For seed 1797410758:

For this seed, the FCFS is the slowest in terms of waiting times. This is because there are processes with long service times in the middle of the process list. This results in many processes after the long process waiting for the long process to finish. This can be seen when process #5’s long service time prevents the faster process #8 from being serviced quickly when it arrives. SJF is slightly faster as it picks the fastest process from those that have arrived however if it can only find slow processes that have arrived, it will run those and block faster future arrivals. RR has faster waiting times than SJF as it gradually works on all processes, such as longer to service processes, whilst also dealing with faster processes efficiently (especially if they are faster than the time quantum used in the RR algorithm. The STRF algorithm had the fastest waiting times by far due to the fast processes having next to no waiting time. This is due to the SRTF algorithm running processes faster than the current process as soon as they arrive whilst leaving the longer process to be ran later. This is very effective when there are a few long to service processes blocking a large amount of shorter processes from being dispatched. Adding a context switch time of 0.5 increased the turnaround times for each algorithm whilst having a particularly profound effect on the RR algorithm, making it the slowest of the four. This is due to the RR algorithm having to switch between running and suspended processes by far the most.

For seed 2688744162:

For this seed, the SRTF algorithm is the fastest, followed by the SJF algorithm and then the FCFS algorithm. This seed features a large amount of quick-to-service processes and these algorithms are able to handle them effectively as they to not add any unnecessary delays. On the other hand, the slowest algorithm of the RR algorithm. The RR algorithm had a quantum of 0.5 on by default, which was slightly higher than the servicing time of most of the processes. This resulted in relatively fast processes being suspended whilst being close to completion, adding delays to waiting time. Adding a context switch time once again strongly effected the RR algorithm, causing it to have a turnabout time almost 3 times longer than the SRTF algorithm.

For seed 3399474557:

For this seed SRTF was the fastest algorithm, followed by SJF, then RR, and lastly FCFS. The results are largely similar to the results for the seed 1797410758. The main difference is that the RR algorithm is now slower. This is due to the presence of faster algorithms that RR suspends frequently. Increasing the time quantum makes the RR algorithm less effected by context switch time but it also makes the algorithm more similar to FCFS.