

[**CS301\_SEP2021 - Operating Systems**](https://blackboard.buid.ac.ae/webapps/blackboard/execute/courseMain?course_id=_854_1)**- Group project:**

**CPU Scheduler Algorithms**

|  |  |
| --- | --- |
| Name: | student ID |
| Mohammad Adnan | **20195481** |
| Jamila Ibrahim | **20194192** |
| Razan Asaad | **20196201** |
| Noor Hawari | **20195745** |

* **Brief:**

CPU Scheduling is the process of selecting which process will have exclusive use of the CPU while another is paused. The basic goal of CPU scheduling is to ensure that anytime the CPU is idle, the OS chooses at least one of the programs in the ready queue to run. Process scheduling is used by an operating system to guarantee that processes run smoothly and with minimal wait periods.

This project is about implementing the algorithms of CPU scheduling, the implementation is done by using java code through Eclipse, **Division of Work:**

The division of jobs between individuals and cooperation among them is the basis for the success of any project or specific work in the team and to reach the goal they seek and the ability to complete the work quickly. Therefore, after we read what is required in our project and understand it well, we divided the work based on the scheduling algorithms:

Mhd Adnan

Round robin and GUI

SJF

Jamila ibrahim

FCFS

Noor hawari

Priority

Razan Asaad

* **Scheduling Algorithms:**

The Algorithms we chose to implement are

**First come First serve (FCFS):** The first come first serve algorithm is an algorithms that executes tasks that are queued based on which task arrived first

The order that the processes arrive at are P1 -> P2 -> P3 -> P4

|  |  |  |
| --- | --- | --- |
| **Process** | **Burst Time** | **Priority** |
| P1 | 12 | 1 |
| P2 | 4 | 4 |
| P3 | 6 | 2 |
| P4 | 2 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |

**Shortest Job First (SJF):** Shortest Job first algorithm is an algorithm that queues tasks and executes them based on The burst time, The one with the shortest burst time will get executed first.

|  |  |  |
| --- | --- | --- |
| **Process** | **Burst Time** | **Priori ty** |
| P1 | 12 | 1 |
| P2 | 4 | 4 |
| P3 | 6 | 2 |
| P4 | 2 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| P4 | P2 | P3 | P1 |

**Priority Scheduling:** Priority scheduling algorithm is an algorithm that selects the task based on their priority, Tasks are executed in order of most important or highest priority to least important or lowest priority

|  |  |  |
| --- | --- | --- |
| **Process** | **Burst Time** | **Priority** |
| P1 | 12 | 1 |
| P2 | 4 | 4 |
| P3 | 6 | 2 |
| P4 | 2 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P3 | P4 | P2 |

**Round Robin (RR):** Round Robin scheduling algorithm is a type of algorithm that assumes that all the tasks are equal, and allocates an equal amount of cpu time for each task until all the tasks are done being executed. Time Quantum = 2

|  |  |  |
| --- | --- | --- |
| **Process** | **Burst Time** | **Priority** |
| P1 | 12 | 1 |
| P2 | 4 | 4 |
| P3 | 6 | 2 |
| P4 | 2 | 3 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P1 | P2 | P3 | P1 |

|  |  |  |  |
| --- | --- | --- | --- |
| P3 | P1 | P1 | P1 |

* **Implementation Method:**

To implement our idea We chose to use JAVA as our programming language, we used JAVA because of several reasons, Firstly we are all familiar with JAVA and have done many projects using the language, Secondly, JAVA is object oriented which suited this project as suppose to C which is procedural Oriented language

* **Results & Analysis:**

Before we started to code our projects, We solved all of the sample task files provided to us with all four algorithms to later compare the results with the output program that we are going to write, We drew the gantt charts to make sure that our results were correct, After we finished coding the program we ran all the files on all the algorithms and compared the output with the results we got from us scheduling the tasks, The output of the program was the same as the results we got when we scheduled the tasks, this means that our program is working correctly

* **Interesting findings:**

We found that implement the FCFS, SJF and Priority scheduling algorithms was fairly simple, it was a matter of sorting depending on either arrival or burst time or priority, but the hardest was Round Robin as we really had to think outside the box in order to implement and make it work correctly.

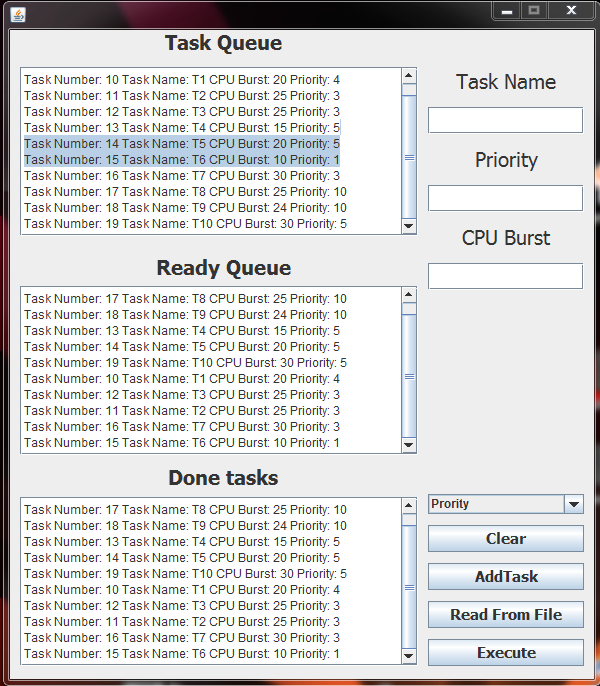
* **Comments on the results and the Whole experience:**

One interesting idea that we wanted to implement but we didn’t get to was that we wanted to sort the tasks using different soring algorithms and see what difference it makes, from bubble sort, merge sort, quick sort, but due to the time constraints we were not able to.

* **Appendix:**

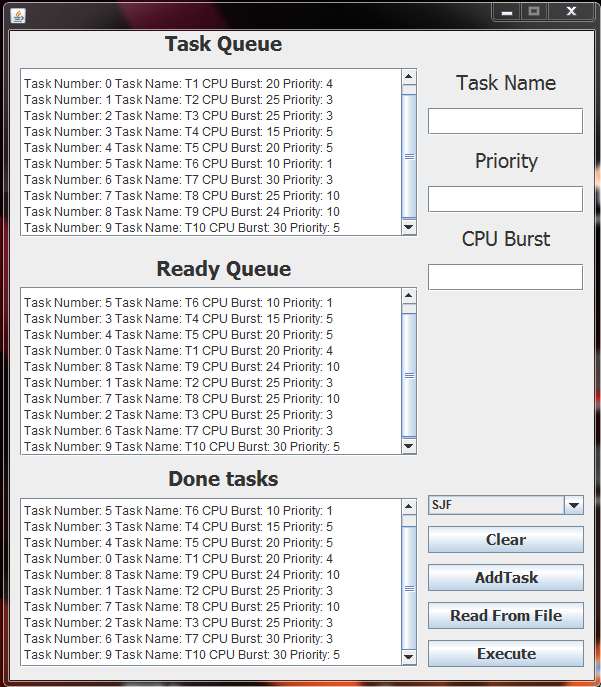
**Priority algorithm:**

Task Queue are the tasks printed from the text file, then when the program sorts them it prints them in the ready Queue Sorted, After that each task that is done executing gets printed in the Done tasks



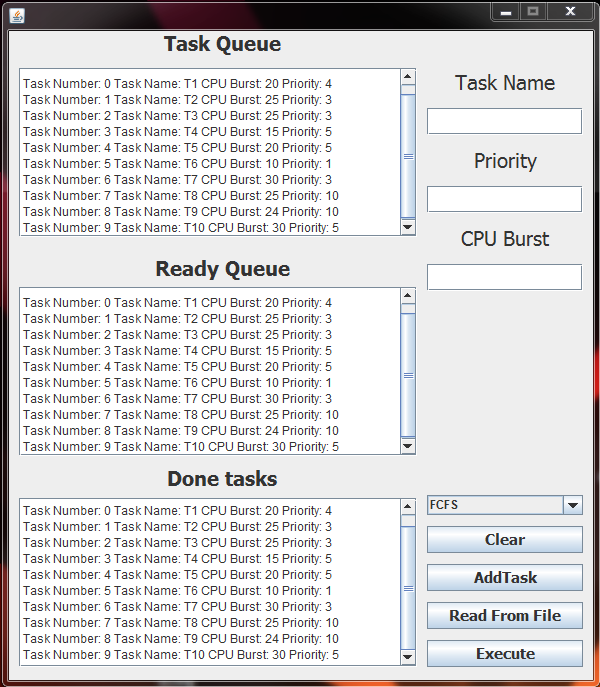
**Shortest Job First Algorithm:**

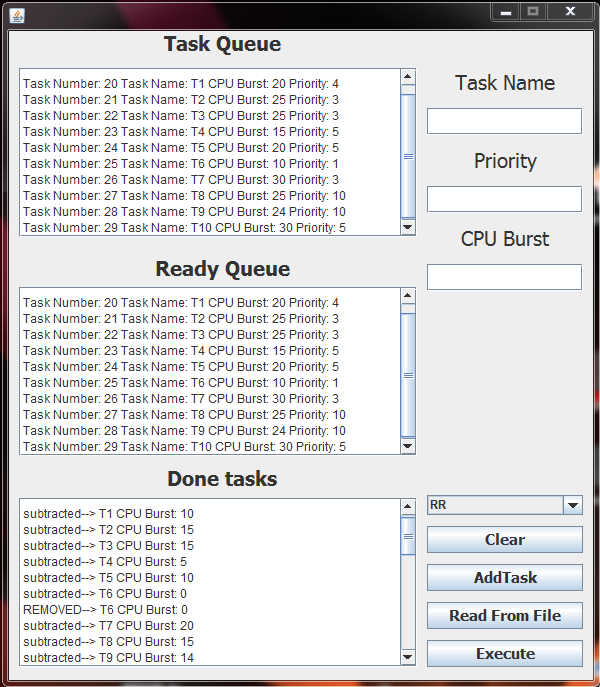
Similar to the priority The tasks are printed from the file to the Task Queue, After than when the user clicks, Execute, The program sorts them and puts them sorted in the ready Queue, After than it starts to execute the tasks and the duration the task takes depends on its Cpu burst value, The larger the value the longer it will take, After a task is done, It gets put in the Done tasks

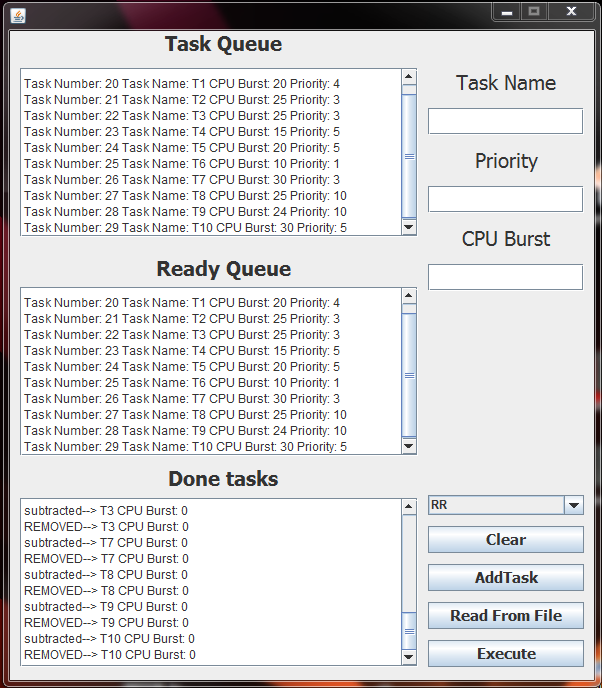
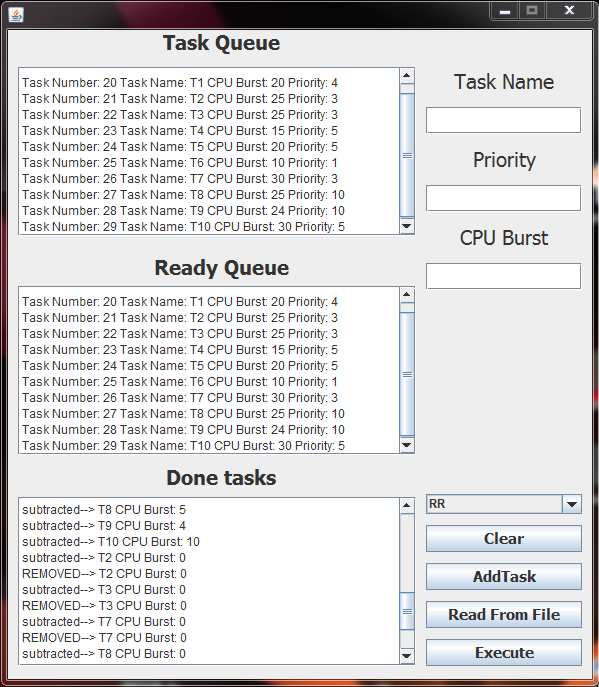
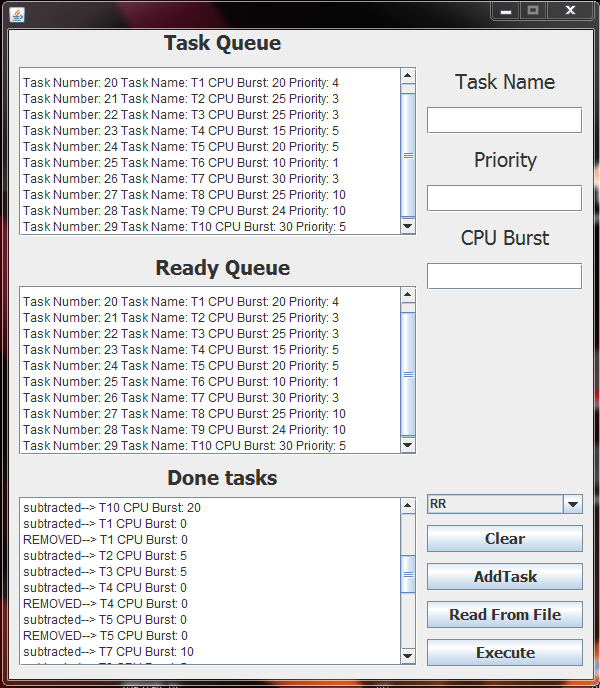


**First Come First Serve:**

Similar to the SJF, The program read the Task file and puts the tasks in the Task Queue, Then when Execute is clicked, it sorts the tasks depending the arrival then it prints them in the Ready Queue, After that when the tasks are finished executing, They get put in the Done Tasks

**:**

**Round Robin:** For the round Robin algorithm, After the program reads the tasks from the file and puts them in the Tasks Queue, When Execute is clicked, Its puts them in the Ready Queue, without any sorting as this is not Priority Round Robin, After than Each Task is allocated 10 milliseconds, and when done, The program prints the Task Name Ex: t1 ,t2 ,t2 And the value of the CPU Burst that is left after processing for the Quantum time, Then when the Task reaches 0 CPU Burst meaning it is done processing, The program will print that it is removed it from the Queue, and will continue processing until all the Tasks are Done. ****

****

**First come First serve (FCFS):**

****

**Shortest Job First (SJF):**



**Priority:**



**Round Robin (RR):**

