

Home work – Set 2

1. How does a user process get privileged operations performed?
2. Name three ways in which the processor can transition from user mode to kernel mode? Can the user execute arbitrary code after transition?
3. Which of the following C library functions do NOT directly correspond to (similarly named) sys-tem calls? That is, the implementations of which of these C library functions are NOT straight forward invocations of the underlying system call?
 - a) system, which executes a bash shell command
 - b) fork, which creates a new child process.**
 - c) exit, which terminates the current process.**
 - d) strlen, which returns the length of a string.
4. Which of the following would not necessarily cause a process to be interrupted?
 - a) Division by zero
 - b) reference outside user's memory space
 - c) page fault
 - d) accessing cache memory
 - e) end of time slice
 - f) none of the above
5. Suppose a machine (architecture: x86, single core) has two runnable processes P1 and P2. P1 executes a line of code to read 1KB of data from an open file on disk to a buffer in its memory. The content requested is not available in the disk buffer cache and must be fetched from disk. Describe what happens from the time the instruction to read data is started in P1, to the time it completes (causing the process to move on to the next instruction in the program), by answering the following questions.
 - a) The code to read data from disk will result in a system call, and will cause the x86 CPU to execute the int instruction. Briefly describe what the CPU's int instruction does.
 - b) The int instruction will then call the kernel's code to handle the system call. Briefly de-scribe the actions executed by the OS interrupt/trap/system call handling code before the read system call causes P1 to block.
 - c) Now, because process P1 has made a blocking system call, the CPU scheduler context switches to some other process, say P2. Now, the data from the disk that unblocks P1 is ready, and the disk controller raises an interrupt while P2 is running. Describe the actions performed by P2 in kernel mode when servicing this disk interrupt.
 - d) Right after the disk interrupt handler has successfully serviced the interrupt above, and before any further calls to the scheduler to context switch from P2, what is the state of process P1?

6. A system call is a routine built into the kernel and performs a basic function.
 - a) True
 - b) False

7. When we execute a C program, CPU runs in _____ mode.
 - a) user
 - b) kernel
 - c) supervisory
 - d) system

8. In _____ mode, the kernel runs on behalf of the user.
 - a) user
 - b) kernel
 - c) real
 - d) all

9. open system call returns the file descriptor as _____
 - a) int
 - b) float
 - c) char
 - d) double

10. A process in user mode cannot execute certain privileged hardware instructions. [T/F]

11. Which of the following C library functions do NOT directly correspond to (similarly named) system calls? That is, the implementations of which of these C library functions are NOT straight-forward invocations of the underlying system call?
 - a) system, which executes a bash shell command
 - b) fork, which creates a new child process
 - c) exit, which terminates the current process.
 - d) strlen, which returns the length of a string