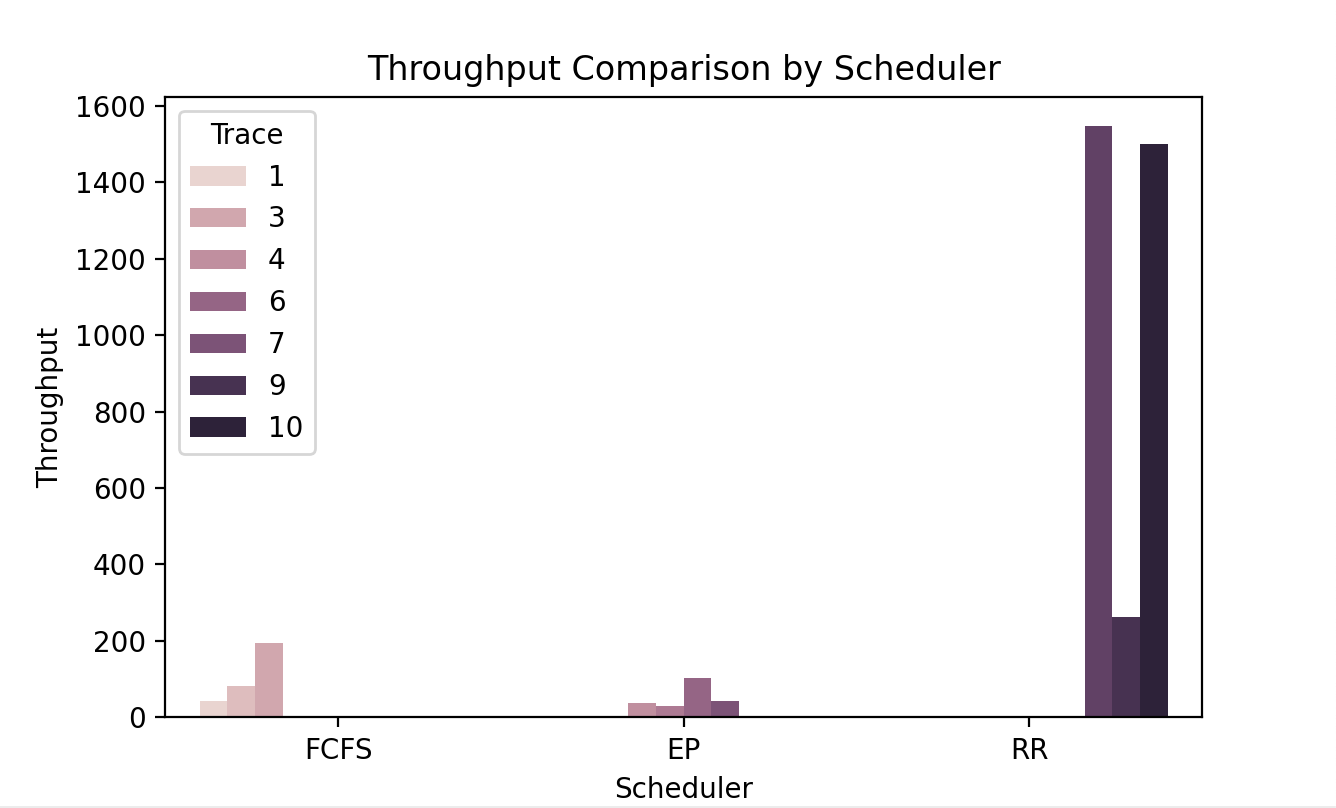
**Part 1**

In Part 1 of the assignment, we simulated three different scheduling algorithms; First Come First Serve (FCFS), External Priority (EP), and Round Robin (RR). Three different approaches that all come with their pros and cons. Choosing the right algorithm is rhetorical and depends on many details. To further explain, the following report will dive into the differences between all three algorithms alongside some simulation analysis to see how they perform against one another by comparing their Throughput, Average Turnaround Time (TAT), and Average Wait Time spent in the ready queue.

Throughput

Throughput is the amount of jobs that can be completed in a given time. Before running any simulation we try to predict who will have the best throughput. Since RR slices CPU time fairly between all jobs, it may lead to the highest. The quantum allows progress through each process leading to a higher number of jobs finishing within a given time. Conversely, EP and FCFS have a high risk of low throughput. The former is at risk of starvation (i.e. lower priority jobs never running) and the latter is a risk of the convoy effect (i.e. shorter jobs take a long time to complete because the first process was long). These deductions were justified by the figure below:



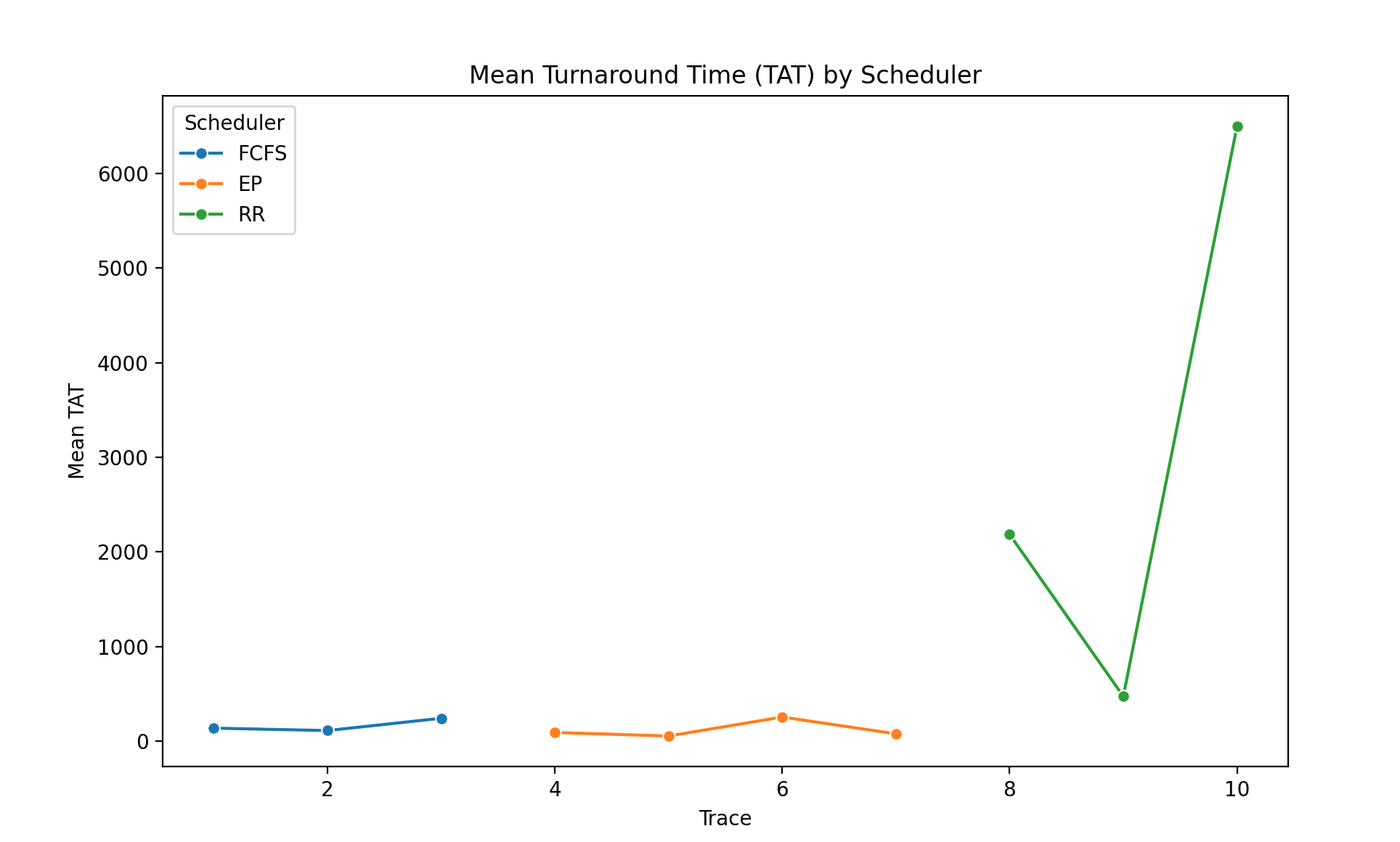
**Figure 1:** Graph Displaying Throughput Versus Scheduler Type

From Figure 1, we see that our predictions were affirmed given that RR yielded the highest throughput followed by FCFS and EP. Please note that the units for the Y-axis aren’t constant because we tried to diversify our test cases to make sure our scheduler works in all conditions.

Average Turnaround Time (TAT)

Based on our knowledge of how these algorithms we can make the following deductions:

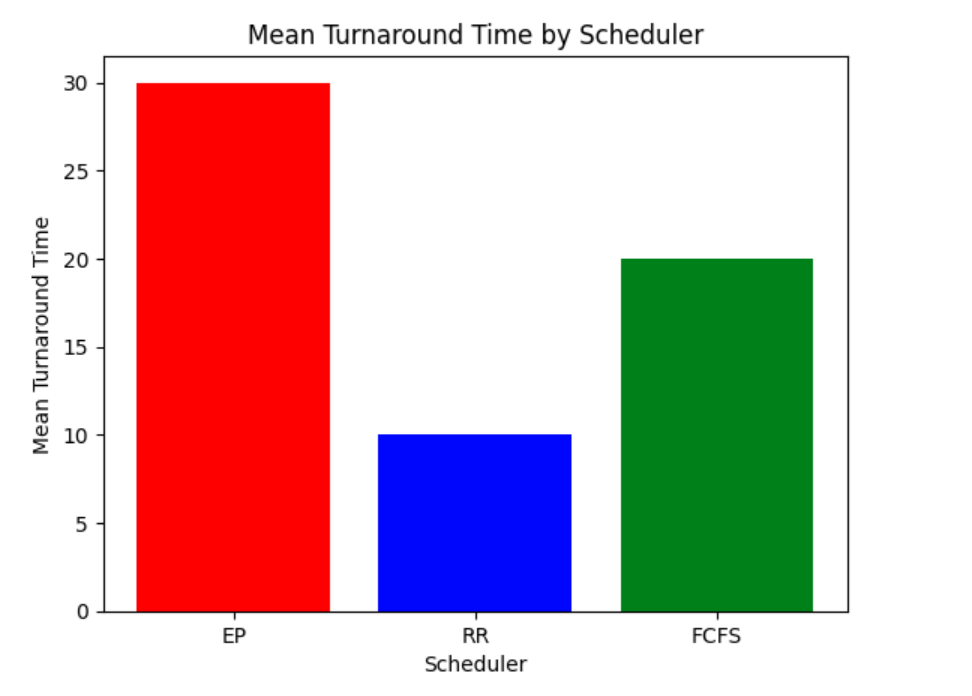
* FCFS will have the worst TAT given that it does not prioritize scheduling based on burst times. This can result in long waiting times - again, due to the convoy effect.
* RR can have moderate TAT if processes are similar in length because it doesn't allow processes to monopolize the CPU. However, due to the constant context switching, it can lead to some kind of TAT.
* EP TAT is dependent on how the priority is set. To further explain, if priority is given based on burst time, TAT can drop significantly. However, if the priority is given based on PID - like in our case - it can lead to a higher average TAT. Figure 2 shows the simulation results:



**Figure 2:**  Graph Displaying Mean TAT Versus Scheduler Type

Average Wait Time in Ready Queue

Based on our understanding of all algorithms, we can deduce that the AWT will be the least in RR given the quantum. We can also - similar - to the previous the former two criteria - EP and FCFS can experience high AWT because of starvation and the convoy effect. Justification can be seen below:



**FIgure 3** Average Wait Time

As expected, RR had the lowest AWT due to its CPU efficiency followed by FCFS and EP.