جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Operating Systems

Instructor: Dr.Yahya Adnan

Name: Issa Zeidan Number: 1160090

Name: Adham Tamimi Number:1162429

Project 1 Report

Abstract:

In this project we implemented a simulator for some CPU scheduling algorithms. We implemented 5 algorithms which were **FCFS**, **SJF**, **SRTF**, **RR**, and finally the **Priority** algorithm. We also made a beautiful and simple GUI design .

Theory:

The 5 scheduling algorithms are:

1) SJF: We got this algorithm from the equation that was given to us by our instructor which is (My ID + Partners ID)%5, there was 5 algorithms and we got the SJF which was number 4 in the list, and it works as follows. If the CPU is available is checks all the jobs that are ready in the queue and chooses the job with the shortest CPU burst time.

2) RR: The round robin algorithm is one of efficient and commonly used scheduling algorithms and it works as follows. There is something called **Quantum Time** and its usually set to be more 80% of jobs burst time (Best possible QT), CPU chooses the jobs depending on their arrival time (priority can be added too), and gives time an amount of time in the CPU which is the QT, when the time is over if the job has finished then it’s terminated else its sent back to the ready queue and the next job is sent to the CPU for the same amount of time.

3) Priority: This algorithm works on priority, every job has its priority and it takes the highest priority job first, finishes it and then continuous to the next highest priority job, first job is usually the

first job to arrive, also it has aging system which means that the longer the job stays in the ready queue the higher it’s priority becomes, so there will be no starvation.

4) SRTF: This is an extra scheduling algorithm which we implemented and it works as follows. The shortest remaining time first is the only algorithm that we have with preemption, now the first job to arrive usually enters, and if another job comes and it has a less CPU burst time then the system interrupts the current job and puts the one with the least remaining time instead.

5) FCFS: Another extra scheduling algorithm which works as follows, simply the first job to come the first to enter the CPU, and this is the simplest scheduling algorithm as far as we know.

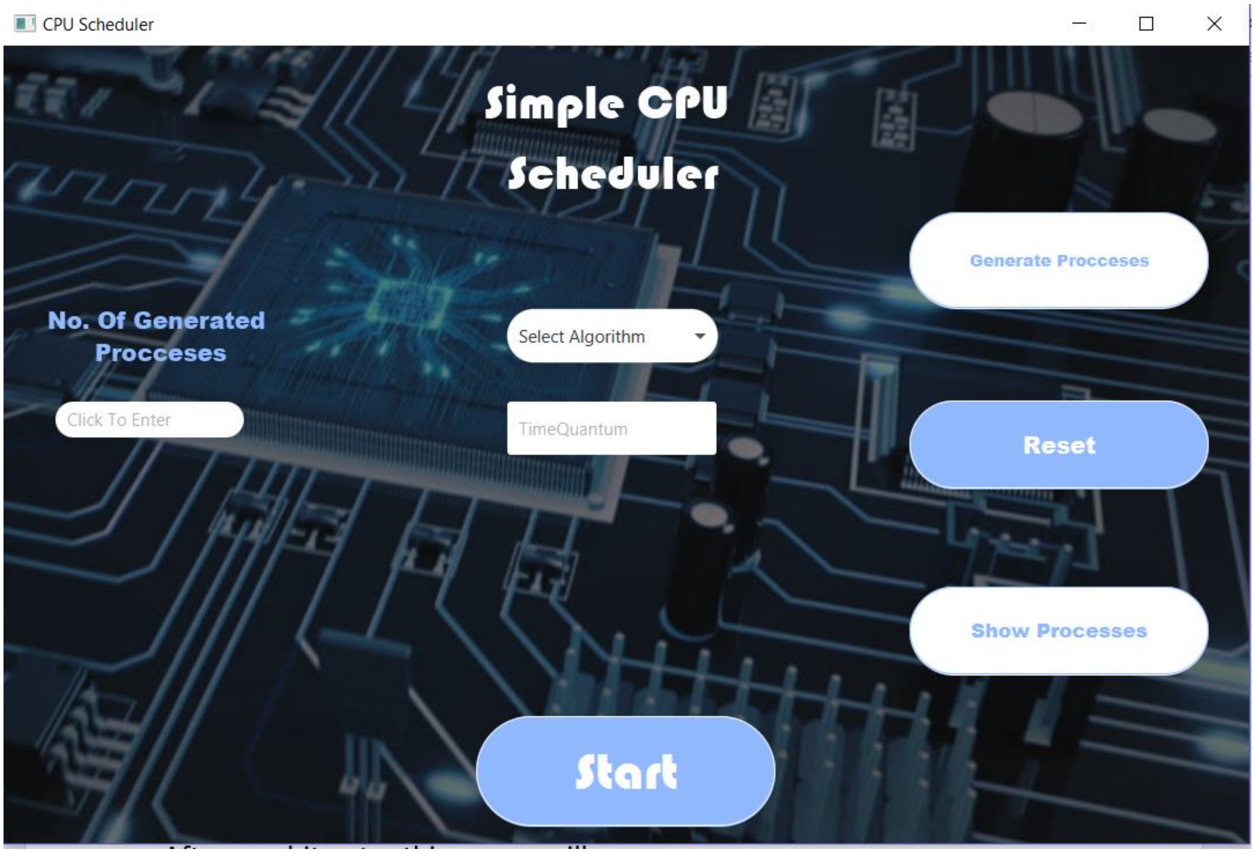
Program Implementation:

We used the java language for our project, we made 7 classes (4 of them were for the scenes using the Scene Builder program), the other 3 classes are Main, Scheduler, and Process. The process class was to create the processes in it and set a random value for their CPU burst, arrival time, a unique ID for each process, and all other things needed for the scheduling algorithms that we have. As for the scheduler class it contains all the scheduling algorithms in it as well as the Gantt chart which we were asked for. And finally, the Main class which shows the scene.

How to Run Our Program:

We have jar file including all the classes + the report, simply open the jar file , a simple screen will appear, simply click Enter if you want to start it or click exit if you want to exit.

After you hit enter this screen will appear:



Now as you can see here, we have a text box to fill the number of processes you want to generate, enter how many processes you want to generate and click on Generate Processes, this will generate a number of processes (the number you put in) with random each with a unique ID and a random arrival time, CPU burst, and Priority. Now if you wish to see these processes you can click on Show Processes which open a new screen with the info of the processes. Next is the scheduling, you can select the algorithm that you want through clicking on Select Algorithm, and then hit start to schedule them.

Note: You cannot enter a value inside the Time Quantum until you choose the Round Robin (RR) as your scheduling algorithm

Note: Do not start the program without generating processes first because it will not work (It won't do anything simply).

Now after you hit start a table will appear showing the process IDs, Burst time, Start time, Turnaround time (TA), and Wait time listed.

If you wish to see the Gantt chart click the button called **Chart** and it will appear.

Each box inside the Gantt chart is 1 tick (1ms) and if there is no job at a certain second it will show it as (**NOP**)

Last thing, if you want to choose another scheduling algorithm click on the **Back** button and then click on **Reset** , then repeat same process.(generate processes, choose algorithm ….etc.).

Conclusion:

As you can see here, this project shows a number of the scheduling algorithms, and it allows us to test them with any number we want of process which random values, and from it you can see which algorithms gives the best average TA time and which and waiting time. As we have noticed so far, the RR algorithm did the best. Also the FCFS algorithm is good with processes that have long burst time. This project wasn't easy to make but still we had fun making it and it taught us a lot of things about the scheduling algorithms. And this can be also used a simulator before applying any algorithm to real life.