Concurrency Answers

Which of the following statement(s) about the second solution for the dining philosopher's problem is(are) not true? Select the best answer.

A. **Incorrect.** This solution breaks the symmetry of the problem.
B. **Incorrect.** This solution can make sure half of the philosophers get to eat at any time.
C. **Correct.** This solution not only avoids deadlock but also has no race conditions.

Solution: There are race conditions between any pair of odd and even number philosophers when they are picking up their first chopsticks.

Recall that there are three semaphores used in the Bounded-buffer problem. Which of the following statements about the three semaphores are correct. Select all that apply.

A. **Correct.** Semaphore "mutex" is used to make sure only one process can update the buffer.
Semaphore "mutex" is placed before and after updating the buffer in both producer and consumer, so this option is correct.

B. **Correct.** Semaphore "full" is initialized to 0 and represent the available resource in the buffer.
Semaphore "full" will increase after the producer update the buffer and will block the consumer when there is no resource in the buffer, so this option is correct.

C. **Correct**. Semaphore "empty" is initialized to N and represent the available empty space in the buffer.
Semaphore "empty" will increase after the consumer updates the buffer and will block the producer when there is no resource in the buffer, so this option is correct.

In the reader-writer problem, wrt semaphore is used in both the writer process and the reader process. Please select options to fill in the two blanks to complete the statement about the usage of wrt in different processes.

Which of the following phrases correctly completes the sentence about wrt semaphores below?

The wrt semaphore is used in writer processes to…

(multiple choice question with **multiple** correct answer)

A. **Correct.** avoid any writers updating the data during the reading.
B. **Correct.** avoid writers getting into the critical section concurrently.

Explanation: The wrt is used at the beginning of the writer process to make sure no one is using the file before updating it. In the reader process, the wrt is used by the first reader to make sure the file is updated and no writer can change it during the reading.

In the reader-writer problem, wrt semaphore is used in both the writer process and the reader process. Which of the following correctly completes the sentence about wrt semaphores below?

The wrt semaphore is used in reader processes to…

(multiple choice question with **single** correct answer)

A. **Correct.** avoid any writers updating the data during the reading.

B. **Incorrect.** avoid writers getting into the critical section concurrently.

Explanation: The wrt is used at the beginning of the writer process to make sure no one is using the file before updating it. In the reader process, the wrt is used by the first reader to make sure the file is updated and no writer can change it during the reading.

In the reader-writer problem, wrt semaphore is used in both the writer process and the reader process. Which of the following correctly completes the sentence about wrt semaphores below?

The wrt semaphore is used in reader processes to…

(multiple choice question with **single** correct answer)

A. **Correct.** avoid any writers updating the data during the reading.
B. **Incorrect.** avoid writers getting into the critical section concurrently.

Explanation: The wrt is used at the beginning of the writer process to make sure no one is using the file before updating it. In the reader process, the wrt is used by the first reader to make sure the file is updated and no writer can change it during the reading.

Which of the following statements about the monitor with condition variables is true. Select the best answer.

A. **Correct.** A condition variable has both wait and signal functions.
B. **Incorrect**. The blocked threads in a monitor can be singled by any condition variables.

Solution: A condition variable has both wait and signal functions. A condition variable's signal function can wake up one of the threads blocked on it.

Is the following statement true or false? Select the best answer.

Conditional variables and semaphores are functionally equivalent.
(multiple choice question with ONE correct answer) → **1 point**
A. **Incorrect**. True
B. **Correct**. False
They are not functionally equivalent, e.g. semaphores maintain counters, while conditional variables are mostly used with monitors.

Is the following statement true or false? Select the best answer.

A binary semaphore can only have a value of either 0 or 1.
(multiple choice question with ONE correct answer) → **1 point**
A. **Correct**. True
B. **Incorrect**. False
A binary semaphore saturates at a value of 1 and is used as a mutex.

Is the following statement true or false? Select the best answer.

TSL (Test-and-Set) is a software-based solution for mutual exclusion.
(multiple choice question with ONE correct answer) → **1 point**

A. Incorrect. True
B. Correct. False
TSL is a hardware solution.

Is the following statement true or false? Select the best answer.

The Test-and-Set (TSL) is superior to the Peterson's solution because it does not result in busy waiting.
(multiple choice question with ONE correct answer) → **1 point**
A. Incorrect. True
B. Correct. False
TSL also suffers from the busy wait problem.

Is the following statement true or false? Select the best answer.

Two threads within the same process can write to the same memory.
(multiple choice question with ONE correct answer) → **1 point**
A. Correct. True
B. Incorrect. False
Two threads in the same process share the same virtual memory and hence can write to the same memory.

Is the following statement true or false? Select the best answer.

"Starvation freedom" means that if a thread/process wants to enter a critical section, it will eventually be granted permission to enter.
(multiple choice question with ONE correct answer) → **1 point**
A. Correct. True
B. Incorrect. False
A thread/process is not "starved" if after asking to enter a critical section, it enters eventually. In contrast, a starved process/thread will not get a chance to enter (forever, or at least it has to wait a long time).

Is the following statement true or false? Select the best answer.

In the context of semaphore implementations, "P" and "V" calls have to be atomic actions.
(multiple choice question with ONE correct answer) → **1 point**
A. Correct. True
B. Incorrect. False
The P and V calls have to be atomic, if not, the increase and decrease of the counter value will cause problems.

Is the following statement true or false? Select the best answer.

The Peterson's solution does **not** require simple assignments and tests since the shared variables do not have to be atomic.
(multiple choice question with ONE correct answer) → **1 point**
A. Incorrect. True
B. Correct. False
The whole basis of the Peterson's solution is its reliance on certain atomic statements (simple assignments and tests). This means no context can happen for these statements.

Suppose that Alice and Bob are customers in a bank. Alice currently has a balance of $300 and Bob has a balance of $700. They make the following transactions: Alice transfers $200 to Bob and Bob transfers $600 to Alice, using two different processes that uses the following code:

Process for Alice:
1. B = Balance (Bob)
2. A = Balance (Alice)
3. SetBalance(Bob, B + $200)
4. SetBalance(Alice, A - $200)

Process for Bob:
1. A = Balance (Alice)
2. B = Balance (Bob)
3. SetBalance(Alice, A + $600)
4. SetBalance(Bob, B - $600)

If there is no mutual exclusion and all steps can be interleaved, what is the minimum amount that Alice can end up with in her bank account after both processes are executed? You can assume that variables A and B are in shared memory and can be modified by either process.
(multiple choice with **single** correct answer) → **2 points**

A. **Incorrect.** $900
B. **Incorrect.** $700
C. **Correct.** $100
D. **Incorrect.** $300

Explanation: The following instruction interleaving produces this balance for Alice: A1, A2, B1, B2, A3 (Alice has $300 and Bob has $900), B3 (Alice has $900 and Bob has $900), B4 (Alice has $900 and Bob has $100), A4 (Alice has $100 and B has $100). This happens because Alice and Bob pulled their own copies of A and B at the beginning and override the SetBalance() instructions from each other.

Suppose that Alice and Bob are customers in a bank. Alice currently has a balance of $300 and Bob has a balance of $700. They make the following transactions: Alice transfers $200 to Bob and Bob transfers $600 to Alice, using two different processes that uses the following code:

Process for Alice:
5. B = Balance (Bob)
6. A = Balance (Alice)
7. SetBalance(Bob, B + $200)
8. SetBalance(Alice, A - $200)

Process for Bob:
5. A = Balance (Alice)
6. B = Balance (Bob)
7. SetBalance(Alice, A + $600)
8. SetBalance(Bob, B - $600)

Using the same scenario as in the question above, consider the following execution sequence:

Time	Process 1	Process 2
0	1.B = Balance (Bob)	1.A = Balance (Alice)

1	3.SetBalance(Bob, B + $200)	3.SetBalance(Alice, A + $600)
2	2.A = Balance (Alice)	2.B = Balance (Bob)
3	4.SetBalance(Alice, A - $200)	4.SetBalance(Bob, B - $600)

At each time X, process 1 and 2 execute each action concurrently (whether process 1 or process 2's statement runs first may be different every time, however, time 1 happens after time 0, and time 2 happens after time 1, etc. What is the output of A and B, respectively at time X = 3?

Enter your answer for A first, followed by your answer for B. Enter your answer for A and then B separated by commas. For example, you would enter 100, 500 if the output for A is 100 and the output for B is 500.
(free response question; text answer)

Correct Answers:
A: 700
B: 300

Explanation:
Time 0: B=700, A=300
Time 1: B=900,A=900
Time 2: A=900, B=900
Time 3: A=700,B=300