**Quiz 3**

 Bookmark this page

Definitions

2/2 points (graded)

Please match the following terms with the letter of their appropriate definitions.

|  |  |
| --- | --- |
| Block    correct | A. A thread will stop running until a certain condition is reached |
| Suspend    correct | B. A thread is completely removed from the OS |
| Sleep    correct | C. A thread decides it no longer needs to run |
| Preempt    correct | D. One thread is running and the SysTick ISR stops running that thread |
| Kill    correct | E. A thread will not run for a prescribed amount of time |

Submit

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Show Answer

Correct (2/2 points)

Review

Efficiency

2/2 points (graded)

Lab 2 implemented spinlock semaphores, and Lab 3 implemented blocking semaphores. In section 3.1 we presented cooperative spinlock semaphores that added an **OS\_Suspend()** in the while loop of the **OS\_Wait()** implementation, so the thread will suspend if the semaphore is equal to 0. For this problem, assume the time to switch threads (about 1us) is very small compared to the periodic interrupt used to run the scheduler (10ms).

Which of the following best describes the efficiency of the three semaphore implementations

Blocking is better than cooperative spinlock, and cooperative spinlock is better than regular spinlock

Blocking is about the same as cooperative spinlock, and both cooperative spinlock and blocking are better than regular spinlock correct

Blocking is better than both cooperative spinlock and regular spinlock, and cooperative spinlock and regular spinlock are about the same

You can not tell without implementing them

Submit

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Show Answer

Correct (2/2 points)

Review

Bounded Waiting

2/2 points (graded)

What is bounded waiting?

When we wake up a sleeping thread we wakeup the first one we find

There is a maximum time we will wait blocked on a resource

Once blocked there are a finite number of threads that will capture the resource before this thread is granted permission correct

We are allowed to continue to run without the resource and will be given the resources when it is available.

None of the above.

Submit

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Show Answer

Correct (2/2 points)

Review

Producer/consumer

2/2 points (graded)

Consider the situation in which a FIFO queue is used to buffer data between one producer thread and a consumer thread. The producer thread generates data and then it puts the data into a FIFO. The consumer thread gets data from the FIFO and processes it. The consumer can process 1000 bytes/sec on average with a maximum of 1500 and a minimum of 500 bytes/sec. The producer generates data at a fixed rate of 750 bytes/sec. It would be very bad to loose data if the producer blocks on a full FIFO.

What will happen?

The system could work, but the bandwidth would increate a lot if the consumer ran faster

The system could work, but the bandwidth would increate a lot if the producer ran faster

The system will NOT work, but it would work if the consumer ran faster

The system will NOT work, but it would work if the producer ran faster

If the FIFO is big enough, this system will not loose data correct

Submit

Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Show Answer

Correct (2/2 points)

Review

Sleeping

2/2 points (graded)

Assume you implement sleeping as described in section 3.4 and Lab 3. There are *N* threads and the thread switch time is *dt*. A thread calls OS\_Sleep(10), where the units of the sleep time are ms.

What is the minimum sleep time?

10ms+ *dt*

*N\*dt*

10ms correct

10ms+ *N*\* *dt*

10ms+( *N*-1)\* *dt*

10ms- *dt*

What is the maximum sleep time?

10ms+ *dt*

*N\*dt*

10ms

10ms+ *N*\* *dt*

10ms+( *N*-1)\* *dt* correct

10ms- *dt*

**Explanation**

If you are lucky then the thread sleep goes off at 10ms and this thread runs next. So the minimum will be 10ms.

If you are unlucky then the thread sleep goes off at 10ms and there are ( *N*-1) other threads that run first. So the maximum will be 10ms+( *N*-1)\**dt*.