Operating Systems Sample Questions

Processes

(1) (a) What is a process? (b) How is a process different from a program?

(2) In the memory layout of a typical process, why do stack and heap grow towards each other (as opposed to growing in the same direction)?

(3) In terms of call-return behavior, how are the fork() and exec() system calls different from other system calls?

(4) Describe the process lifecycle illustrating the states and transitions.

(5) Which state transitions occur in a process lifecycle when a process
   A. Makes a blocking read() system call
   B. Exceeds its CPU timeslice
   C. Is interrupted by a hardware interrupt
   D. Dereferences a NULL pointer.
   E. Attempts to acquire a blocking lock that is taken by another process?
   F. Is pre-empted
   G. Voluntarily yields the CPU

(6) Which state transitions (if any) occur in a process lifecycle when a process
A. Runs too long on the CPU?
B. Tries to read keyboard input, but no input is available?
C. Receives a signal?
D. Attempts to execute a System ISA instruction in user space?
E. Attempts to perform down() operation on a semaphore whose value is zero?

(7) During a process lifecycle, what events can cause the following transitions?

   (a) Ready to Running state

   (b) Running to Ready state

   (c) Ready to Blocked state

   (d) Blocked to Ready state

(8) What is a zombie process? Why does the Operating System maintain the state of zombie processes? List two ways in which a parent process can prevent a child process from becoming a zombie.

(9) Why are frequent context switches expensive in terms of system performance?

(10) What is cold-start penalty? What are some ways to reduce it?

(11) What are some key factors that affect application performance after a context switch?

(12)