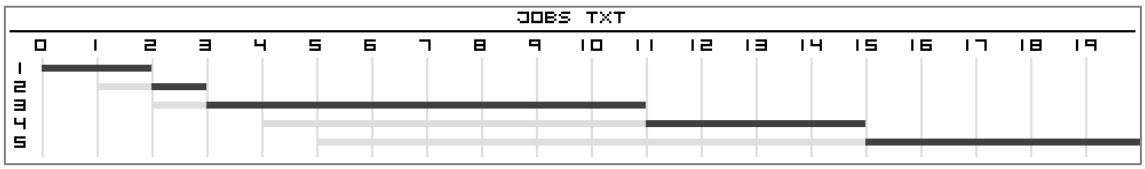
Part 1 – FCFS (First Come, First Serve)

Command: ./fcfs jobs.txt

```
Average Wait Time      : 3.8 s
Throughput (p/min)     : 15 p/min
Average Turnaround Time: 7.8 s
```

Visualisation (FCFS)

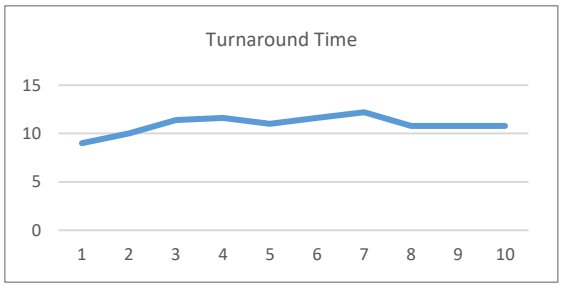


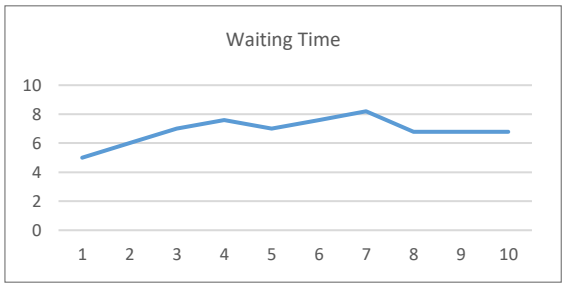
Part 2 – RR (Round Robin)

Command: ./rr jobs.txt

```
Average Wait Time      : 5 s
Throughput (p/min)     : 15 p/min
Average Turnaround Time: 9 s
```

Next, we will vary the quantum and display the changes in the average turnaround time and waiting time for processes in "jobs.txt". This change will be from 1 to 10.





Visualisation (RR)

Part 3 – SJF (Shortest Job First)

Non-preemptive

Command: ./sjf jobs.txt -x

```
Average Wait Time      : 3.8 s
Throughput (p/min)     : 15 p/min
Average Turnaround Time: 7.8 s
```

Preemptive

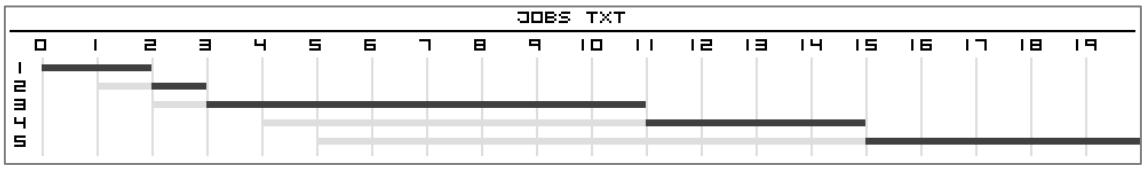
Command: ./sjf jobs.txt

```
Average Wait Time      : 2.8 s
Throughput (p/min)     : 15 p/min
Average Turnaround Time: 6.8 s
```

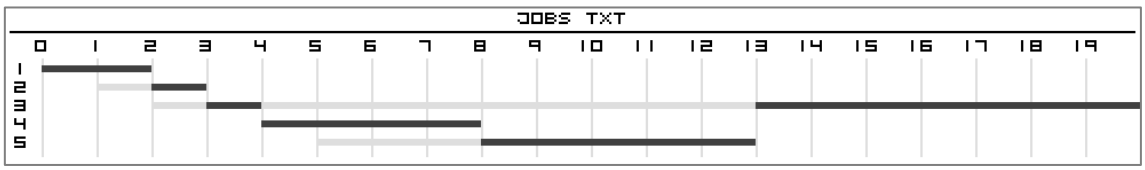
For the remainder of SJF, we will be utilising the ***non-preemptive*** *calculation.*

Comparing the waiting times of SJF to RR and FCFS's... SJF comes in at 3.8s, FCFS comes in at 3.8s, and RR comes in at 5s. Both SJF and FCFS give the shortest average wait time in this case because SJF simply does the same thing as FCFS with these jobs in this file, because processes come in and are executed in that order and must be completed before starting another. If we ran the **preemptive** SJF, it would come in at **2.8s**, being the fastest.

Visualisation (SJF Non-preemptive)



Visualisation (SJF Preemptive)



Part 4 – Priority Scheduling (Extra Credit)

Command: ./priority jobs.txt

```
Average Wait Time      : 2.8 s
Throughput (p/min)     : 15 p/min
Average Turnaround Time: 6.8 s
```

Finally, for extra credit, we will compare Priority scheduling to the other three. We will assume non-preemptive SJF for this comparison, along with preemptive Priority scheduling:

FCFS	3.8s
RR	5s
SJF (non-preemptive)	3.8s
Priority	2.8s

Priority based scheduling utilises the priority of each job, where the lower the number means the higher the priority is to the CPU. In addition, it is preemptive. Therefore, if a job with a higher priority appears, the CPU will switch control over to that job instead.

Visualisation (Priority)