UNIVERSITY OF THESSALY

POLYTECHNICAL SCHOOL

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

«Analysis of Shortest Job First algorithm variant»

2nd Project

Operating Systems

Gianni Ioanna

Kapetanos Georgios

Zafeiri Stamatia Varvara

Supervisor: Christos Antonopoulos

2021-22

1. REQUIREMENTS

The goal of this project is to implement a variant of Shortest Job First scheduling algorithm where also the waiting time of a process in the ready queue can be taken into consideration besides only the expected burst.

As expected burst we describe the estimated time that the process will use the CPU on the next round.

On the following pages 3 scheduling types will be analyzed:

1. Round Robin variant that was already given
2. Shortest Job First using only the exp_burst
3. Shortest Job First using also the time each process spends in the ready queue

2. ALGORITHM DESCRIPTIONS

A. Round Robin : Based on this algorithm, all processes share the processor equally based on a specific time period

B. Shortest Job First: The SJF algorithm chooses the process to be executed that has the lowest execution time. This is achieved by monitoring the previous burst times of each process and the estimation of the future burst via:

$$Exp\_Burst_i = \frac{Burst_{i-1} + \alpha * Exp\_Burst_{i-1}}{1+\alpha}$$

Based on the exp_burst, the priority of each process is calculated via:

$$Goodness(k)_i = \frac{1 + Exp\_Burst(k)_i}{\min_{m=0}^{N}(1 + Exp\_Burst(m)_i)}$$

C. Shortest Job First + waiting time in queue : Here despite the above, also the waiting time in the ready queue is taken into consideration so the priority of each process is calculated via:

$$Goodness(k)_i = \frac{1 + Exp\_Burst(k)_i}{\min_{m=0}^{N}(1 + Exp\_Burst(m)_i)} * \frac{\max_{m=0}^{N}(1 + WaitingInRQ(m)_i)}{1 + WaitingInRQ(k)_i}$$

In cases B & C lower goodness = higher priority

3. EXPERIMENTAL ANALYSIS

For the experimental analysis 3 types of profiles where used:

- Only programs that use calculations (no or limited usage of I/O)– non-interactive
- Only programs using I/O – interactive
- Both

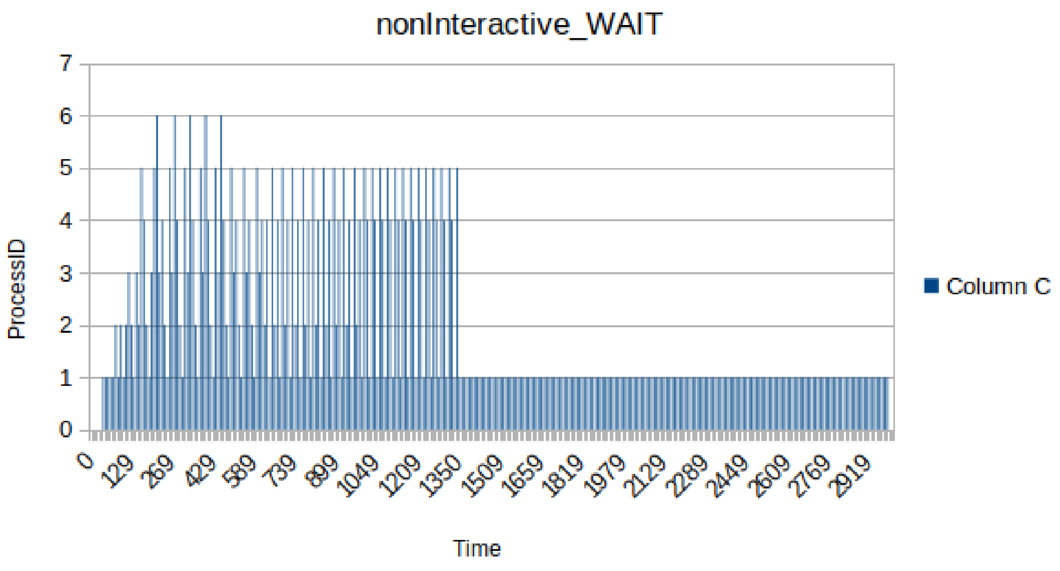
Particularly, the profiles created are:

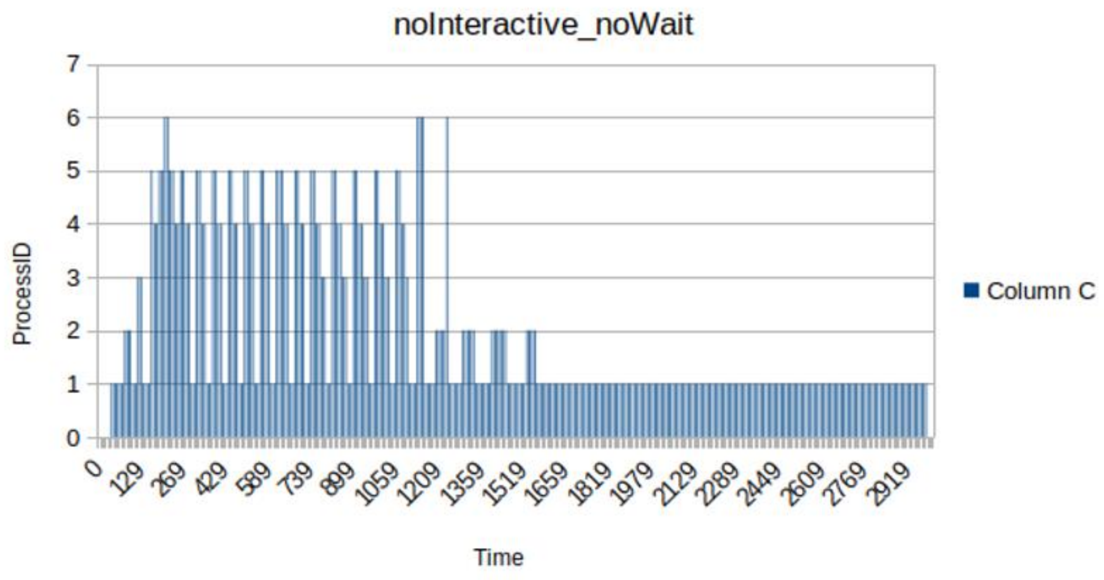
1. Non-interactive processes
   a. 1st profile includes 6 processes
   b. 2nd profile includes 4 processes
2. Interactive processes
   a. 6 only interactive processes
   b. 5 only interactive processes spawned simultaneously
3. 4 interactive and non-interactive processes

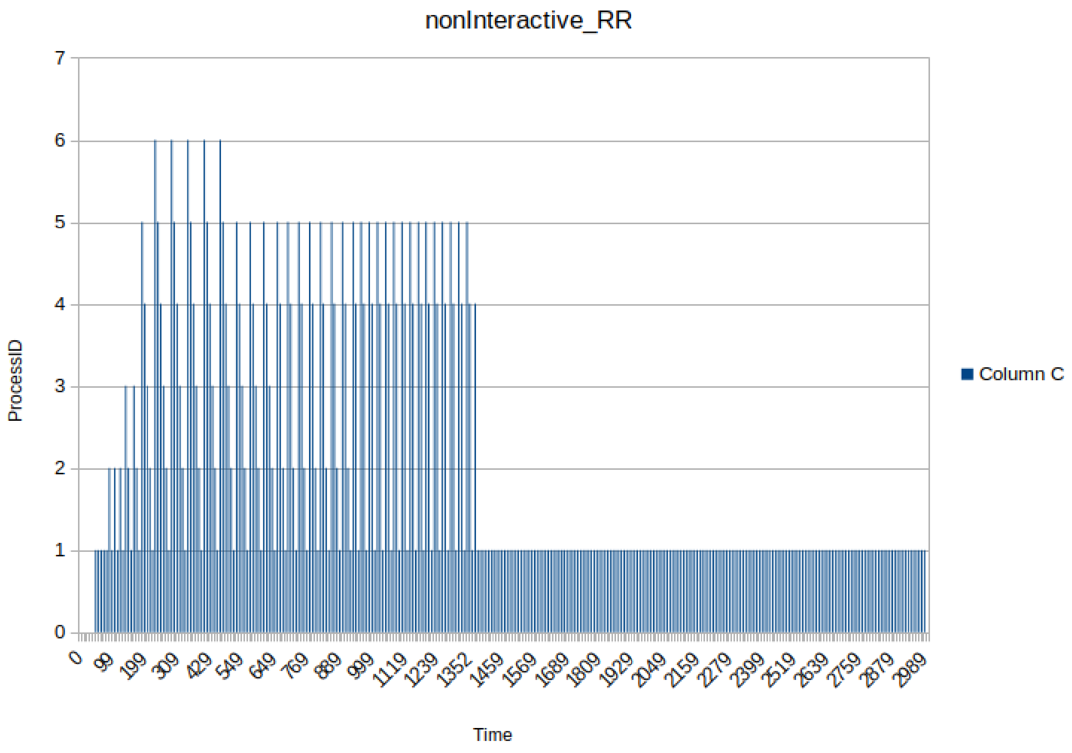
Based on the profiles above there will be a comparison between the 3 algorithms using graphical representation of the results.

NON-INTERACTIVE PROCESSES

On the first profile 6 interfaces were used that start one after another with little spawn time difference and the first interface has higher work duration than the others.







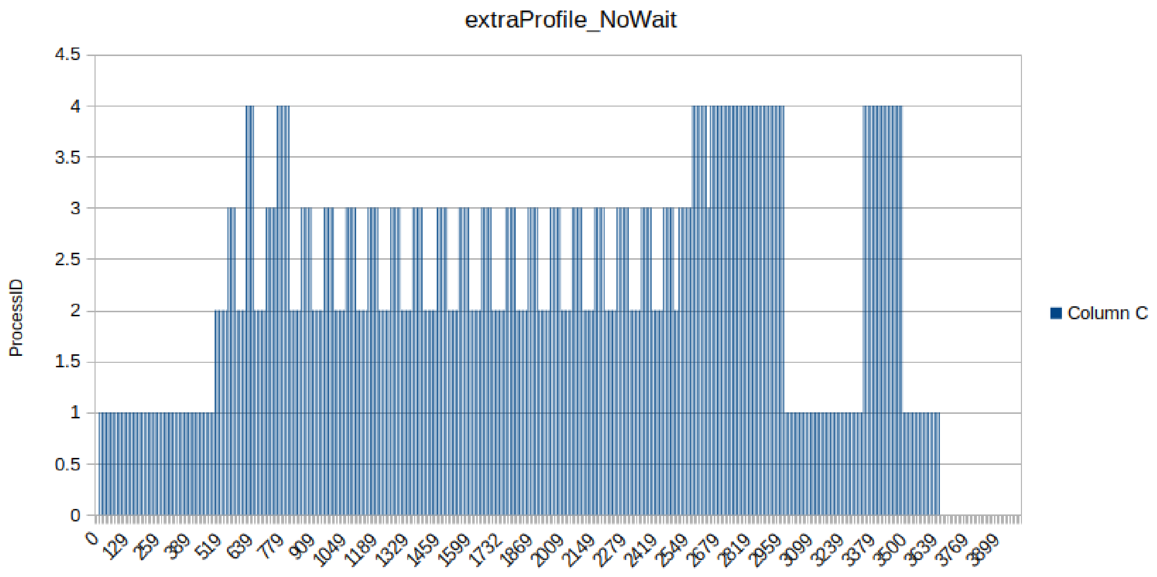
From the graphs above we can see that there is an issue as far as it concerns fairness between the interfaces when only exp_burst is used since an interface can hold the processor for longer time.

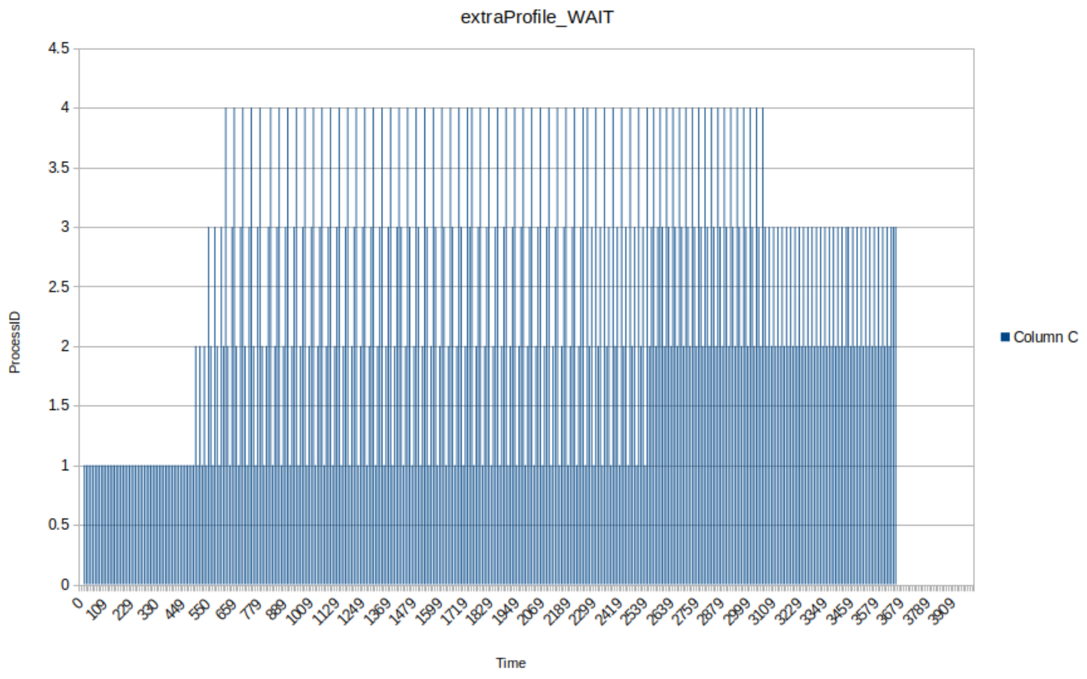
This issue does not occur when the waiting parameter is added and fairness is established between the interfaces.

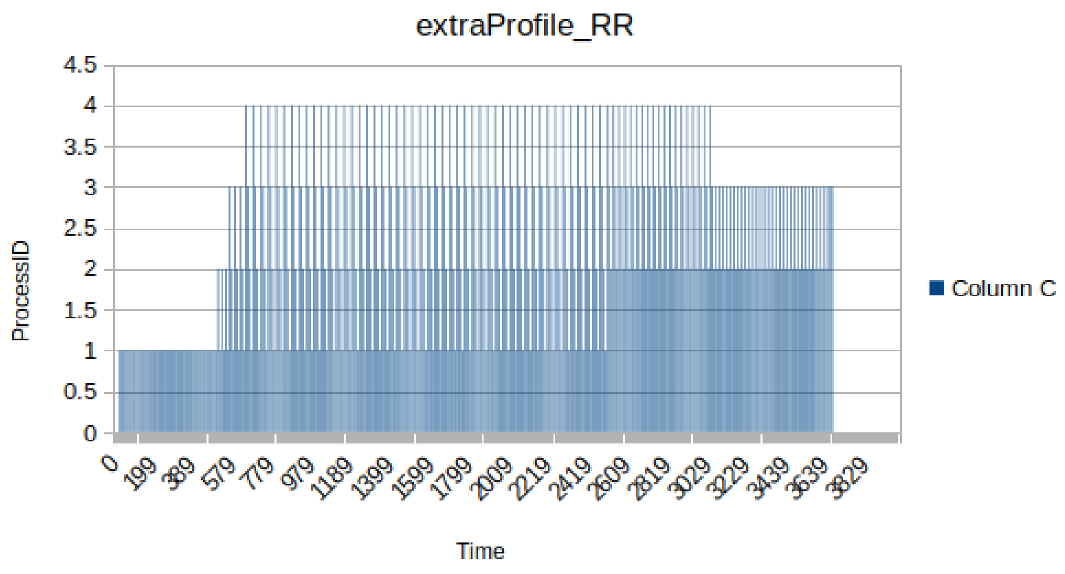
As it is expected, the “bigger” interface finishes last here.

In addition, another observation is the fact that Round Robin and SJF with wait give more or less the same results. This is due to the fat that each time an interface leaves the ready queue the waiting time is reset while the exp_burst will grow. As a result, the probability of the same process to retake the processor right after is little. Fairness exists though since after the process gets into the queue the waiting time will start to grow indicating that after a while the interface will use the Processor again. On the other side, RR keeps the fairness between the interfaces by swapping at fixed times.

On the 2nd **Profile**, there are 4 interfaces. The difference here is that the first interface begins way earlier than the others and all interfaces have more or less similar work durations.





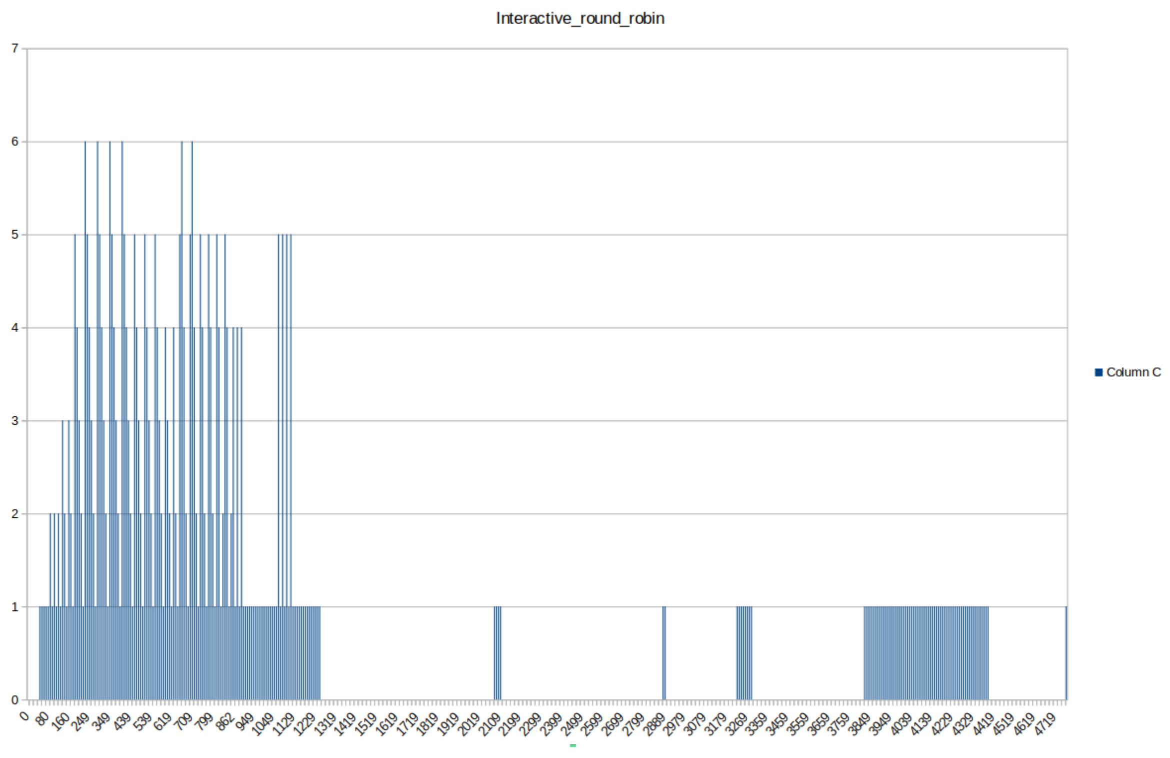


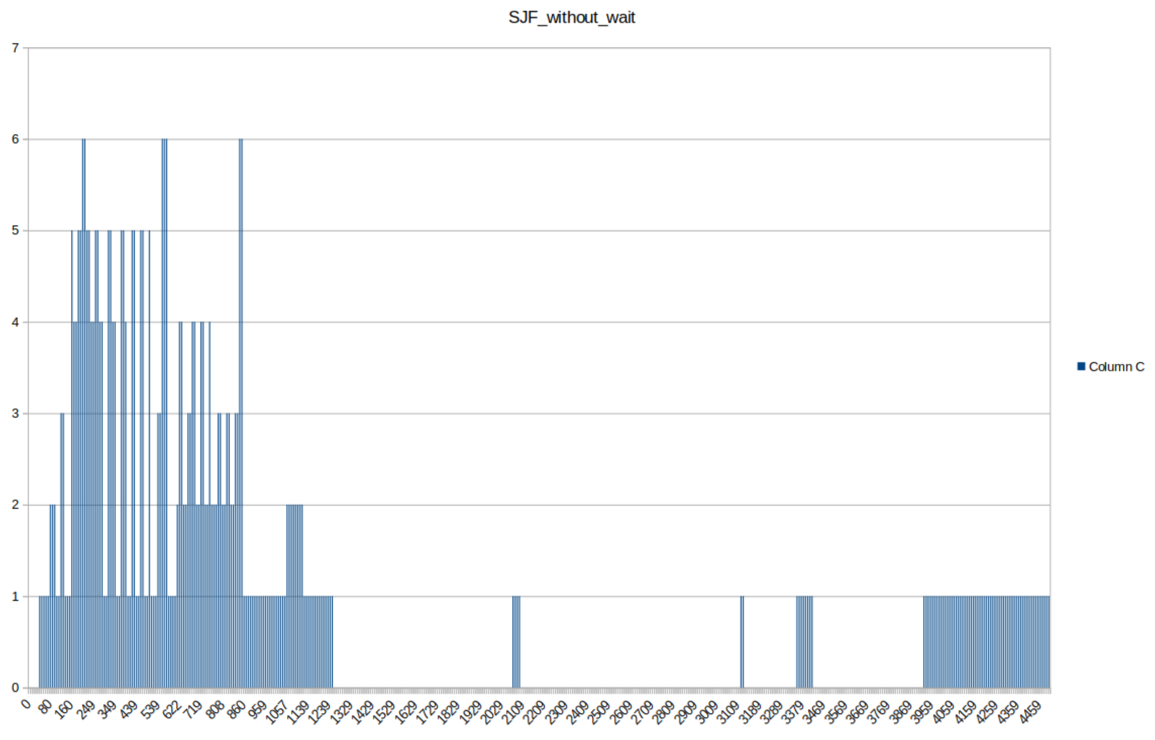
Based on the above, as far as it concerns the goodness computation, since the first interface is working alone and gets the CPU all the time, the exp_burst rises. As a result, when it comes to choosing which interface will run, when the new interfaces come, with initial exp_burst 0, they'll get priority and the first interface will wait.

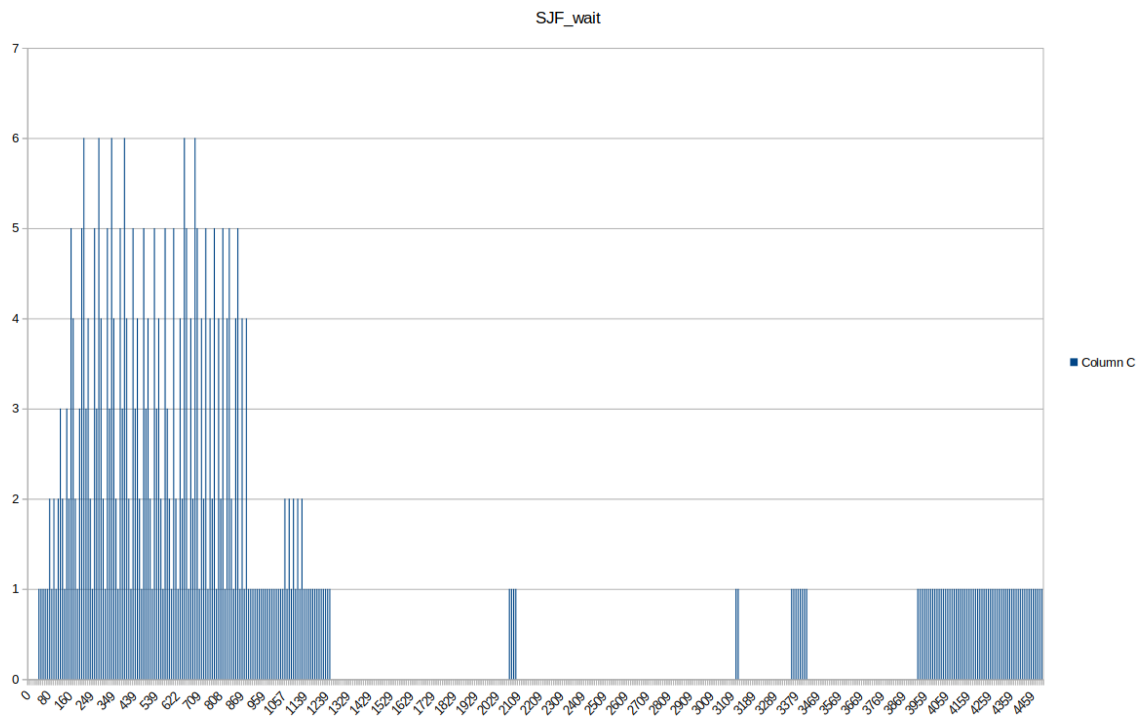
This is solved with using also the wait parameter.

Also, it is again visible the similarity of the RR and SJF_wait.

- **INTERACTIVE PROCESSES**

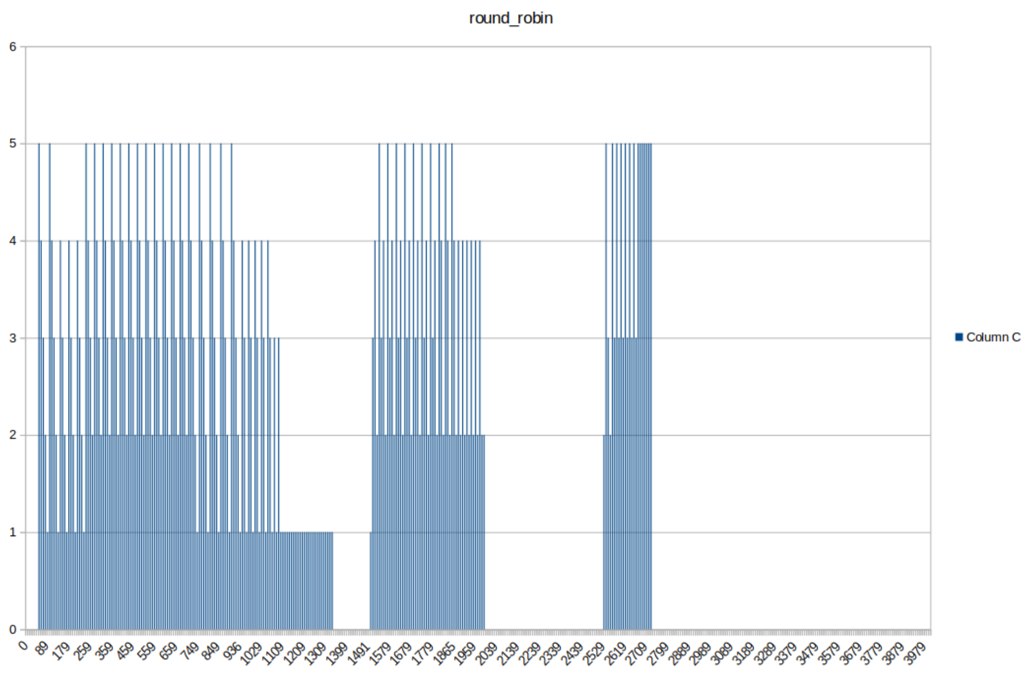

Interactive_round_robin
0
1
2
3
4
5
6
7
Column C

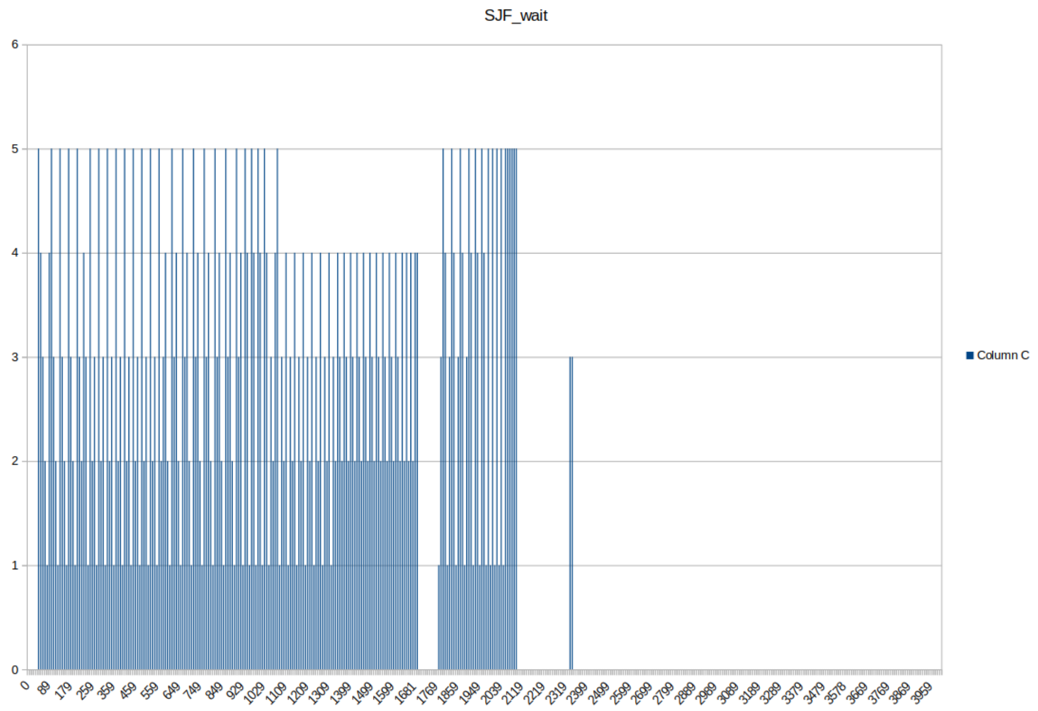

SJF_without_wait
0
1
2
3
4
5
6
7
Column C



As previously noted in the non-interactive profiles, the results from Round Robin are similar to SJF with wait. Unlike those two algorithms, SJF without waiting time in queue has longer burst duration on processes because of a high probability of a process retaking the execution on a subsequent CPU slice, causing higher latency and wait time on processes.

- ## INTERACTIVE PROCESSES Simultaneous spawning





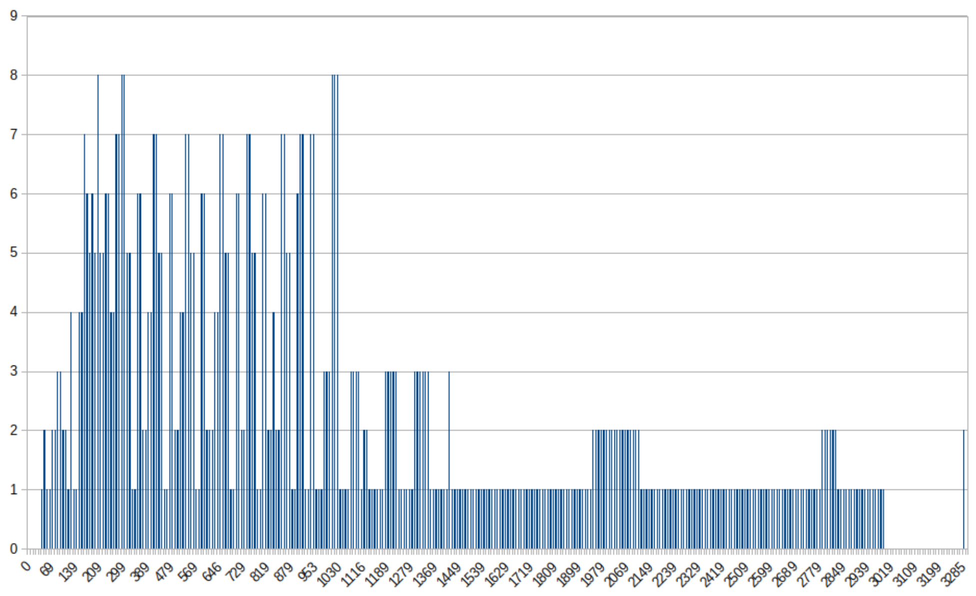
In this profile, five interactive process of work duration 400 ms are spawned simultaneously at 40 ms. While using round robin algorithm, each process is being executed in a repetitive manner based on the time it was added on the run queue. However, the SJF with waiting time in queue, takes into account the time each process was woken up from sleep and assigns them a lower priority.

○ INTERACTIVE AND NON-INTERACTIVE PROCESSES

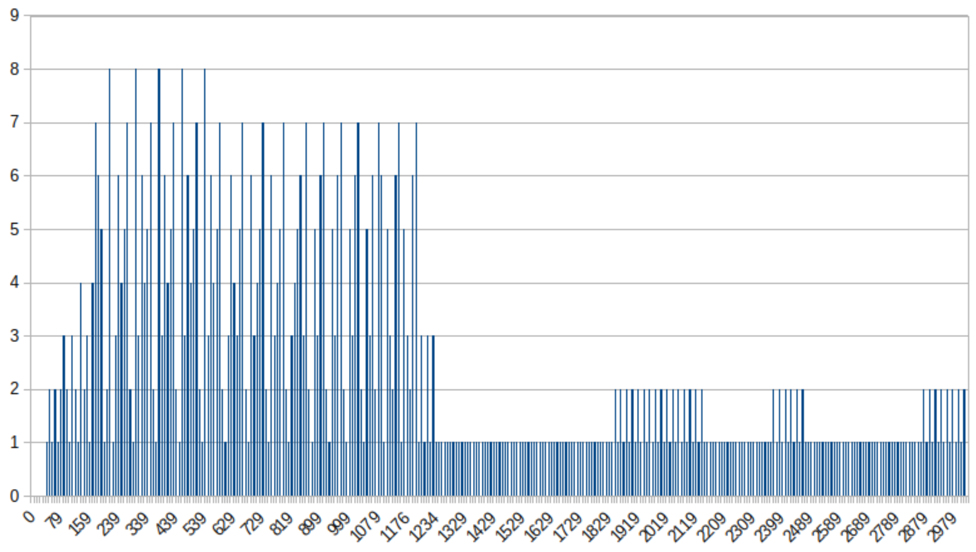
On this profile there are 4 interactive and 4 non-interactive processes. Processes 1, 3, 5 and 7 are non-interactive and 2, 4, 6, 8 are interactive. 1 and 2 start one after another with little spawn time difference and they have higher work duration than the others. 5, 6, 8 have the same spawn time and duration.

SJF with no wait algorithm, frequently runs processes in successive CPU time slices, while SJF with wait and Round robin retain a fairer switch between each process.

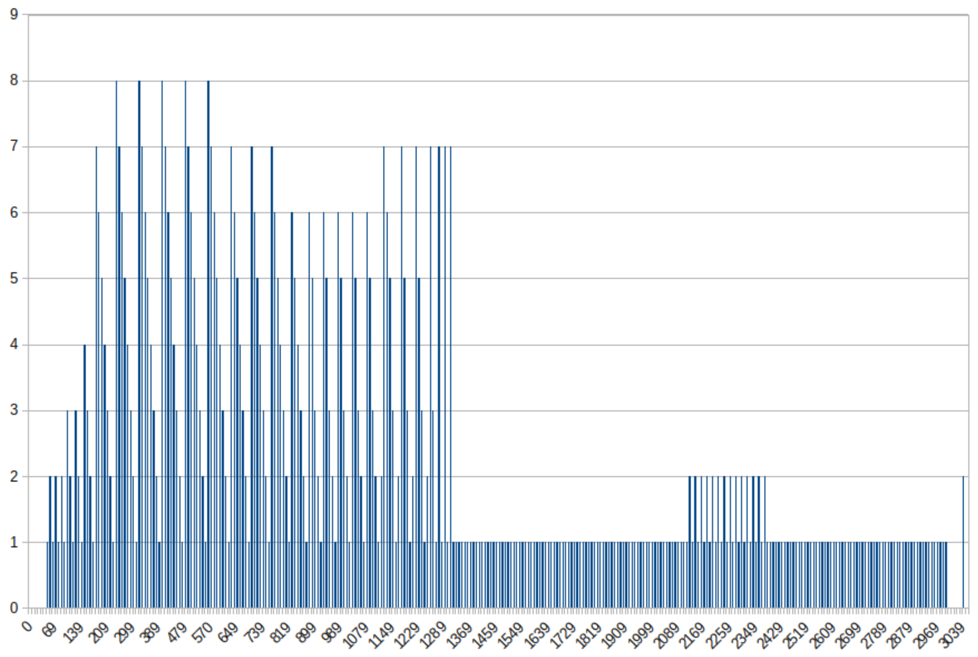
Mixed no wait



Mixed with wait



Mixed round robin



9
8
7
6
5
4
3
2
1
0
0
69
139
209
299
389
479
570
649
739
819
899
989
1079
1149
1229
1289
1369
1459
1549
1639
1729
1819
1909
1999
2089
2169
2259
2349
2429
2519
2609
2699
2789
2879
2969
3039