Hassan Ardeshir 610398202

CPU Scheduling

CPU scheduling is the process of allocating CPU time to various processes running in a computer system. The goal of CPU scheduling is to make the system efficient, fast, and fair to all processes. There are various algorithms used for CPU scheduling and each algorithm has its own advantages and disadvantages. In this report, we will discuss four popular CPU scheduling algorithms: Round Robin (RR), First Come First Serve (FCFS), Priority Scheduling, and Shortest Job First (SJF) And then for each of them, we have embedded a simulation in cpp language for testing.

First Come First Serve (FCFS): FCFS is the simplest scheduling algorithm. In this algorithm, the process that arrives first is executed first. The CPU is assigned to each process in the order in which they arrive in the ready queue. This algorithm is non-preemptive, meaning that once a process starts executing, it runs until completion.

Shortest Job First (SJF): In SJF scheduling, the process with the shortest estimated execution time is executed first. The CPU is assigned to the process with the shortest estimated time in the ready queue. If two processes have the same estimated execution time, then the one that arrived first is executed first. This algorithm is non-preemptive, meaning that once a process starts executing, it runs until completion.

Round Robin (RR): In RR scheduling, each process is assigned a small time slice called a quantum. The CPU is repeatedly assigned to each process in the ready queue in a round-robin fashion. If a process does not complete its execution within the quantum, the CPU is assigned to the next process in the ready queue. This ensures that each process gets a fair share of the CPU.

Priority Scheduling: In this algorithm, each process is assigned a priority. The process with the highest priority is executed first. If two processes have the same priority, then the one that arrived first is executed first. This algorithm is preemptive, meaning that if a process with a higher priority arrives while a lower priority process is executing, the lower priority process is preempted, and the higher priority process is executed.

As we know, the following data are important for scheduling in processes:

Waiting time: It is the amount of time a process spends waiting in the ready queue for the CPU to be allocated to it.

Turnaround time: It is the time from the arrival of a process to its completion. It is the sum of the waiting time and the execution time for a process.

Completion time: It is the time at which a process completes its execution.

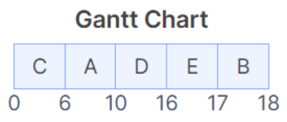
All three of these metrics are used to evaluate the performance of a CPU scheduling algorithm. They help determine the efficiency and fairness of the algorithm in allocating resources to processes. These metrics can be used to optimize the scheduling algorithm and improve the overall performance of the system.

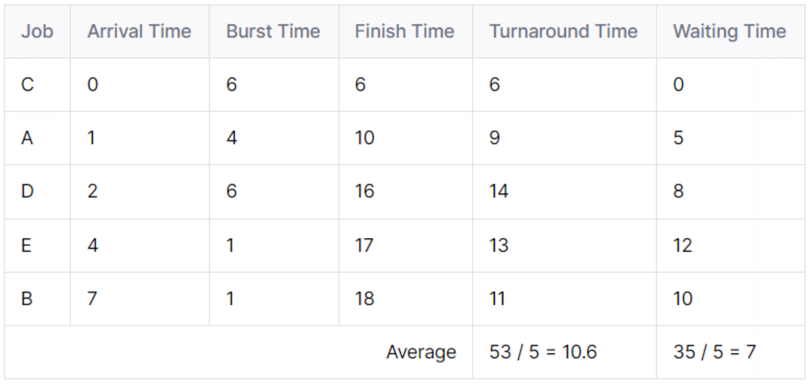
For example, we consider the following processes:

	Arrival Time	Burst Time	Priority
Process 1 (A)	1	4	3
Process 2 (B)	7	1	1
Process 3 (C)	0	6	1
Process 4 (D)	2	6	7
Process 5 (E)	4	1	4

And the simulator gives us the following output:

First Come First Serve (FCFS):





Job	Arrival Time	Burst Time	Finish Time	Turnaround Time	Waiting Time
C	0	6	6	6	0
A	1	4	10	9	5
D	2	6	16	14	8
E	4	1	17	13	12
B	7	1	18	11	10
			Average	53 / 5 = 10.6	35 / 5 = 7

FCFS Scheduling Results:

Process 1| Waiting Time: 5| Turnaround Time: 9| Completion Time: 10

Process 2| Waiting Time: 10| Turnaround Time: 11| Completion Time: 18

Process 3| Waiting Time: 0| Turnaround Time: 6| Completion Time: 6

Process 4| Waiting Time: 8| Turnaround Time: 14| Completion Time: 16

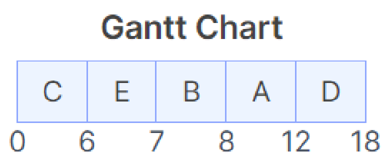
Process 5| Waiting Time: 12| Turnaround Time: 13| Completion Time: 17

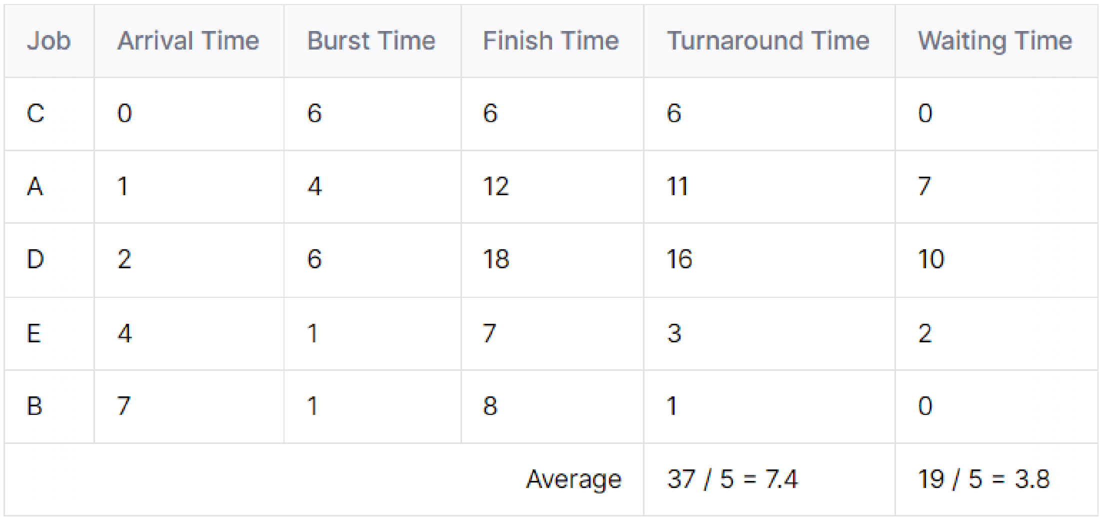
Average waiting time: 7

Average turnaround time: 10.6

Order of processes scheduled: [Process 3 for 6ms, Process 1 for 4ms, Process 4 for 6ms, Process 5 for 1ms, Process 2 for 1ms]

Shortest Job First (SJF):





Job	Arrival Time	Burst Time	Finish Time	Turnaround Time	Waiting Time
C	0	6	6	6	0
A	1	4	12	11	7
D	2	6	18	16	10
E	4	1	7	3	2
B	7	1	8	1	0
			Average	37 / 5 = 7.4	19 / 5 = 3.8

SJF Scheduling Results:

Process 1| Waiting Time: 7| Turnaround Time: 11| Completion Time: 12

Process 2| Waiting Time: 0| Turnaround Time: 1| Completion Time: 8

Process 3| Waiting Time: 0| Turnaround Time: 6| Completion Time: 6

Process 4| Waiting Time: 10| Turnaround Time: 16| Completion Time: 18

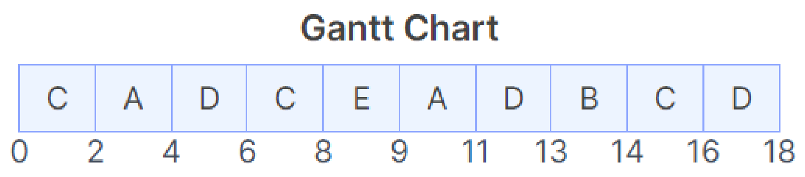
Process 5| Waiting Time: 2| Turnaround Time: 3| Completion Time: 7

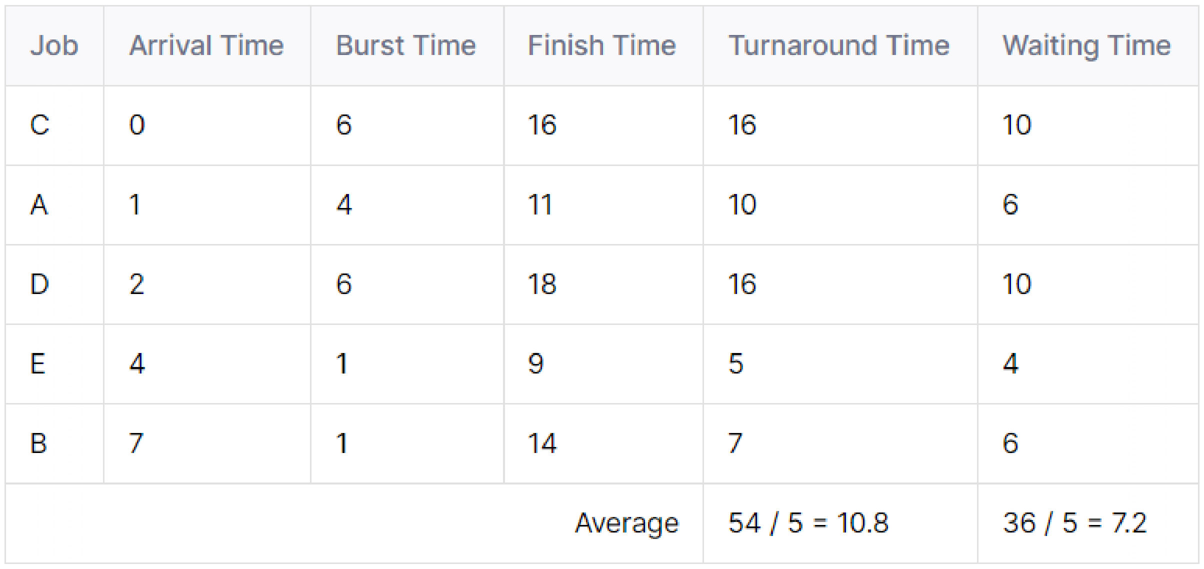
Average waiting time: 3.8

Average turnaround time: 7.4

Order of processes scheduled: [Process 3 for 6ms, Process 5 for 1ms, Process 2 for 1ms, Process 1 for 4ms, Process 4 for 6ms]

Round Robin (RR): (Quantum Time : 2)





Job	Arrival Time	Burst Time	Finish Time	Turnaround Time	Waiting Time
C	0	6	16	16	10
A	1	4	11	10	6
D	2	6	18	16	10
E	4	1	9	5	4
B	7	1	14	7	6
			Average	54 / 5 = 10.8	36 / 5 = 7.2

RR Scheduling Results:

Process 1| Waiting Time: 6| Turnaround Time: 10| Completion Time: 11

Process 2| Waiting Time: 6| Turnaround Time: 7| Completion Time: 14

Process 3| Waiting Time: 10| Turnaround Time: 16| Completion Time: 16

Process 4| Waiting Time: 10| Turnaround Time: 16| Completion Time: 18

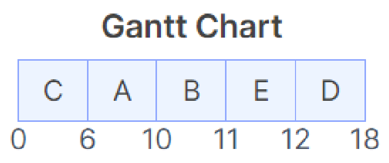
Process 5| Waiting Time: 4| Turnaround Time: 5| Completion Time: 9

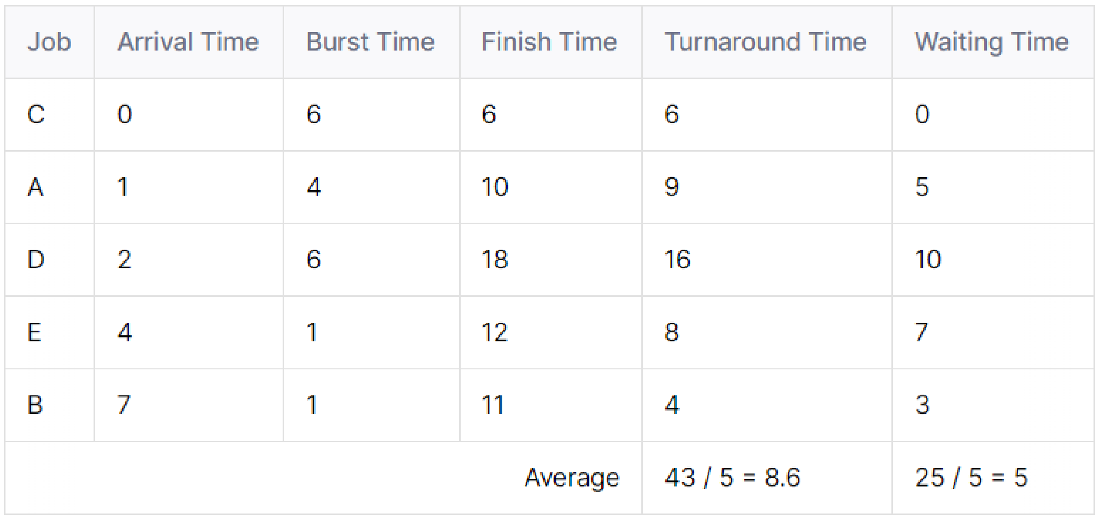
Average waiting time: 7.2

Average turnaround time: 10.8

Order of processes scheduled: [Process 3 for 2ms, Process 1 for 2ms, Process 4 for 2ms, Process 3 for 2ms, Process 5 for 1ms, Process 1 for 2ms, Process 4 for 2ms, Process 2 for 1ms, Process 3 for 2ms, Process 4 for 2ms]

Priority Scheduling:





Job	Arrival Time	Burst Time	Finish Time	Turnaround Time	Waiting Time
C	0	6	6	6	0
A	1	4	10	9	5
D	2	6	18	16	10
E	4	1	12	8	7
B	7	1	11	4	3
			Average	43 / 5 = 8.6	25 / 5 = 5

Priority (non-preemptive) Scheduling Results:

Process 1| Waiting Time: 5| Turnaround Time: 9| Completion Time: 10

Process 2| Waiting Time: 3| Turnaround Time: 4| Completion Time: 11

Process 3| Waiting Time: 0| Turnaround Time: 6| Completion Time: 6

Process 4| Waiting Time: 10| Turnaround Time: 16| Completion Time: 18

Process 5| Waiting Time: 7| Turnaround Time: 8| Completion Time: 12

Average waiting time: 5

Average turnaround time: 8.6

Order of processes scheduled: [Process 3 for 6ms, Process 1 for 4ms, Process 2 for 1ms, Process 5 for 1ms, Process 4 for 6ms]

The simulation that was proposed has the ability to show the correct results in the above algorithms and actually show what the output will be in the proposed algorithms and in what order the processes should be executed and what cpu time they will occupy.

It also has the ability to be implemented for all algorithms and the operating system can choose the best algorithm among the outputs based on its needs.

The schedulers mentioned are similar in many functions. All of them need a queue for ready processes, in which processes that have not yet been fully executed are placed in a certain order. For example, in the FCFS algorithm, this queue is in the order of arrival time. So, the main difference between these algorithms is only in the way of dealing with this queue, in what order to arrange it and in what order to execute it. For example, in RR, you specify a quantum time, and this queue is sorted by arrival time and executes each process according to the quantum time, and discusses the continuation of that process at the end of the queue. In this case, each process gets a balanced amount of CPU time.

cmake must be used to build the simulator, and the output program is located in build.