OS Tasks, Processes & Threads

This assignment is due on September29th at 11:00 PM. Post all your answers to the corresponding quiz in *eLearning*. Late assignments will not be accepted. See course syllabus for details on all course policies.

Problem 1 – 2 Points

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

a) Set the clock.
b) Write to main memory.
c) Access data on a DVD drive.
d) Turn off interrupts.

Problem 2 – 2 Points

For the following activities check involvement of an interrupt handler in the OS as triggered by either a software or hardware interrupt

a) receive events from keyboard or the mouse
b) open a file in a word processor program
c) print a message to the screen
d) sort numbers stored in an array in memory assuming all numbers are in memory

Problem 3 – 2 Points

Classify the following systems as a hard or soft real-time system. A hard real-time system ensures that tasks are executed within fixed time constraints to ensure proper responsiveness; failure to comply with the constraints may have catastrophic consequences. A soft real-time system only ensures that tasks are executed according to their priorities.

a) A vending machine.
b) A robot arm in a manufacturing plant.
c) The control system of an elevator.
d) The navigation system of an aircraft.

Problem 4 – 2 Points

Consider a multicore computer system and a multithreaded program written using the many-to-many threading model. Which of the following scenarios may lead to an optimal performance for the program assuming that the program is CPU-bound (computational intensive)?

a) The number of threads far exceeds the number of cores in the system.
b) The number of threads matches the number of cores in the system.
c) The number of threads is below the number of cores in the system.

Problem 5 – 2 Points

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

a) Identify a new process ID.
b) Allocate free memory for the process.
c) Create a new process control block in memory.
d) Switch from user mode to kernel mode.