

Scheduler

Introduction

Johnny is undertaking his undergraduate Data Structures course in the current semester. He is curious about the differences/ challenges of a Queue, LinkedList, and Stack when only one can be used to solve a single problem.

Can you use an example to show him the differences between them?

Requirements

You must build three scheduling systems (A, B, C) to show Johnny the differences between a Queue, LinkedList and Stack.

For each of the scheduling systems, you are only allowed to use a single type of data structure, as defined below:

1. Queue – Scheduling System A
2. LinkedList – Scheduling System B
3. Stack – Scheduling System C

You will be given a *tasks.txt* file that contains a list of tasks that will be arriving to your CPU at the same time. An example of *tasks.txt* is shown below:

```
                                tasks.txt
isPrime int 2147483640
fib int 12
fib int 39
getNthUglyNo int 149
longestPalSubstr String helloolleh
isPrime int 10
sumOfDigitsFrom1ToN int 101
```

Each line of *tasks.txt* consists of **three** parts: method **name**, **input type** and **input**. The methods' definitions are given in *StarterPack.java*.

Your primary task is to read *tasks.txt* and fetch the data into the scheduling systems, then use a suitable methodology such as First In First Out (FIFO), Last In First Out (LIFO) and so on to execute the tasks.

While executing the tasks, you need to keep track of the time to produce a report with each scheduling system's average response time and turnaround time.

Response time is the time elapsed from the time of task arrival until the task is executed.

Turnaround time is the time elapsed from the time of task arrival until the time of task completion.

Some Ideas to Explore (Bonus marks! 🤖)

1. Optimizing the scheduling system*
2. Give me some “WOW” factor

*Tips: You might use certain scheduling algorithms, such as Shortest Job First (SJF), to improve the performance of the scheduling system, but first, you need to know which method or what input will result in a shorter execution time. To know this, you have no choice but to study the methods implemented in *StarterPack.java*.

Project Marks / Rubrics

The following marks serve as a guideline:

Mastery of data structure: 20%

- Showcase structure development – data, methods, constructors, and any required components to successfully implement the structure.

Fulfilment of question’s requirements: 15%

- Showcase workable traversing/searching of structure – 5%
- Showcase workable element/data editing (adding, deleting, sorting, etc.) – 5%
- Showcase correct result – 5%

Bonus marks (optimisation, wow factor): 10%

- GUI has some merit.
- Less popular project has some merit.

Submission of 5-minute Video Presentation: 5%

- Showcase mastery of data structure.
- Showcase project requirements achievement.

Submission of Project report – no marks but required for formalities. Can be a 2-page report describing your group member details, 1 paragraph describing project selection (why you chose the project), and 1 paragraph reflecting the challenges faced during the development. The report can be Word Doc, PDF or PPT slides (as long you can submit through Spectrum).

Project Deadline – will be published on Spectrum.