|  |
| --- |
|  |
| User Manual |
| CS 450: Assignment 1 |
| **Andrew Ribeiro** |
| **12/7/2009** |
|  |

|  |
| --- |
| This user manual guides you through operating the analytic evaluation program. In addition, this guide helps one analyze the results of the program. |

Using The Program

The main menu has two options:

1. Single algorithm performance evaluation: this option allows you to test a single algorithm.
2. Algorithm comparing: this option allows you to test multiple algorithms at the same time with the same test data.

## Single Algorithm Performance Evaluation

In this section, three algorithms are presented. Choice an algorithm of your choice to run a simulation with that algorithm. After you run the simulation, you are asked if you would like to run another simulation with a different algorithm, but with the same test data; If you wish to run another algorithm press 1, if not press a number other than 1 (i.e. 2).

After you complete your tests, you will also find a text file with the name of the algorithm in the same folder as the program. This file contains relevant test data and a printout of the results summary. Please see the Results section of this guide for more information about understanding the contents of these files.

## Algorithm Comparing

In this section, three algorithms are also presented. One chooses algorithms by typing the algorithm numbers that they would like to compare, separated by a space, and at the end one presses Ctrl + Z. For example, if you wanted to compare algorithms one and two, you would enter this: 1 2 Ctrl+Z.

After a choice is made, the results are displayed to the screen and are printed in a file called algcmp.txt in the same folder as the program. After the results are displayed, press enter to return to the main menu.

Results

The results produced by this program may be confusing. However, this guide should help you understand what the results mean. There are two types of results: results printed to the console and results printed to file. The results printed to the file include the results printed to the console, but have more information for analysis.

## Console Results

The typical result one gets when they run a simulation looks like below:

\*\*\* SJF(Preemptive) RESULTS \*\*\*\*

\* Number of processes: 20

\* Average wait time: 135 cycles

\* Average burst time: 25 cycles

\* Average priority: 2 cycles

\* Total cycles: 513

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Here is the explanation of each line in the results printout:

* **The algorithm name**
* **Number of processes**: this is how many processes where handled by the CPU simulator during this trial.
* **Average wait time**: this is how long each process had to wait before the CPU started processing it. This is the most important statistic, because it demonstrates the efficiency/responsiveness of the algorithm. This is the number you will be comparing in your analysis.
* **Average burst time**: this is the average size of the process burst times. This statistic is only to show that data sets are similar when comparing multiple algorithms.
* **Average priority**: this is the average priority of each of the processes. This statistic is only to show that data sets are similar when comparing multiple algorithms in this program ( because none of the algorithms utilize priority).
* **Total cycles**: this is how many cycles it took the CPU to process all of the processes.

## File Results

The file results contain the additional information:

* A cycle by cycle printout: shows the current cycle, the process that is being worked on, the amount of the process that has been worked on, and the arrival time.
  + Syntax: Cycle(number): ProcessName(current work out of total work) Arrival time
  + Example:
    - Cycle(5): Process15(1 of 35) 5
    - Cycle(6): Process16(1 of 11) 5
    - Cycle(7): Process1(1 of 4) 7
    - Cycle(8): Process1(2 of 4) 7
* A printout of the ready queue contents: this applies to the SJF algorithm. This is a printout of the contents of the ready queue when the contents are sorted according to the burst time.
  + Example: 43 56 78
    - This means that the ready queue contains the burst times of 43 56 and 78. The first burst is the current burst that is being worked on.

Understanding The Algorithms

**First Come First Serve** (*FCFS*): processes CPU bursts according to the time in which they arrived at the ready queue. So if process 1 came into the ready queue at time 0 and process 2 came into the ready queue at time 1, process1 would be processed by the CPU first.

**Shortest Job First** (*SJF*): processes CPU bursts according to the length of the burst time. The process with the shortest burst time is processed first. There are two different variations of SJF:

* **Shortest Job First Preemptive** (*SJFP*): if the CPU is working on a process and a shorter job comes into the ready queue, the CPU will switch to the shorter job the next cycle.
* **Shortest Job First Non-Preemptive** (*SJFNP*): if the CPU is working on a process and a shorter job comes into the ready queue, the CPU will finish the current job before processing the next shortest job.