**CS3502 OPERATING SYSTEMS**

**Mid Term Exam Sample Questions and Study Guide**

* **All questions from HW1 and HW2.**
* **All UNIX/Linux system command from HW1 and HW2**
* **All C/C++/Java code covered so far**
* **Text Book coverage: Ch1 .. Ch5**
  + On OS Introduction ch1
  + On OS Structures:                  2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7
  + On Process Concepts:             3.1, 3.2, 3.3, 3.4, 3.5, 3.6
  + On Threading/Concurrency:   4.1, 4.2, 4.3, 4.4, 4.5, 4.6
  + On CPU Scheduling:              5.1, 5.2, 5.3, 5.5

1. Linux file has three levels of security associated with it that matches the three classes of users that may access that file. What are those?
2. Mention some of your favourite disk utilities and commands.
3. All computers follow roughly the same set of steps to transition from a power-off state to a running state. Can you enumerate the steps of the booting sequence?
4. What do you understand by IRQ?
5. Define the following terms in your own language:
6. Shell
7. Process
8. In the Lab, we carried out experiment for changing file attributes. Suppose I write a command:

chmod 700 foo

What the command is actually doing?

1. What is the difference, if any, between the following terms:
   1. Proprietary vs. open operating system
   2. Batch vs. interactive operating system
   3. Host vs. guest operating system
2. Define the term, direct memory access (DMA)
3. Can UNIX fork() return an error? Why or why not?
4. Can UNIX exec() retuen an error? Why or why not?
5. How many processes are created if the following program is run?

main(int argc, char \*\* argv) {

forkthem( 5) ;

}

void forkthem (int n) {

If ( n > 0) {

fork();

forkthem(n-1);

}

}

**}**

1. Briefly describe (with at least 4 to 5 sentences for each question) the two major functions of an operating system using suitable examples for each one of the scenarios
2. The following questions relate to Boot Loaders
   1. What is a boot loader and what is its primary purpose?
   2. Where is a boot loader found on a permanent storage device such as a hard disk
   3. What is a 2 stage boot loader?
   4. Why is 2-state boot loading used?
3. What is a system call and how does it operate?
4. Describe three general methods for passing parameters to the operating system in a system call
5. From a Linux perspective, briefly (2 sentences) describe the use of user id (a number) and group id (a number)
6. What is virtual memory? Why is it used on most modern operating systems?
7. In operating system terminology (and not in data structure terminology) what is heap and what is stack?
8. What is the difference between source code, pseudo code, and an algorithm
9. Describe the difference between syntax and semantic errors. How do you detect them? How do you fix them?
10. Briefly (2 to 3 sentences each) discuss pass-by-value versus pass-by-reference mechanisms for passing parameters to methods.
11. Briefly describe the difference between compiling and linking phases involved in creating an executable from a C++ program
12. What is a function prototype and when is it needed
13. What do you mean by *big-endian* and *little-endian* system?
14. Keeping in mind the various definitions of ***operating system***, consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answers.
15. What are the two models of inter process communication? What are the strengths and weaknesses of the two approaches?
16. Explain the difference between a monolithic kernel and a microkernel.
17. What does it mean to preempt a process?
18. What is the purpose of the UNIX pipe command, i.e., vertical bar character |
19. What is POST?
20. What do you mean by LILO and GRUB?
21. What do you know about BIOS? In the class we discussed that PC/BIOS PROM monitor is rather limited in its ability to access the system hardware. What new standard is being proposed to replace the traditional legacy BIOS?
22. What are the differences between a trap and an interrupt? What is the use of each function?
23. What is the purpose of the command interpreter? Why is it usually separate from the kernel? Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?
24. What is the purpose of interrupts? What are the differences between a trap and an interrupt? What is the use of each function? How are multiple interrupts dealt with? Can traps be generated intentionally by a user program? If so, for what purpose?
25. What is the purpose of system calls, and how do system calls relate to the OS and to the concept of dual-mode (*kernel* mode and *user* mode) operation?
26. Why do some operating systems store the operating system in firmware, while others store it on disk?
27. Study Memory hierarchy
28. Explain the Role of the Linker and Loader (g++ -o –c etc.)
29. Draw process state diagram.
30. For shortest job first, if the scheduler assigns a task to the processor, and no other task becomes schedulable in the meantime, will the scheduler ever preempt the current task? Why or why not?
31. Devise a workload where FIFO is pessimal – it does the worst possible choices – for average response time.
32. Suppose you do your homework assignments in SJF-order. After all, you feel like you are making a lot of progress! What might go wrong?
33. Most round-robin schedulers use a fixed size quantum. Give an argument in favor of and against a small quantum.
34. A CPU-scheduling algorithm determines an order for the execution of its scheduled processes. Given *n* processes to be scheduled on one processor, how many different schedules are possible? Give a formula in terms of *n*.
35. Explain the difference between preemptive and non-preemptive scheduling.
36. What advantage is there in having different time-quantum sizes at different levels of a multilevel queueing system?
37. Why is it important for the scheduler to distinguish I/O-bound programs from CPU-bound programs?
38. Which of the following scheduling algorithms could result in starvation?
    1. FCFS b. SJF c. RR d. Priority
39. How does the kernel know if an application is in an infinite loop?
40. All computers follow roughly the same set of steps to transition from a power-off state to a running state. Can you enumerate the steps of the booting sequence?
41. What are the two main functions of an operating system?
42. What is the difference between timesharing and multiprogramming systems?
43. What is the difference between kernel and user mode? Explain how having two distinct modes aids in designing an operating system.
44. A computer has a pipeline with four stages. Each stage takes the same time to do its work, namely, 1 *nano second*. How many instructions per second can this machine execute?
45. What is a trap instruction? Explain its use in operating systems.
46. When an interrupt or a system call transfers control to the operating system, a kernel stack area separate from the stack of the interrupted process is generally used. Why?
47. Explain how time quantum value and context switching time affect each other, in a round-robin scheduling algorithm.
48. Five jobs are waiting to be run. Their expected run times are 9, 6, 3, 5, and X. In what order should they be run to minimize average response time? (Your answer will depend on X.)
49. The aging algorithm with a = 1/2 is being used to predict run times. The previous four runs, from oldest to most recent, are 40, 20, 40, and 15 msec. What is the prediction of the next time?