CONCURRENCY AND SYNCHRONIZATION. QUIZZES. SOLUTIONS

Is it reasonable to use Peterson method to solve the synchronization critical - section problem when designing a new Macbook(R) Pro?

a) Yes
b) No
c) It depends on the components we order for such computer
d) There is not a clear answer.

Answer: b

Explanation: According to "Operating Systems Concepts" (p. 209), SW-based solutions such as Peterson's are not guaranteed to work on modern computer architectures.

Do producer process and consumer process have to be synchronized?

a) No, two processes can never be run concurrently
b) Yes, this way we avoid that the consumer tries to consume an item that has not yet been produced by (the producer)
c) No, the first process to be executed is the one whose priority is highest. The second process with higher priority will be executed secondly
d) Yes, the short-term scheduler is in charge of performing this parallel execution

Answer: b

Explanation: A producer process can produce one item while the consumer process is consuming another item. However, the producer and consumer have to be synchronized, so that the consumer does not try to consume an item that has not yet been produced (by the producer process)

The Race Condition problem is solved with...

a) The introduction of a Critical Section implemented by semaphores
b) The introduction of a specific C algorithm in stdio.h library
c) The introduction of a Lock implemented by semaphores
d) All of the previous are correct

Answer: a

Explanation: Because although a Critical Section can be implemented in many ways, the semaphore is one of them, but not the introduction of a specific C algorithm in stdio.h library. A lock is also a way to implement a Critical Section, but not a solution to the Race Condition neither.

Select the incorrect answer: Monitors...

a) Are stubs for concurrent functions
b) Only one processes can be active within the monitor at a time
c) Provide similar performance as semaphores
d) Are high level abstraction entities that provides a convenient and effective mechanism for process synchronization

Answer: a

Explanation: a, monitors are not stubs for anything

What is a semaphore?

a) A preemptive kernel.

b) A synchronization tool which is accessed by several operations at the same time.

c) A specific piece of code that can only be executed by a process at a given time.

d) A synchronization tool that can only be accessed through two atomic operations.

Answer: d)

Explanation: Semaphores are integer variables that apart from initialization can be accessed only through two atomic operations: wait() and signal(). Can be counting or binary, acting like locks to provide mutual exclusion.

Which of the following is not true regarding the solution to the critical section problem?

a) It requires mutual exclusion, progress and bounded waiting.

b) The easiest hardware-based solution are semaphores.

c) A feasible approach is preventing interrupts from occurring while a share variable is being modified.

d) Race conditions are prevented by requiring that critical regions be protected by a lock.

Answer: c)

Explanation:
Preventing interrupts is a simple solution for uniprocessor systems, because we can be sure that the current sequence of instructions will execute in order and no unexpected changes will be made to the shared variable. However, in multiprocessor systems is time-consuming since the message is passed to all the processors delaying the entry into each critical section. That is why it is not always feasible.

The implementation of semaphores, ¶

a) Must be done carefully to avoid the problems of deadlock and starvation.

b) Is restricted to only two values, 0 and 1, and they are also known as mutexes.

c) Consists of two functions: wait() and signal(), which respectively use the wakeup() and block() operation.

d) Allows a condition variable to be associated with several mutex variables for the POSIX threads.

Answer: a)

Explanation: if the operations wait() and signal() are not properly used, they can result in deadlock (processes waiting indefinitely for an event that can be only caused by one of the waiting processes) or starvation (processes may never be removed from the semaphore queue in which they are suspende d) . Moreover, b) is false since semaphores can also counting semaphores, ranging over an unrestricted domain; c) is false because wait() uses the block() and signal uses the wakeup(); and d) is false as the POSIX threads are constrained to have condition variable associated to only one mutex variable.

Select the correct statement concerning semaphores:

a) They can be implemented as a counting semaphore, as a binary semaphore or as a mutex lock

b) They are a hardware-based solution to the critical section problem and are only accessed through the atomic operations wait() and signal()

c) Busy waiting occurs when a process that tries to enter in its critical section has to wait for the execution of the critical section of other process and does not block itself

d) When using the block() operation in a process, the process for which this process is waiting is blocked and introduced in a waiting queue related to the semaphore, finally the process that executed the block() is waked up and introduced in the CPU.

Answer: c)

Explanation: C since mutex is the same than binary, the semaphore is software-based and the block() operation blocks the process that execute it and is used to prevent the busy waiting.

Select the correct statement:

a) Deadlock is caused when a process is waiting for another process to end but this other process is interrupted by the operating system.

b) When using priority inversion the processes with higher priority are executed first and the ones with lower priority wait in the semaphore queue.

c) With just one semaphore and two processes that only interact with each other is impossible to cause a deadlock.

d) When a procces stays in the semaphore queue indefinetly it is a deadlock problem.

Answer: c

Explanation: C since a deadlock is caused when two or more processes are waiting for an event that can only be caused by one of them and the priority inversion is the other way, the processes with high priority wait for the ones with low priority.

Which of the following would fix the critical section problem when operating with more than two processes?

a) Peterson's solution.

b) TestandSet.

c) Semaphore implemented with Swap instructions.

d) None of the above.

Solution: c)

Which of the following solutions for the problem of dining philosophers necessarily produces a deadlock?

a) Allowing each philosopher to only grab an even number of chopsticks.

b) Allowing only a even number of philosophers sitting in the table.

c) Representing each chopstick with a semaphore.

d) None of the above.

Solution: c)

Which of these is a requirement to solve a problem of Critical-section?

a) Mutual exclusion.
b) Progress.
c) Bounded waiting.
d) All of the above.

Answer: d

Explanation:
D since all of these are the requirements to solve the solution for a problem of critical-section.

Which of these is a necessary condition for a deadlock to occur?

a) Mutual Exclusion.
b) Hold and Wait.
c) All the choices are correct.
d) No preemption.

C is the correct answer, since for a deadlock to occur, those the possible answer in this question are necessary situations for a deadlock to occur, and if one of those conditions fail, the deadlock will be prevented.

The critical section is:

a) A state in which a process is forced to stop so that other process with more priority can be executed
b) A state in which a process is forced to stop due to his bad functionality
c) A segment of code which modifies and reads variables common to all processes
d) A segment of code which waits for permission for a exit section.

correct is c

A Deadlock is:

a) A state in which a process with low priority is never executed
b) A state in which 2 or more processes are waiting indefinitely for an event that can be caused by only one of the waiting processes.
c) A process that can not exit the critical section
d) None of them are correct

correct is b

1. Critical section:

a) Is the section of code in which requests are implemented.
b) Two processes can be executing in a critical section in order to change common variables.
c) If no process is executing in its CS, only processes executing in the remainder section can be selected to enter.
d) There must be a limit on the number of times that other processes are allowed to enter their CS after a process has made a request.

The answer is d) because bounded waiting is one of the requirements for the solution of CS problem as well as mutual exclusion (only a process can be executing in its CS, where can change variables, files...)

and progress (only processes not executing in remainder section can be selected to enter the CS).

2. Select the correct statement about monitors:
    a) The signal() operation has no effect if there is no process waiting.
    b) The overhead is smaller than mutex and semaphores.
    c) Monitor ensure only one process at a time can be active, so the programmer does not need to code this synchronization.
    d) All the answers are correct.

Answer: d)

Explanation: D since monitors are abstract entities composed by operations, data and functions where the synchronization is implicit. There are two operations on a condition variable wait() and signal(), which resumes a waiting process so only has effect if the former has been previously called, not always as in semaphores.

3. Select the incorrect statement:
    a) A semaphore is used for critical section problems.
    b) A mutex is a specific type of semaphore.
    c) An atomic operation can be interrupted.
    d) A mutex is dangerous because it can block a thread forever.

Correct answer is C.
Atomic operations can't be interrupted.

4. Select the false statement: A deadlock can be prevented by...
    a) allowing a thread to force a lock release from other threads.
    b) not allowing the locking of a second resource without releasing the first one.
    c) using more than one mutex simultaneously.
    d) avoiding circular wait conditions (threads waiting for each other to release resources).

Correct answer is C.
Multiple mutexes can cause deadlocks. It's the exact opposite.

What kind of problems appears with semaphore operations?
    a) Several processes are in critical section
    b) There is a deadlock
    c) Either there is a deadlock, or there are several processes in critical section
    d) All the answers are correct

Correct answer: d) All the answers are correct.

Which of the following are characteristics from monitors?
    a) They are high level abstractions that provide a convenient and effective mechanism for process synchronization
    b) Incorporated only in Java
    c) Several processes must be active within the monitor at a time
    d) None of the above

Correct answer: a) Other languages (like C#) have incorporated monitors, and only one process may be active within the monitor at a time. Quiz topic 6

Question 1: What the critical section does?

a) It interrupts the OS
b) It modifies and reads variables common to all processes
c) It modifies variables common to all processes
d) It reads variables common to all processes

Correct answer is b

Question 2: What is the simple solution for critical section implementation?

a) Provide the system with a special I/O device.
b) Semaphores
c) A GPU
d) Encapsulation

Correct answer is b

1.What is a deadlock:

a) indefinite blocking. A process may never be removed from the semaphore queue in which it is suspended.
b) two or more processes are waiting indefinitely for an event that can be caused by only one of the waiting processes.
c) Scheduling problem when lower-priority process holds a lock needed by higher-priority process.
d) remove one of processes in the waiting queue and place it in the ready queue.

The correct answer is b.

1. A philosopher can on the state 'Eating' when:

a) If both philosophers from each side are 'eating' also.
b) If and only if one philosopher is on the ‘eating'.
c) If both philosophers from each side are not 'eating'.
d) If that philosopher is not on the 'hungry' state.

The correct answer is c, the philosophers from each side can't be eating.1.

3.What is the main advantage of spinlocks?

a) No context switch is required when waiting
b) They are efficient when waiting for long periods of time for a lock
c) It does not waste CPU resources
d) They are more efficient than other types of locks in single processor systems

The correct answer is a. The main advantage of a spinlock is that no context switch is required and they are really helpful when context switches take a lot of time and the process does not have to wait for a long time for the lock.

Which one of the following would violate the critical section requirements

a) If a process if executing it's critical section no other process can execute it's critical section
b) Unbounded waiting there should be no limit on the number of times a process is allowed to enter the critical section when another process requested to enter it's critical section
c) If no process is executing only the processes that are not executing their remainder sections can participate in deciding which process will execute it's critical section next
d) All of them violate the requirements

The correct answer is b there the waiting should be bounded because if it is unbounded deadlocks will occur because a process might wait indefinitely.1.

Which of the following requirements must be satisfied to solve the critical-section problem?
a) If a process is executing in its critical section, then only processes with higher priority can be executing in their critical sections
b) If no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing in their remainder sections can participate in deciding which will enter its critical section next, and this selection can be postponed indefinitely
c) There exists a limit on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted
d) The three previous definitions correspond to requirements respectively known as "mutual exclusion", "progress" and "bounded waiting", and all of them must be satisfied to solve the critical-section problem

Correct answer: c.
Explanation: the last answer would be correct if option a) was "...then no other processes can be executing in their critical sections" and option c) was "...and this selection cannot be postponed indefinitely".

Which of the following sentences is not true?
a) Timing errors cannot occur when semaphores are used
b) A monitor type is and ADT
c) The monitor construct, without the condition construct, is not sufficiently powerful for modeling some synchronization schemes
d) Java and C#, among others, have incorporated the idea of the monitor

Correct answer: a.
Explanation: semaphores can have timing problems, like in the dining-philosophers original problem (without modifications to avoid the deadlock problem).

What is a semaphore?
a) It is a type of synchronization which uses an integer variable and two atomic operations.
b) It is a type of mutex using signals.

c) It is a hardware implemented method for achieving synchronization
d) None of the above.

Correct answer is a) . Explanation: A semaphore S is an integer variable that, apart from initialization, it is accessed only through two standard atomic operations: wait( ) and signal( ).

How can we provide a deadlock-free solution to the dining-philosophers problems?
a) Using semaphores for each chopstick.
b) Using mutex.
c) Using an asymmetric solution.
d) All of the above are correct.

Correct answer is c) . Explanation: Using semaphores could lead to a solution but can produce a dead lock, but using the asymmetric solution (that is,an odd-numbered philosopher picks up first her left chopstick and then her right chopstick, whereas an even- numbered philosopher picks up her right chopstick and then her left chopstick) would lead to a secure manner to solve this problem.

A Solution to the critical-Section problem IS NOT required to satisfy...Which condition?
a) It must ensure mutal exclusion
b) It has to change locks as fast as possible
c) It must ensure the progress of the program (I.e.The program must never stop indefinitely)
d) It must provide a limit number of times other processes can access to the critical section after a process has made a request of accessing the critical section.
Correct answer is b) . Although an efficient and optimal program is always preferred, the rest of the options are much more important than "speed".

Choose the incorrect statement:
a) To avoid the race condition, we need to assure that only one process manipulates a shared data each time.
b) If a Producer process and a Consumer process are executed at the same time, the value of the counter will not be affected; it will always be correct.
c) A process must acquire a lock before entering into a critical section and release this lock after leaving this critical section.
d) The TestAndSet instruction reads and modifies automatically a variable.

*The correct answer is b) because if The Producer process and Consumer process are executed concurrently, the value of the counter will be incorrect, thus it is false. The rest of the statements are true.

Select the INCORRECT statement regarding the race condition:
a) A preemptive kernel is free from race condition.
b) A solution to the race condition could be to ensure that the data that all processes want to access to manipulate can only be accessed by one of them at a

time.
      c) A nonpreemptive kernel is free from race condition.
      d) Kernel structures such as Structures for maintaining memory allocation are prone to possible race conditions.

The correct answer is a) . Explanation: a is incorrect because the preemptive kernel is not free from the race condition, it is the nonpreemptive the one free
because it can only have one process active in the kernel at a time. The others are true.

The critical section is:
      a) A section of code that when the process enters the risk of a coding error increases.
      b) A section of data reserved for critical instruction of the process.
      c) A section of data that can be accessed by several processes.
      d) A section of code in which a process changes common variables.

The correct answer is d) . Explanation: d is the correct one because the critical section is a section of code, not of data, in which the process modifies common variables, also it can update a table or wirte a file. This section must be executed by one process at a time to avoid race condition.

The race condition consists of
      a) Using hardware-implemented synchronization methods like swap
      b) Having a father process that is executed faster than the child
      c) Having two processes accessing to the same memory location
      d) Not being able to modify the processor registers due to a bad lock

The correct answer is c. The race condition is a common problem in multi-threaded programs that consists on having the threads trying to modify the same memory position at the same time, and, on this way, the output can be buggy since each thread will do a different thing without control.

In multi-threaded programming, a monitor is
      a) An useful tool for debugging our programs
      b) A high-level abstraction for avoiding two concurrent processes to enter at the same critical section
      c) A hardware implementation for making our threads to avoid race condition
      d) The most efficient method for using synchronization on any program

The correct answer is b. A monitor is a high level abstraction made in order to safely allow access to a method or a variable by more than one thread.

Which of the following is false about Peterson's solution to the critical-section problem?
      A) . The solution involves both a turn variable and an array of two to determine which process can enter the critical section.
      B) . The current process always tries to enter the critical section for itself first before letting the other process attempt.

C) . This type of approach is nonpreemptive.
D) . Bounded waiting is successfully implemented by not changing the turn variable inside the waiting while loop.

Answer: B, in Peterson's solution each process first sets turn to the other process before attempting to enter the critical section, thus each process surrenders the option to enter the critical section to the other process before attempting to enter themselves.

Which of the following can cause a deadlock situation in a semaphore implemented critical section?
A) . Switching the signal(mutex) and wait(mutex) pieces of code.
B) . Executing a wait(mutex) before and after the critical section.
C) . Omitting either the wait or signal command.
D) . Both B and C

Answer: D, in the case of B when a wait is executed before and after the critical section, no processes can enter in the critical section because no signals are created opening it up. In the case of C a deadlock can be reached since the necessary wait or signal to enter or exit a critical section won't be present.

What is a critical section?
a) The part of program with infinite loop.
b) The part of process where resources (variables, files, et c) are shared between other processes.
c) The part of process which is never executed.
d) The space in memory where a process is stored.
The correct answer is b. A critical section is a part of process where other processes can have influence - they can change values of variables, context of files etc. - resources which in the same time can use other processes.

Which atomic operations does a semaphore include?
a) An infinite loop.
b) A Boolean flag.
c) Wait and signal.
d) Addition and subtraction.

Answer: The correct answer is c. A semaphore include wait() and signal() operations. A Boolean variable is a part of hardware synchronization. An infinite loop and addition and subtraction are not parts of semaphore mechanism.

The requirement of the critical section is NOT:
a) mutual exclusion
b) progress - no infinitely postponed
c) bounded waiting
d) validity of the result

Answer: d. It is not a requirement. It is more like a goal to obtain.