CSCI 4320 (Principles of Operating Systems), Fall 2008

Homework 3

Assigned: October 6, 2008.

Due: October 13, 2008, at 5pm. *Not accepted late* (except that you may submit the optional programming problem until 11:59pm).

Credit: 10 points.

1 Reading

Be sure you have read Chapter 2, sections 4 and 5.

2 Problems

Answer the following questions. You may write out your answers by hand or using a word processor or other program, but please submit hard copy, either in class or in my mailbox in the department office.

1. (5 points) Five batch jobs (call them A through E) arrive at a computer center at almost the same time. Their estimated running times (in minutes) and priorities are as follows, with 5 indicating the highest priority:

job	*running time*	*priority*
A	10	3
B	6	5
C	2	2
D	4	1
E	8	4

For each of the following scheduling algorithms, determine the turnaround time for each job and the average turnaround time. Assume that all jobs are completely CPU-bound (i.e., they do not block). (Before doing this by hand, decide how much of optional programming problem 1 you want to do.

- First-come, first-served (run them in alphabetic order by job name).
- Shortest job first.
- Round robin, using a time quantum of 1 minute.
- Round robin, using a time quantum of 2 minutes.
- Priority scheduling.

2. (5 points) Suppose that a scheduling algorithm favors processes that have used the least amount of processor time in the recent past. Why will this algorithm favor I/O-bound processes yet not permanently starve CPU-bound processes, even if there is always an I/O-bound process ready to run?

3 Programming Problems

Optionally, do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu`, with each file as an attachment. Please use a subject line that mentions the course number and the assignment (e.g., "csci 4320 homework 3"). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department's Fedora 9 Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (Optional — up to 10 extra-credit points) The starting point for this problem is a Java program that simulates execution of a scheduler, i.e., generates solutions to problem 1. Currently the program simulates only the FCFS algorithm. Your mission is to make it simulate one or more of the other algorithms mentioned in problem 1. The more algorithms you correctly simulate, the more points you will receive. The program consists of several classes, collected in a package called `scheduler`:

   - API documentation[1].
   - Code[2]. (Class `SchedulerTest` contains the `main` method of interest.)
   - Sample input[3].

   Feel free to rewrite anything about this program, including starting over in a language of your choice. Just remember that the program has to run on one of the department Linux machines, and it needs to accept input from standard input — i.e., no GUIs, Web-based programs, etc. The latter requirement is to make it easier for me to test your code, at least partially automatically. If you make changes to the format of the input — and I prefer that you don't — change the comments so they describe the changed requirements.

[1] `http://www.cs.trinity.edu/~bmassing/Classes/CS4320_2008fall/Homeworks/HW03/Problems/scheduler/docs/`

[2] `http://www.cs.trinity.edu/~bmassing/Classes/CS4320_2008fall/Homeworks/HW03/Problems/scheduler/source/scheduler`

[3] `http://www.cs.trinity.edu/~bmassing/Classes/CS4320_2008fall/Homeworks/HW03/Problems/scheduler/sample.in`