eLEARNING University of West Florida

SYS & NET I - 0498 COP4634 201308

COURSE MATERIALS COMMUNICATION ASSESSMENTS MY TOOLS

Quiz Submissions - Homework 1

Daniel Davis (username: DanielDavis.dkd6)

Attempt 1

Written: Sep 29, 2013 8:57 PM - Sep 29, 2013 9:01 PM

Submission View

released: Sep 30, 2013 1:00 AM

Question 1 **1 / 2 points**

Which of the following instructions should be privileged, i.e. run in kernel mode?

✗ ☑ Write to main memory.

➡ ✓ ☑ Turn off interrupts.

➡ ✓ ☑ Set the clock.

➡ ✗ □ Access data on a DVD drive.

Hide Feedback

This is not a privileged instruction. A user program may access memory in user mode.

Privileged so that a process cannot control CPU time and disable any communication between devices and CPU.

It is privileged so that a process cannot be arbitrarily suspended from the CPU and replaced by another process for the duration of its time.

Question 2 **2 / 2 points**

Select activity or activities that involve interrupt handlers in the OS.

✓ ☑ Receive events from keyboard or the mouse

✓ ☑ Printing a message to the screen.

✓ ☑ Open a file in a word processor program.

✓ □ Sort numbers stored in an array in memory assuming all data are in memory.

Hide Feedback

Pressing a key on the keyboard or moving the mouse triggers hardware interrupts that cause the processor to read the keystroke or mouse position.

Printing a message requires access to the graphics adapter, which is an output device. Any access of a device requires privileged instructions.

Opening a file requires system call, which triggers a switch to kernel mode. It is a privileged instruction to protect unauthorized access from user files.

Question 3 **0.5 / 0.5 points**

Classify the vending machine as a hard or soft real-time system:

○ hard real-time

✓ ● soft real-time

Hide Feedback

It is sufficient to priorities the tasks that the vending machine has to complete.

Question 4 **0 / 0.5 points**

Classify the robot arm in a manufacturing plant as a hard or soft real-time system:

✗ ● soft real-time

➡ ○ hard real-time

Hide Feedback

This cannot be a soft real-time system because all jobs need to be completed within fixed time constraints.

Question 5 **0.5 / 0.5 points**

Classify the control system of an elevator as a hard or soft real-time system:

○ soft real-time

✓ ● hard real-time

Hide Feedback

All tasks must be completed within fixed time constraints to ensure that the elevator works properly.

Question 6 **0.5 / 0.5 points**

Classify the aircraft navigation system as a hard or soft real-time system:

○ soft real-time

✓ ● hard real-time

Hide Feedback

Hard because all jobs should be completed on time and catastophic events may follow if not completed on time.

points

Consider a multi-core computer system and a multi-threaded program written using the many-to-many threading model. Which of the following scenarios may lead to an optimal performance assuming the program is CPU-bound (computationally intensive)?

✗ ● The number of threads is below the number of cores in the system

➡ ○ The number of threads matches the number of cores in the system

○ The number of threads far exceeds the number of cores in the system

Hide Feedback

Having fewer threads than cores generally means you can't take advantage of all available cores. This will result in degradation of performance.

Question 8 **1.5 / 2 points**

Indicate which of the following actions by the OS are necessary to create a new process using system call fork()? (Note: order not relevant)

➡ ✓ ☑ Switch from user mode to kernel mode.

✗ ☑ Locate free memory.

➡ ✓ ☑ Indentify a new process ID.

➡ ✓ ☑ Create a new process control block in memory.

Hide Feedback

A user process will enter the kernel-mode when it starts executing fork(), which is a privileged instruction. Other cases, when the user process will switch to kernel mode include:
When it is interrupted (e.g. by the timer).
When an exception occurs (e.g. divide by 0).

The new process created will initially not be located in a separate address space. It will initially share the same address space as the caller process.

System call fork() is used to create new process. It takes no arguments and returns a process ID.

Process control block is a data structure in the kernel that contains information needed to manage a particular process.

Attempt Score: 6 / 10

Overall Grade (highest attempt)**:** 6 / 10

Close