

COS 331  
Homework1  
Due Friday, September 26<sup>th</sup>, 11:59PM

NOTE: make sure you carefully read the submission requirements posted with the assignment!

**Homework Questions**

- 1) How does system hardware interact with (get the attention of) the OS?
- 2) How do user programs interact with (make a request of) the OS?
- 3) What are the two main functions of the OS?
- 4) True or False: the OS is dependent on hardware support for its proper functioning.
- 5) Provide two examples of software interrupts.
- 6) What is stored in the Systems Call Table?
- 7) What is the purpose of a Trap instruction?
- 8) Would a divide by 0 error generate a non-maskable interrupt? Why or why not?
- 9) What might cause the generation of a non-maskable interrupt?

- 10) As discussed in class, the 'execution state' of a process is restored (so that the program can continue its execution) after a system call is completed. What is the last register of the process's state to be restored and why?
- 11) We talk about "jumping" to an Interrupt Service Routine through the Interrupt Vector. What does that involve?
- 12) What is one (specific) approach that the OS can use to ensure that a process stays within its own address space? You need to provide an explicit approach. Saying 'MMU' or 'Virtual Memory' is not specific (nor covered in class).
- 13) What are five major activities of an operating system regarding process management?
- 14) The mode bit is used in conjunction with { } to ensure that a user-level process cannot execute an instruction that is potentially harmful to the system.
- 15) Briefly explain how the OS provides a convenient interface for programmers with respect to program memory.
- 16) What is a context switch?
- 17) Is 'printf' a system call?

18) What are three major activities of an operating system regarding memory management?

19) How does the OS guarantee it will regain control when turning control of the CPU to a user application? Be specific—

20) Which of the following instructions should be privileged?

- a) Set the system timer.
- b) Read the system timer
- c) Turn off interrupts.
- d) Issue a trap instruction.
- e) Access a device driver.
- f) Write to the terminal.

21) As noted, an OS provides N system calls numbered from 0 .. N-1. There is an include file (called “syscall.h”) that provides the mapping between system call names and their number. A portion of the [syscall.h](#) file from a Mac is shown below on the left and a portion of the [system call table](#) is shown below on the right:

```
#define SYS_syscall      0
#define SYS_exit         1
#define SYS_fork         2
#define SYS_read         3
#define SYS_write        4
#define SYS_open         5
#define SYS_close        6
#define SYS_wait4        7
//lots more
```

Portion of syscall.h

0xBB00	0x DD00
0xBB04	0x DD04
0xBB08	0x DD08
0xBB0A	0x DD0C

Portion of System Call Table (starts at address 0xBB00)

Assume the program invokes the [read](#) system call. What is the address of the first instruction of the handler for the read system call?

22) In what way do system calls protect underlying system resources?

23) In what way do system calls provide a convenient interface for users?

24) Provide a C program that creates a new process with the `fork()` system call. After the `fork()`, the parent process should write a message stating that it is the parent and the child process should print out a message saying it is the child. Please provide your code below.