CSci 144: Introduction to Operating Systems

Quiz 1 (Fall 2008)

8:00-8:40am, McF 208, 10/07/2008

Name (Last, First): _____

Multiple choices (please select ONLY one answer) (9 points)

1. Which statement about resources is false?
   a. A process/thread must request a resource before using it
   b. The operating system can provide resources
   c. Resources may be requested from another process
   d. A thread suspends its operation until a requested resource is allocated.

2. Operating system functions are normally categorized into one of these categories except
   a. Process, thread and resource management
   b. Memory management
   c. Device management
   d. Window management

3. Primary operating system design issues include the following except
   a. Efficiency in the use of machine resources
   b. Compact memory representation
   c. Resource isolation
   d. Maximizing availability of resource for use by applications

4. Which is not required for a thread to overlap its CPU and I/O operations?
   a. The thread has other work to do, while waiting for the I/O to complete
   b. The programming language must provide support for overlap
   c. The I/O operation must be slow, relative to CPU speed
   d. The OS must provide tools to poll the device

5. Which statement about switching threads is incorrect?
   a. A new PC value (for a different thread) can be chosen by the scheduler
   b. A trap instruction is executed
   c. An interrupt occurs
   d. A process requests a control transfer to a thread in another process

6. Basic process states include the following except
   a. Blocked b. done c. running d. ready

7. Which is the least common reason that a running thread might cease using the CPU?
   a. Thread completes execution
   b. Thread requests resource, and blocks
   c. Thread voluntarily releases CPU
   d. Thread involuntarily releases CPU

8. Non-preemptive scheduling strategies that commonly rely on process/thread execution time include the following except
   a. FCFS b. SJN c. Priority d. Deadline

9. Which statement about disabling interrupts to resolve race conditions is wrong?

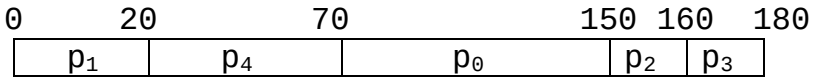
a. In theory, a program can disable interrupts when it enters a critical section, and re-enable interrupts when finished with a critical section, to eliminate race conditions.
b. Disabling/enabling interrupts may negatively affect the I/O system.
c. Programs with infinite loops in their critical sections are a significant problem with the interrupt-based approach.
d. User-mode programs are the best place to invoke disableInterrupt().

Short problems: (6 points)

1. Assume that you use priority scheduling where a small integer means a higher priority

i	execution time	priority
0	80	3
1	20	1
2	10	4
3	20	5
4	50	2

a. Calculate the turnaround time for process p2.
b. Calculate the average wait time for the processes.



a. Turnaround time for p_2 = 160
b. Average wait time

W(p_0) = 70

W(p_1) = 0

W(p_2) = 150

W(p_3) = 160

W(p_4) = 20

Average = (70+0+150+160+20)/5 = 80

2. Two processes, p1 and p2, have been designed so that p2 prints a byte stream produced by p1. Write a skeleton for the procedures executed by p1 and p2 to illustrate how they synchronize with each other using P and V.

```
P1( )
{
   while(TRUE)
   {
      P(sem1);
      P(mutex);
      b = produce();
      V(mutex);
      V(sem);
   }
}

P2( )
{
   while(TRUE)
   {
      P(sem);
      P(mutex);
      print(b);
      V(mutex);
      V(sem1);
   }
}

int sem = 0, mutex=1;
    sem1 = 1;
```