

Project Description:

You will design an inheritance hierarchy for your scheduling algorithm. Your base class will be Schedule. You will have three derived classes. Each of the derived classes will represent the Shortest Job First (SJF), Priority, and Round Robin algorithms. Any common methods or data members used should be in the base class. If methods are to be overridden in the derived classes, the methods should be virtual.

For each algorithm you will need to simulate a set of processes in the ready queue. For each process, you will maintain a Process Control Block with at least the following information:

```
int pid;           //process ID, may be randomly generated within some reasonable range
int proState       //state of the process (ready, running, waiting)
int cpuTime        //CPU Burst time (next burst time, may be randomly generated)
int arrTime;       //You can determine Arrival time as all 0, however, allow for various arrival times
int waitTime       //calculated
int respTime       //calculated
int turnTime       //calculated
```

You may add other attributes if it fits your routines.

Driver: the driver should create the scenario for the various scheduling algorithms to run against each other. When a ready queue is simulated, each algorithm will simulate scheduling the queue. After each data set is processed, your program will create a table showing the following:

1. Average Wait Time
2. Average Response Time
3. Average Turnaround Time

Your driver should be able to generate a reasonable set of test cases to both ensure the algorithms are correct and to be able to assess how the different algorithms behave with different data sets. You will be evaluated for how you test this project.

Workload: This is a group and an individual assignment.

Group responsibilities: Design and implement the inheritance hierarchy and a driver to execute the schedulers to compare them against each other. As a group, you should create the class headers (prototypes) for each algorithm. Any implementation required for the Base class should be completed by the group.

Individual responsibilities: After the design and implementation of the Base class, Derived class stubs, and the driver, implement the algorithm you are assigned. This must be an individual assignment. If a group member fails to implement an algorithm, the group will present with empty stubs for that algorithm.