinitely for the next

26.	**What is the term for describing the situation where shared data may be manipulated concurrently and the outcome of the execution depends upon the order of access?**	Race Condition
27.	**What is the term used to describe the segment of code where shared data is accessed and possibly manipulated?**	Critical section