

Project Specs

Saturday, September 30, 2023 12:42 PM

Description:

Given a set of (simulated) "jobs", run the following Job Scheduling Algorithms and evaluate the performance of each.

(1) FIFO

(2) Shortest Job Algorithm (non-preemptive)

(3) Shortest Remaining Job Algorithm (pre-emptive)

(4) "Highest" Priority Algorithm (assigned)

(5) Round-Robin: with and without context switch

All algorithms should be analyzed by collecting

- turn-around time

- throughput for a fixed length of time.

Output should include: a table

(similar to the "quiz" output table)

Showing values for each algorithm;

also calculate the average-turnaround time

also calculate the average-turnaround time for each algorithm. and through-put.

The jobs will be randomly generated and stored as "job Objects". Each jobObject will include:

- arrivalTime
- CPUTime
- priority
- remaining time

Your program will generate 25 jobs, simulate (run) the algorithms and display the original job information and the table.

Documentation should include

- data structures used and why
- "runtimes" (Big-"O") for the code
- summary of your results
- appropriate usage of each algorithm.

I am posting pseudocode to generate the 25 jobs.

generate the ^u₂₅ jobs.

C++ or Java are fine. No Python!

Job Object Class

Saturday, September 30, 2017 1:08 PM

Attributes (make "friendly"/"protected")

- arrTime - random 1-250
- cpuBurst - random 2-15
- priority - random 1-5 (5 is highest)
- exitTime - algorithm result
- turnAroundTime - algorithm result
- remainingTime - updated in algorithms

Methods:

- constructor(s)
- toString (to display initial jobs and algorithm results)
- compareTo (general, by arrival time)
// Note: different algorithms may need a different compare, i.e. by priority, etc.

Those can be defined in the algorithm methods.

algorithm methods.