California State Polytechnic University, Pomona

Process Schedulers Comparison

Project 1

Joshua Gibbs
CS 431 - Operating Systems
Dominick Atanasio
11 February 2018

Intro

In this project, we compare five different process schedulers or configurations of those schedulers. The program is broken up into four different class files, Menu, ProcessScheduler, ProcessLog, and Process. Our starting point is in Menu, where we ask the user for input data to test with, load said data, clone the data for each scheduler, invoke the scheduler to process our data, and then output information and averages. When loading our data from a text file, we create a new instance of our Process class. The Process class contains process information such as PID, priority, burst time, and end time. It also contains a “compute” method which takes inputs for the current cpu time, and an optional quanta. This sets the new remaining burst time appropriately, decides if the process has finished running, and returns the amount of CPU time used. In addition, it also logs everything that happened for output in our CSV files. This is done using the ProcessLog class. After creating an array of processes from our text file, the array is sent to each process scheduler, where each process is computed until completion.

Analysis

The first process scheduler algorithm we used was the First-Come-First-Serve (FCFS) scheduler, which works by completing each process without pause in the order it came. The next algorithm used was Shortest-Job-First (SJF), which works the same as FCFS, except the process queue is sorted by shortest burst time instead of arrival. Round-Robin, our next

algorithm, works by computing each process for a fixed amount of time before moving to the next one. Processes are worked on in the order they arrived in the queue. The only difference between this method and our last method: Lottery Scheduler, is the way the next process is picked. The Lottery Scheduler picks the next process randomly, but giving processes with a higher priority a better chance at getting picked.

In all four test data sets, the FCFS algorithm performed the worst, taking on average 8771 CPU cycles for our largest 27-process data set. The next worst performer was SJF, which averaged 5685 cycles on the same data. When we begin computing processes for a fixed quanta, our average completion times drop by a minimum factor of five. Round-Robin with a quanta of 50 averaged 983 on our largest data set, followed by the Lottery scheduler with a quanta of 50 with an average of 809 cycles. By far the best performer of all five tests was Round Robin with a quanta of 25. This algorithm was able to complete each task in an average of 526 cycles; over 16 times better than FCFS.

Conclusion

For any sized data set, algorithms which broke computation up into several cycles, effectively taking turns with other processes, came out far ahead of those which completed entire processes as they received them. Of the five algorithms used, Round-Robin performed best with a quanta of 25.