

## Report for Scheduler Policies

The following policies have been implemented:

- 1) First Come First Serve (FCFS)
- 2) Multi Level Feedback Queue (MLFQ)

### 1) First Come First Serve (FCFS):

It's a scheduling algorithm used in computing to determine the order in which tasks or processes are executed by a computer's central processing unit (CPU).

In FCFS, the first task that arrives is the first one to be executed. Once a task starts running, it continues until it's finished or until it voluntarily relinquishes the CPU.

In this , we go through the list of all process list and select the process whose entry time is the smallest(first obtained process,which is present in myproc()->ctime) , and we execute it until it is in “RUNNABLE” state.

We observe that average runtime(rtime) and waittime(wtime) of this process for the given “schedulertest” are :

rtime : 14 ticks

wtime : 129 ticks

Also , we see that the order of process's getting completely executed is:

- 1) Process 5
- 2) Process 6
- 3) Process 7
- 4) Process 8
- 5) Process 9
- 6) Process 0
- 7) Process 1
- 8) Process 2
- 9) Process 3
- 10) Process 4

## 2) Multi Level Feedback Queue (MLFQ):

It's a CPU scheduling algorithm that dynamically adjusts the priority of processes based on their behavior. In this , we go through all process in the list and keep track of the process whose priority is highest or whose qnum is lowest , and execute it for 1 tick and increment its runtime . If at any point runtime of a process in that queue exceeds the given queue's timeslice , we decrease the priority or increase qnum.

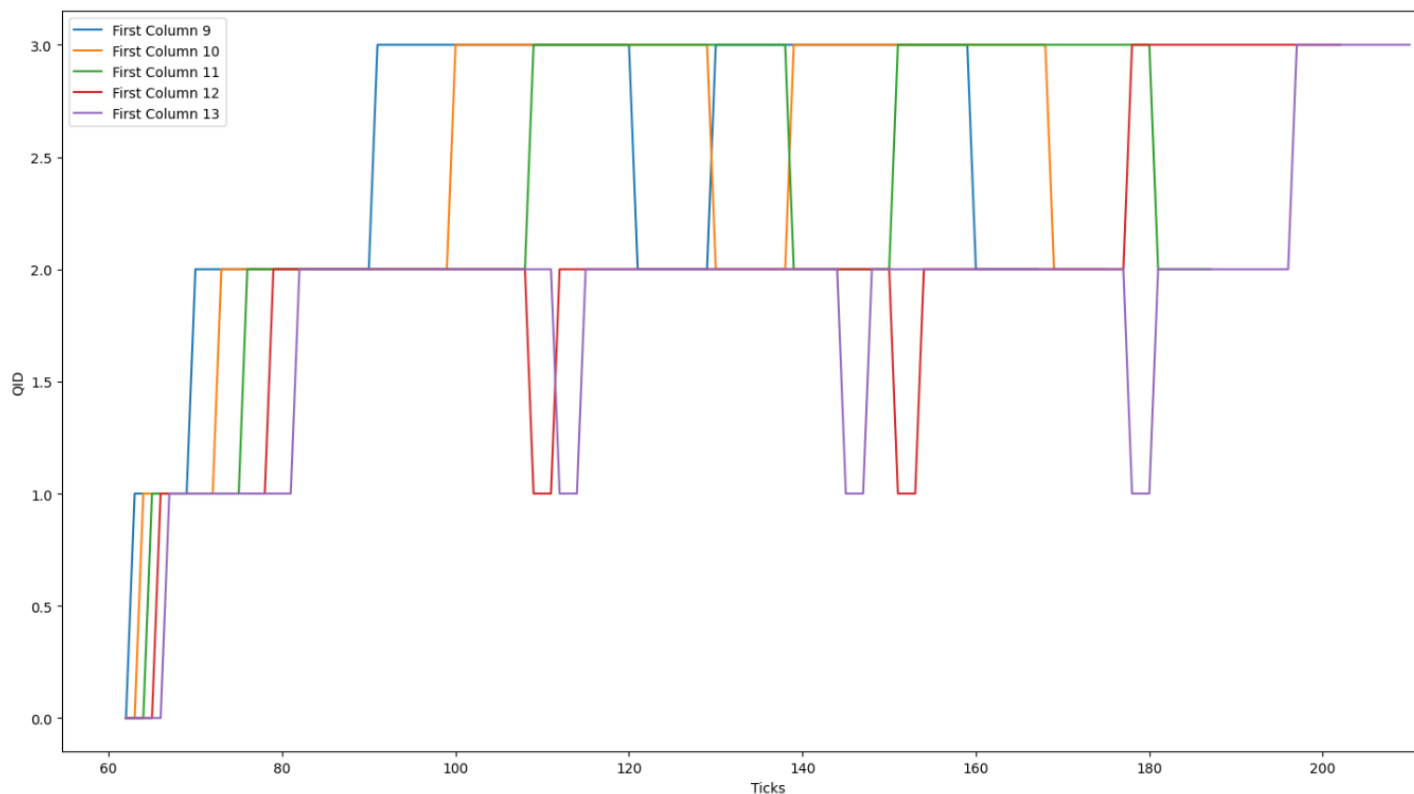
If a process's waittime exceeds aging time(30 ticks) , we increase the process' priority or decrease its qnum . Thus , we are able to implement MLFQ.

We see that average runtime and waittime for this is as follows:

Rtime: 14 ticks

Wtime: 149 ticks

Here's the graph for the above for aging time=30 ticks.



## Default Scheduler:

The default is Round Robin Scheduler. The Round Robin (RR) scheduling algorithm is a preemptive CPU scheduling method where each process is assigned a fixed time slice (which is 1 tick here). When a process's time slice expires, it is moved to the back of the ready queue, and the CPU scheduler picks the next process in line.

Here rtime: 14 ticks

Wtime: 156 ticks.