

The enter_CS() and leave_CS() functions to implement critical section of a process are realized using test and set instruction as follows-void enter_CS(X){

while (test-and-set(X));
}

void leave_CS(X){

X = 0;

In the above solution, X is a memory location associated with the CS and is initialized to 0. Now, consider the following statements-

i. The above solution to CS problem is deadlock-free

ii. It ensures mutual exclusion

iii. The processes enter CS in FIFO order

iv. More than one process can enter CS at the same time.

Which of the above statements is true?

○ II only

I and II

II and III

OIV only



Submit

You have used 1 of 1 attempt



Checkboxes

0.0/2.0 points (graded)

Suppose someone suggested the following code to the critical section problem for many processes (Process 0 to N where N > 2) using the compare and swap instruction. Decide if the proposed solution is correct or identify any issues with the proposal. There is no partial marking for this question. (&lock is how you access the lock variable in a code low level code, so don't worry about the & sign in the code.)

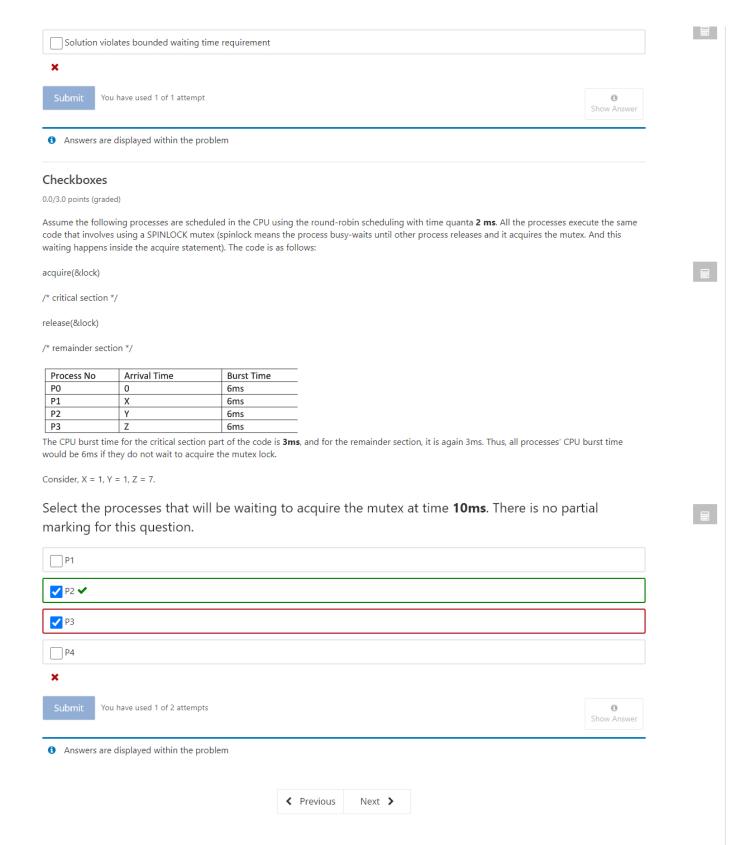
```
do {
      waiting[i] = true;
      key = true;
      while (waiting[i] && key) {
            key = compare_and_swap(&lock, 0, i) != 0;
      waiting[i] = false;
      /* critical section */
      j = (i + 1) \% n;
      while ((j != i) && !waiting[j]) {
            j = (j + 1) \% n;
      if (j == i) {
            lock = 0;
      } else {
            waiting[j] = false;
      /* remainder section */
} while (true);
```

Select the right ones from the following -

Solution is correct.



✓ Solution violates progress requirement



© All Rights Reserved



About Us BracU Home USIS Course Catalog