Scheduling

Tom Kelliher, CS 411

1 Announcements

2 Discussion

1. turnaround time
2. response time
3. time-slice/quantum
4. FIFO
5. SJF
6. preemptive vs. non-preemptive
7. SRT (preemptive SJF)
8. RR
9. MLFQ
   - Rule 1: `If Priority(A) > Priority(B), A runs (B doesn't)`
   - Rule 2: `If Priority(A) = Priority(B), A & B run in RR`
   - Rule 3: When a job enters the system, it is placed at the highest priority (the topmost queue)
   - Rule 4a: If a job uses up an entire time slice while running, its priority is reduced (i.e., it moves down one queue).
   - Rule 4b: If a job gives up the CPU before the time slice is up, it stays at the same priority level.

   These are the basic rules. They can be exploited, as described in the reading. Preventative measures for the exploitation are described.

3 Assignment

These programs and READMES are in the the Scheduling directory, which you may obtain from upstream.

The program `scheduler.py` allows you to see how different schedulers perform under scheduling metrics such as response time, turnaround time, and total wait time. See the `README.md` for details. Run the program with the `-h` switch to see a list of all command-line options.

1. (Ungraded. A problem of this type will be on the exam.) Compute the response time and turnaround time when running three jobs of lengths: 5, 10, and 15 with the SJF, FIFO and RR (time-slice of 2).
2. (Graded) What happens to response time with RR as quantum lengths increase? Write an equation that gives the worst-case response time, given `N` jobs.

The program `mlfq.py` allows you to see how the MLFQ scheduler presented in this chapter behaves. See the `README.md` for details. Run the program with the `-h` switch to see a list of all command-line options.

1. (Ungraded. A problem of this type might be on the exam.) Run a few randomly-generated problems with just two jobs and two queues; compute the MLFQ execution trace for each. Make your life easier by limiting the length of each job and turning off I/Os.
2. (Graded) How would you configure the scheduler parameters to behave just like a round-robin scheduler?
3. (Graded) Craft a workload with two jobs and scheduler parameters so that one job takes advantage of the older Rules 4a and 4b (turned on with the -S flag) to game the scheduler and obtain 99% of the CPU over a particular time interval.

Submit your answers to the three graded problems as a text entry in Canvas, or upload a Word or PDF document.