CS 143A: Principles of Operating Systems

Due Date: Thu, 14 April 2016, 11:55 PM via EEE DropBox
Homework #1 (Total Marks=100)

Question 1: Polling, Interrupts, Traps [6, 6, 4, 4]

a. Polling and interrupts are two ways used by operating systems to check whether I/O is done. Briefly describe how each of them works, and explain the main difference between these two (no more than 6 lines each).

b. If an active typist is typing 60 words per minute into a word processor considering 4 character-per-word on average, and the computer operates at one billion machine instructions per second, how many machine instructions can be executed in the time it takes a character to be typed on the keyboard?

c. Which of the following I/O methods is interrupt driven?
   1. Synchronous I/O
   2. Asynchronous I/O

d. What is the difference between an interrupt and a trap?

Question 2: Parallel, Distributed, Real-time Systems [10, 6, 4]

a. Briefly explain the difference between multiprogramming and timesharing.
b. Give two real-world examples for distributed computing system applications and explain why the systems are distributed.
c. For each of the following systems, indicate whether they are a) hard real-time systems, b) soft real-time systems, or c) not real-time systems at all by checking ☒ the appropriate boxes. Give one sentence for each of your choices to briefly explain why.

		Hard real-time	Soft real-time	Not real-time
1.	Spotify			
2.	Android system interface			
3.	Adaptive cruise control for vehicles			
4.	High-frequency stock trading system			

Question 3: System Calls [9, 6, 5]

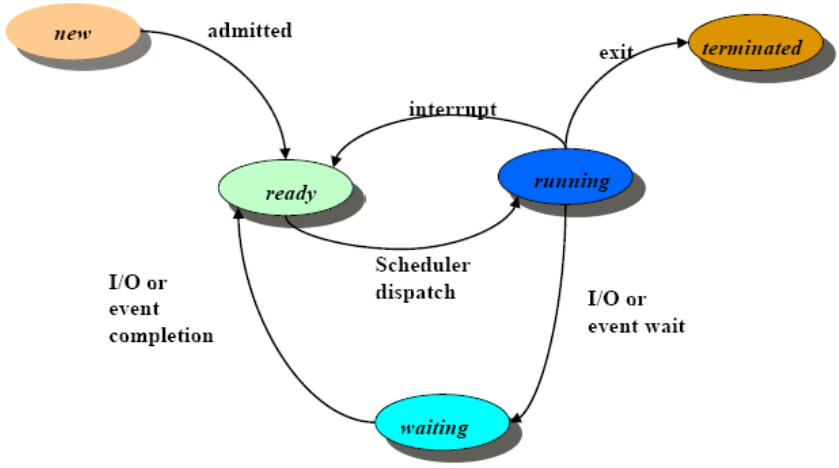
a. For each of the brief description about the following system calls used in Unix, write the name of the system call and its parameters. (Hint: Use the *man* pages.)
   1. Read from a file descriptor
   2. Wait for process termination
   3. Create a child process
b. Suppose you want to write a program that opens the file passed to it on the command line (argv[1]) and writes another file back out (with the same contents). List 3 system calls that will be used in this program.

c. For each of the following operations, indicate whether they should be privileged or not by checking ☒ the appropriate boxes.

	Yes	No
Searching a string for a particular sequence		
Generate an interrupt		
Process hardware interrupt		
Check list of currently running processes		
Access network card buffer		

Question 4: Processes [15, 5]

a. The following figure shows all states and state transitions during the lifetime of a process (see figure below). As discussed in class, a medium-term scheduler can swap a process in or out of memory. When a process is swapped out, it is *suspended*.
Modify the state diagram below by drawing in all new states and state transitions (using directed egdes) that are necessary to accommodate the new *suspended* state. Also name the conditions under which the state transitions you added should occur.



b. For each of the following, indicate whether they are stored in the process control block by checking ☒ the appropriate boxes.

	Yes	No
CPU registers		
Number of processes		
Program counter		
Process id		
File permissions		

Question 5: Threads[6, 5, 5, 4]

a. Why is that threads are faster to create than processes?
b. Suggests one application that benefit from the use of threads
c. Distinguish between user-level threads and kernel-level threads in terms of cost of switching. Which of these is more expensive to switch? Explain briefly.
d. When a process is multithreaded, indicate which of these resources are shared among the threads, and which are private to each thread by checking ☒ the appropriate boxes

	Shared	Private
Memory		
Program counter		
Execution stack		
Global variables		