Practice Problems Covering Chapters 1 and 2

Choose only one answer from the following

1. If you want to write a program to multiply two matrices in a parallel way. Which method will be faster?
   a. threads  b. processes  c. multiprogramming (e.g. pipes)  d. just serial program

2. Shortest job first is more suitable for:
   a. Real-time system  b. Batch system  c. Interactive system  d. In all of them

3. Which of the following is NOT an advantage of using threads
   a. Makes programming model simpler when many things in an application are going on at once.
   b. They are easier to create than processes
   c. Threads of a program have separate address spaces hence increase protection.

Answer the following questions:

1. Describe the two general roles of an operating system, and elaborate why these roles are important.

   The two roles are:
   • Provide the application programmers with a clean abstract set of resources instead of the messy hardware. This makes programming easier and more portables among machines running the same OS.
   • Managing hardware resources to ensure fairness, protection, and efficient execution.

2. Why don’t we have a finished/killed state for a process?

   A process state is needed in order to schedule that process. It is stored in the process table. However, when a process exits or is killed it won’t exist in the process table anymore and won’t need to be scheduled.

3. What is the relationship between race condition and critical region?

   A race condition may cause potential problems when more than one process is accessing the same memory item(s). Race condition means that the final outcome
depends on which process will do an action first. The part of a program that accesses the share memory is called the critical region. To avoid critical region we must not allow two programs to be their critical regions at the same time.

4. Is an interactive system a real-time system? How about the other way around?

No necessarily. If the system becomes slow in an interactive system, the user may get upset. But if the system misses a deadline in a real-time system, the program may fail. A real-time system may or may not be interactive. For instance, a system that must send a signal every 5 seconds is real-time yet it doesn’t interact with a user.

5. We have seen 3 states for a process (running, blocked, and ready). When drawing the state diagram we found 4 transitions only instead of 6. Why are these two transitions omitted?

The two missing states are:

- Ready->blocked: This is impossible because a ready process is not running so cannot initiate any I/O.
- Blocked -> running: Usually doesn’t happen except in some rare situation in a single process system where a process is blocked on an I/O and the I/O finishes.