CSCE 4600.021

Project 1

Due June 30, 2016

Members: Alexis Kunkel, Charles Neeler, Margarita Sanchez

This report will evaluate performance and differences between the round-robin, FIFO and SJF scheduling algorithms. Round robin scheduling, also known as cyclic executive, is when a time slice (quantum) is assigned to each process in equal portions and in circular order. Round Robin scheduling does not have any priority but can result in starvation. First- in- first- out (FIFO) scheduling queues processes based on the order they appear in the ready queue. Shortest- Job- First (SJF) scheduling assigns available CPU to the process with the smallest estimated run time or smallest next CPU burst. SJF needs to know how long a job or process will run which is not always provided.

Each of these scheduling algorithms has a differentiating quality that helps determine which is “better” for different situations. The Shortest- Job- First algorithm generally is the fastest as this algorithm is designed to handle maximum throughput and to service small processes before servicing large processes. The Round Robin algorithm generally works best with a shorter list of processes. The First- In- First- Out algorithm is the simplest to understand and implement.

At a quick glance of the following table one can see that for our tests the Shortest- Job-First algorithm was the fastest for a single processor while the First- In- First- Out algorithm is the fastest when using multiple processors.

|  |  |  |
| --- | --- | --- |
| Overall Comparison of Wait Time | | |
|  | Single Processor | Multi-Processor |
| Round Robin | 189926.4 | 17528.32 |
| S.J.F | 160016.7 | 151105.6 |
| F.I.F.O. | 170565.7 | 40424.2 |

The following graphs give an in-depth look at the wait time of each algorithm with different number of processors.

**First- In- First- Out (FIFO)**

**Shortest- Job- First (SJF)**

**Round Robin**

**First- In- First- Out with Multi-Processors**

**Shortest- Job- First with Multi-processors**

**Round Robin with Multi-processors**

When running these same scheduling algorithms with a multiprocessor system with four processors we should be able to see the different ways varying algorithms are effected. Each of the scheduling algorithms run on each of the four processors. In a sense, the multiprocessors spilt the work that the single processor was doing which should make the processes run faster because of increased through-put. The round robin algorithm through multiple processors will help reduce the total waiting time but this method is not the “best.” Round robin algorithm through multiple processors allows some processes to run at the same time since they are given the same time slice. The First- In- First- Out algorithm through multiple processors allows the increased through put still with no sense of prioritization. This design does allow for a reduced total wait time. The Shortest- Job- First algorithm through multiple processors does still optimally provide a shorter wait time as the processes are divided between multiple processors.

The use of multiple processors has helped decrease the average wait time as depicted above. Multiple processors reduced the wait time as much as ten times less, specifically when using the Round Robin algorithm. The Round Robin scheduling algorithm had the largest ratio in reduction when multiple processors were introduced. Other scheduling algorithms were effected by at least a 1/10th reduction.

Overall adding multiple processors helps reduce the wait time. The most “optimal” algorithm for multi-processors as well as single processors remains the Shortest- Jon- First algorithm. Shortest- Job- First and First- In- First- Out algorithms produce marginally different data for both a single processor and multiple processors.