1. 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

B

☑

2. An instruction that executes atomically ____.

A) must consist of only one machine instruction

B) executes as a single, uninterruptible unit

C) cannot be used to solve the critical section problem

D) All of the above

B

3. 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

A

4. 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

B

6. The first readers-writers problem ____.

A) requires that, once a writer is ready, that 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.

C

A ___ type presents a set of programmer-defined operations that are provided mutual exclusion within it.

A) transaction

B) signal

C) binary

D) monitor

D

_____________ occurs when a higher-priority process needs to access a data structure that is currently being accessed by a lower-priority process.

A) Priority inversion

B) Deadlock

C) A race condition

D) A critical section

A

9. 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()

B

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()

C

_____ is not a technique for handling critical sections in operating systems.

A) Nonpreemptive kernels

B) Preemptive kernels

C) Spinlocks

D) Peterson's solution

D

A solution to the critical section problem does not have to satisfy which of the following requirements?

A) mutual exclusion

B) progress

C) atomicity

D) bounded waiting

C

A(n) ________ refers to where a process is accessing/updating shared data.

A) critical section

B) entry section

C) mutex

D) test-and-set

A

_____ can be used to prevent busy waiting when implementing a semaphore.

A) Spinlocks

B) Waiting queues

C) Mutex lock
D) Allowing the wait() operation to succeed
B

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.
D

How many philosophers may eat simultaneously in the Dining Philosophers problem with 5 philosophers?
A) 1
B) 2
C) 3
D) 5
B

18. 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 semaphore.
D) Counting semaphores can be used to control access to a resource with a finite number of instances.
C

_____ is/are not a technique for managing critical sections in operating systems.
A) Peterson's solution
B) Preemptive kernel
C) Nonpreemptive kernel
D) Semaphores
A

21. Which of the following statements is true?
A) Operations on atomic integers do not require locking.
B) Operations on atomic integers do require additional locking.
C) Linux only provides the atomic_inc() and atomic_sub() operations.
D) Operations on atomic integers can be interrupted.
A

A(n) ___________ is a sequence of read-write operations that are atomic.
A) atomic integer
B) semaphore
C) memory transaction
D) mutex lock
C

Another problem related to deadlocks is ____________.
A) race conditions
B) critical sections
C) spinlocks
D) indefinite blocking
D

Write two short methods that implement the simple semaphore wait() and signal() operations on global variable S.

```
wait( s ) {
while(s <= 0 ) { s--; }
}

signal( s ) {
s++;
}
```

Describe a scenario when using a reader-writer lock is more appropriate than another synchronization tool such as a semaphore.

A tool such as a semaphore only allows one process to access shared data at a time. Reader-writer locks are useful when it is easy to distinguish if a process is only reading or reading/writing shared data. If a process is only reading shared data, it can access the shared data concurrently with other readers. In the case when there are several readers, a reader-writer lock may be much more efficient.

Race conditions are prevented by requiring that critical regions be protected by locks.
T

The value of a counting semaphore can range only between 0 and 1.
F

A deadlock-free solution eliminates the possibility of starvation.
F

The local variables of a monitor can be accessed by only the local procedures.
T

A thread will immediately acquire a dispatcher lock that is the signaled state.
T

A nonpreemptive kernel is safe from race conditions on kernel data structures.
T

Linux mostly uses atomic integers to manage race conditions within the kernel.
F