

Project 4: CPU scheduler

In this project we are implementing 3 scheduling algorithms. First come first serve, priority, and round robin algorithms. We used an add function from schedulers.h on all 3 algorithms. This function takes in the name of the task, its priority, and burst time. It will assign attributes to the structure we declared. We also need to memory allocate space each task from the text file. Then, we insert the tasks into the list.

For the first scheduling algorithm, the first come first serve, we iterate through the whole list of tasks and reads the item. If head is not null then we pass into the task and burst time into the run function. Delete function will just delete the address of the previous task.

In our 2nd algorithm, priority, we have the same add function. First we made 2 nodes, one for current task and another node for our next task. Our scheduling function we loop through the whole list of tasks while check if the current task is great than the next task's priority. If it is then we make the current task our new task. Else, we move to the next task, pass in the next task and it's burst time into the run function. Finally, deleting the address of the second node.

The round robin algorithm iterates through the whole list of tasks. This time, we compare the burst time with the Quantum, which is the fixed amount of time in the cpu which is set to 10. If the current burst time is larger than what the time quantum can handle, the quantum will decrement from the burst time. If it the burst time exceeds quantum, that task will be placed into a circular queue. Then, it will pass in the entire list and deletes previous task.

Results

```
herropaul@herropaul-VirtualBox:~/Desktop/Proj_4_sample_code-20201201T022734Z-001
/Proj_4_sample_code/CPU_scheduler$ ./fcfs schedule.txt
Running task = [T8] [10] [25] for 25 units.
Running task = [T7] [3] [30] for 30 units.
Running task = [T6] [1] [10] for 10 units.
Running task = [T5] [5] [20] for 20 units.
Running task = [T4] [5] [15] for 15 units.
Running task = [T3] [3] [25] for 25 units.
Running task = [T2] [3] [25] for 25 units.
Running task = [T1] [4] [20] for 20 units.
```

```
herropaul@herropaul-VirtualBox:~/Desktop/Proj_4_sample_code-20201201T022734Z-001
/Proj_4_sample_code/CPU_scheduler$ ./priority pri-schedule.txt
Running task = [T6] [1] [50] for 50 units.
Running task = [T5] [1] [50] for 50 units.
Running task = [T4] [1] [50] for 50 units.
Running task = [T3] [1] [50] for 50 units.
Running task = [T2] [1] [50] for 50 units.
Running task = [T1] [1] [50] for 50 units.
```

```
herropaul@herropaul-VirtualBox:~/Desktop/Proj_4_sample_code-20201201T022734Z-001
/Proj_4_sample_code/CPU_scheduler$ ./rr schedule.txt
Running task = [T8] [10] [25] for 10 units.
Running task = [T7] [3] [30] for 10 units.
Running task = [T6] [1] [10] for 10 units.
Task [T6] finished.
Running task = [T5] [5] [20] for 10 units.
Running task = [T4] [5] [15] for 10 units.
Running task = [T3] [3] [25] for 10 units.
Running task = [T2] [3] [25] for 10 units.
Running task = [T1] [4] [20] for 10 units.
Running task = [T8] [10] [15] for 10 units.
Running task = [T7] [3] [20] for 10 units.
Running task = [T5] [5] [10] for 10 units.
Task [T5] finished.
Running task = [T4] [5] [5] for 5 units.
Task [T4] finished.
Running task = [T3] [3] [15] for 10 units.
Running task = [T2] [3] [15] for 10 units.
Running task = [T1] [4] [10] for 10 units.
Task [T1] finished.
Running task = [T8] [10] [5] for 5 units.
Task [T8] finished.
Running task = [T7] [3] [10] for 10 units.
Task [T7] finished.
Running task = [T3] [3] [5] for 5 units.
Task [T3] finished.
Running task = [T2] [3] [5] for 5 units.
Task [T2] finished.
```

Contributions:

Paul Nguon - Helped wrote the code for FCFS, Priority, and RR files

- add() and schedule()
- ReadMe File.

Sam Chen - Helped wrote the code for FCFS, Priority, and RR files

- add() and schedule()
- Report

Sam Chen - 013502214

Paul Nguon - 015728505