

This assignment has two parts:

- 1) Scheduler implementation in Java and 2) Experimentation, using a simulation program.

MAIN TASKS

- 1) Read up on and implement basic metrics for scheduler evaluation by completing **Process.java**

- a. turnaround time: `getTurnaroundTime()`
- b. waiting time: `getWaitingTime()`
- c. response time: `getResponseTime()`

In this simulation framework, a process is ready from the moment it arrives.

- 2) Read up on and implement four schedulers

- a. Round Robin

Round Robin RRScheduler.java

- b. Ideal Shortest Job First (no prediction necessary)

IdealSJFScheduler.java

- c. Multi-level feedback queue with Round Robin

FeedbackRRScheduler.java

- d. Shortest Job First using exponential averaging

SJFScheduler.java

First-Come-First-Served (`FcfsScheduler.java`) is given to you as an example.

- 3) Design and conduct experiments

This is a valuable general skill that you will come to re-use in future modules (and in any future career requiring analytical ability).

You need to perform **three different** experiments, each investigating the performance of one or more schedulers, given specific inputs.

Make sure your hypotheses have a narrow, yet informative scope (i.e., you need to be able to learn something from the results).

For example, study the **effect of varying a single parameter on a single algorithm** while keeping the others stable. If you vary too many parameters at a time, it might be difficult to link cause and effect.

Alternatively, you could **compare the performance of different algorithms given a particular input** (e.g., a majority of CPU-bound processes, many/few).

Run each experiment several times using multiple, slightly different, input files in order to achieve better reliability of results (average the differences in output for the presentation).