**SJF and Round Robin on Raspberry Pi**

**Project Description:**

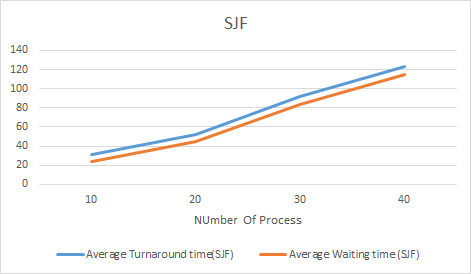
We implemented Shortest Job First and Round-Robin scheduling algorithm using C Language. We took different number of processes and calculated their average waiting and turnaround time.

**Implementation:**

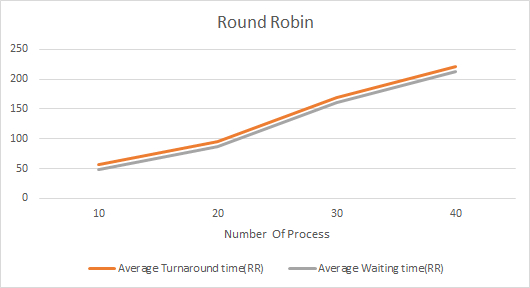
10, 20, 30, and 40 processes were taken respectively for both the algorithms. The input to code is given through the filing. The code is implemented in C language. Raspberry Pi 3 was the platform.   
We assume that the arrival time of all the process in zero and time quantum of Round-Robin is 8 units.

**Project Result:**

In SJF according to the definition, short processes are executed first and then followed by longer processes. The throughput is increased because more processes can be executed in less amount of time.   
The major drawback is that the longer processes will have more waiting time, eventually they'll suffer starvation.



In Round-Robin each process is served by the CPU for a fixed time quantum, so all processes are given the same priority. Starvation doesn't occur because for each round robin cycle, every process is given a fixed time to execute. No process is left behind.  
The throughput in Round-Robin largely depends on the choice of the length of the time quantum. If time quantum is longer than needed, it tends to exhibit the same behavior as FCFS. If time quantum is shorter than needed, the number of times that CPU switches from one process to another process, increases. This leads to decrease in CPU efficiency.



In both the graph as the number of processes increases the average waiting and turnaround time increases.

Conclusion:

As compared to Round-Robin, SJF is more efficient irrespective of the time quantum. But if the time quantum is selected maximum then Round-Robin is more efficient than the SJF. Efficiency of the Round-Robin depends on the time quantum.

Project Link: <https://github.com/HassaanSaleem/Round-Robin-vs-SRTF>

