Project - CPU Scheduling

Due Friday 28 April 2023

Description:

One of the main tasks of an operating system is scheduling processes to run on the CPU. In this assignment, you will build a program that simulates three CPU scheduling policies discussed in class. In general, your simulator should select a process to run from the ready queue based on the scheduling algorithm chosen at runtime. Since the assignment intends to simulate a CPU Scheduler, it does not require any actual process creation or execution, but only simulated events of process arrival and execution. When an event occurs in your simulator, it will simply print out the nature of the event and the process that was involved in that event. For more details on Discrete event simulation

Design Details:

1. Write a class called Process which stores the ID, arrival time, and CPU burst length of a process, all are integers.
   Please note:
   a. Process IDs must be unique.
   b. Arrival times and CPU bursts don't have to be unique.
   c. one cycle will be lost
   d. You can also add properties to a process that will help you compute the statistics about that process such as its wait time, response time, and turnaround time.
   e. The methods of the Process class are the get and set methods for each data member, the constructor for the class, and any method needed to compute the statistics for that process.

2. Write a class called **Generator** to generate a list of objects of type Process, representing the CPU processes. Note that all the properties of a process are integers, so generate random processes accordingly.

3. Write a class called Scheduler to simulate a CPU scheduler for an operating system. The scheduler contains the ready queue, which is a queue of limited capacity. Note that your method of adding and removing processes to that queue will depend on the scheduling policy implemented.

   You'll implement **First Come First Serve, Shortest Remaining Time** First which is the preemptive version of Shortest Job First, and **Round Robin** scheduling policies.

4. Create a driver class that will start the simulation and print results into a results file named results.txt.

   a. The input to your program should be the total simulation time, the average interarrival time between processes, and the quantum to be used in the Round Robin scheduling policy.