CS4320 (Principles of Operating Systems): Homework 3

Assigned: November 7, 2000.

Due: November 15, noon. Not accepted late.

Credit: 30 points.

Reading

Be sure you have read chapters 4 and 5. (You can skim sections 4.6 and 5.5.)

Problems

Turn in hardcopy answers (either handwritten or generated by your favorite word-processing or text-formatting program) to the following problems.

1. (2 points) Chapter 4 distinguishes between "long-term scheduling" and "short-term scheduling". Which of these two operating system functions needs to be as efficient as possible; which can be slower? Why?
2. (5 points) Previously we discussed how to implement semaphores (as a way of coordinating processes' activities — refer to chapter 6), using queues to avoid busy waiting. Each queue might contain zero or more processes that are waiting for the associated semaphore's value to become positive. Chapter 4 describes a scheme for representing and keeping track of all processes in the system, in which each process is either "running", "ready", or "waiting", and queues are used to keep track of "ready" or "waiting" processes. How could we modify this scheme to work well with the queue-based semaphore implementation? In particular:
   - Should we consider the processes in our semaphores' queues to be running, ready, or waiting?
   - What are the possible results of one process's executing a "wait" operation on a semaphore? (Consider the possible effects on the process's own state and also effects on other processes' state, and how we can represent this in terms of queues. For example, when a process issues a request for input/output, its state changes from "running" to "waiting" and it is added to the appropriate I/O queue.)
   - What are the possible results of one process's executing a "signal" operation on a semaphore? (Consider possible effects as for "wait".)
3. (2 points) Exercise 5.1 from Silberschatz and Galvin.
4. (6 points) Exercise 5.3 from Silberschatz and Galvin. (Use the definitions of "turnaround time" and "waiting time" presented in the text.)
5. (4 points) Exercise 5.5 from Silberschatz and Galvin.
6. (4 points) Exercise 5.9 from Silberschatz and Galvin.
7. (2 points) Chapter 5 distinguishes between "internally defined priorities" and "external priorities" (section 5.3.3). Consider the multilevel scheme of figure 5.6 (p. 138), in which the different queues have different priorities. Are the relative priorities of the queues internally defined or external?

8. (5 points) Chapter 5 section 4 briefly describes an CPU scheduling for a system with more than one processor in which each processor is "self-scheduling", and mentions that care must be taken to be sure two processors do not choose the same process and that processes are not lost from the queue.

   - How could these potential problems arise?
   - How could you avoid them?

   (*Hint:* Recall our earlier discussion of material from chapters 4 and 6.)