CMPSC 440
Operating Systems
Spring 2014

Laboratory Assignment Five: Using Simulation to Evaluate Scheduling Algorithms

Introduction

Modern operating systems normally have a kernel that runs a scheduler. The goal of the scheduler is to pick which process will next run on the central processing unit (CPU). There are a wide variety of scheduling algorithms, such as first-come first-served (FCFS), shortest job first (SJF), round-robin (RR), and priority (PR) scheduling. Since the behavior of these algorithms is workload-driven and they are often difficult to study analytically, operating systems designers may use simulators to better understand the trade-offs in efficiency and fairness. In this laboratory assignment, you will use a scheduling simulator, implemented by Jim Weller, to evaluate several scheduling algorithms.

Using Simulation to Evaluate Process Schedulers

To learn more about the simulator that we will use to study the scheduling algorithms, please visit the following Web site: `http://jimweller.com/jim-weller/jim/java_proc_sched/`. Next, use your terminal window to navigate into the version control repository that is used for sharing source code and type the command `git pull`. Please change into the `lab5/jimweller/` directory and study the files that are now available. How many files are there? What is their purpose?

The file called `jobList.dat` contains three lines that specify the details about the execution characteristics of a process to be scheduled by the operating system. For instance, the first line in the file contains three numbers, "10 5 1", with the first value corresponding to the amount of CPU time required by the process and the second and third numbers denoting the time that separates the arrival and the priority of the process, respectively. If you carefully study `jobList.dat` you will notice that the first simulation will only run a total of three processes.

To run the simulator, you should type `java -jar cpu.jar` in the terminal window. What do you now see on your screen? To run a simulation, you must first load the `jobList.dat` file. Then, you should pick a scheduling algorithm, indicate the speed at which the algorithm should be animated, and check whether or not the algorithm should consider prioritization or preemption. Now, you can click the check box to run the algorithm. (Please note that the "check box" is really a toggle button that does not render correctly on Ubuntu. If you click this button and the simulator does not run as anticipated, then click it again and see if this starts the animation.)

Once you have finished a run of the simulator, you will observe that it reports a wide variety of evaluation metrics. During this laboratory assignment, we will primarily focus on the mean values for response, turnaround, and wait time. Ideally, a scheduler will run a workload of jobs in a manner that yields low values for all of these metrics. Of course, it is also important to find a scheduler that executes processes in a manner that leads to low standard deviations for these evaluation metrics. What were the metric values for one of the runs of the simulator?

In this laboratory assignment, you will use the scheduler simulator to evaluate the effectiveness of the first-come first-served, shortest job first, round-robin, and priority scheduling algorithms. In the first phase of your study, you should develop a better understanding of the algorithms and the simulator by using the `jobList.dat` file provided in the version control repository. During all of your experiments, you should record the mean values for response, turnaround, and wait time and note whether the standard deviations are high or low. You also should run every algorithm with and without either preemption or prioritization. When you are done with this study, you must be able to explain which scheduling algorithm is best suited to run this listing of three processes.

After you have finished this preliminary study, you need to create a job list that contains at least ten to twenty process descriptions. Then, you should repeat the study again, using all combinations of algorithm and scheduling preference. Again, you should aim to determine which algorithm is the best and explain how you arrived at this conclusion. Finally, you should use the random workload generator provided by the scheduler to study at least five additional job lists. Students who would like to speed up the simulation time are encouraged to try a higher value for the frames per second (FPS) setting. As in the previous experiments, you should try to draw a conclusion as to which algorithm is the best and be able to, whenever possible, explain why it is the best.

After you have finished running all of the simulations, you must prepare a comprehensive report of your findings. Using tables of data and paragraphs of results analysis, the report must highlight and discuss all of your experimental results. In advance of talking about your findings, the report should clearly explain how each of the scheduling algorithms actually works. Also, the report should formally define the evaluation metrics and discuss the threats to the validity of the simulation study. Students are encouraged to write their reports in the LaTeX text formatting language and use the R language for statistical computation to visualize and analyze their results.

Summary of the Required Deliverables

This assignment invites you to submit printed and signed versions of the following deliverables:

1. A screenshot demonstrating that you were able to correctly configure and use the simulator
2. All of the `.dat` files used in the simulation studies that you conducted your experiments
3. A comprehensive report on all of the runs of the scheduling simulator that includes:
    (a) A clear description the behavior of each of the scheduling algorithms
    (b) A commentary on how preemption and prioritization influences scheduling decisions
    (c) A formal definition of the evaluation metrics used in your study
    (d) A description of the job workloads used during the simulations
    (e) A detailed analysis of the results from running the scheduler simulations
    (f) A statement of which scheduling algorithm is best suited for a modern operating system
4. A reflective discussion of the challenges that you encountered when completing this assignment

In adherence to the honor code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. As such, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.