**CONTEXT**

1. INTRODUCTION
2. DESCRIPTION
3. ADVANTAGES AND DISADVANTAGES
4. CODE SNIPPET
5. TEST CASES
6. GITHUB REPOSITORY
7. CONCLUSION

**INTRODUCTION**

### *The priority scheduling algorithm is one of the most common algorithms for scheduling jobs in batch systems.*

### Types of Priority Scheduling Algorithm:

Priority scheduling can be of two types:

1. **Preemptive Priority Scheduling**: If the new process arrived at the ready queue has a higher priority than the currently running process, the CPU is preempted, which means the processing of the current process is stoped and the incoming new process with higher priority gets the CPU for its execution.
2. **Non-Preemptive Priority Scheduling**: In case of non-preemptive priority scheduling algorithm if a new process arrives with a higher priority than the current running process, the incoming process is put at the head of the ready queue, which means after the execution of the current process it will be processed.

**DESCRIPTION**

Every process is assigned a number which denotes the **priority**, and based on this priority the processes are executed. Therefore, the process having the highest priority (1) is executed first and then the priority 2, 3 and so on.

There can be some scenarios where more two or more processes may have the same priority. In this case, the processes are executed based on **First In First Out** order or in other words, **First Come First Serve**.

We do not consider the **arrival time** of the jobs in this non-preemptive priority scheduling algorithm. This means that unless a job gets completely executed, the CPU won’t leave the current job before it completes its execution.

Only once the process gets out of the job queue after successful execution, the CPU is allowed to process another job from the queue.

**DEFINTITION OF THE VALUES**

* Completion Time: Time at which process completes its execution.
* Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time
* Waiting Time(W.T): Time Difference between turn around time and burst time.  
  Waiting Time = Turn Around Time – Burst Time
* Burst time: The time required for the execution of process.

**ADVANTAGES**

* This algorithm is very simple to implement.
* The aging technique is implemented to reduce the starvation of lower priority processes.

**DISADVANTAGES**

* Starvation or indefinite blockage of the lower priority processes.
* Since this is a non-preemptive implementation, the waiting time is comparatively higher.
* The average turnaround time is higher as compared to the preemptive priority scheduling algorithm.
* If a system failure occurs, all the unfinished lower priority jobs get

vanished from the system.

**ALGORITHM**

**STEP1:-** Enter the no of processes.

**STEP2:-**  Enter the values for requested arrival time,burst time, and priority of the processes

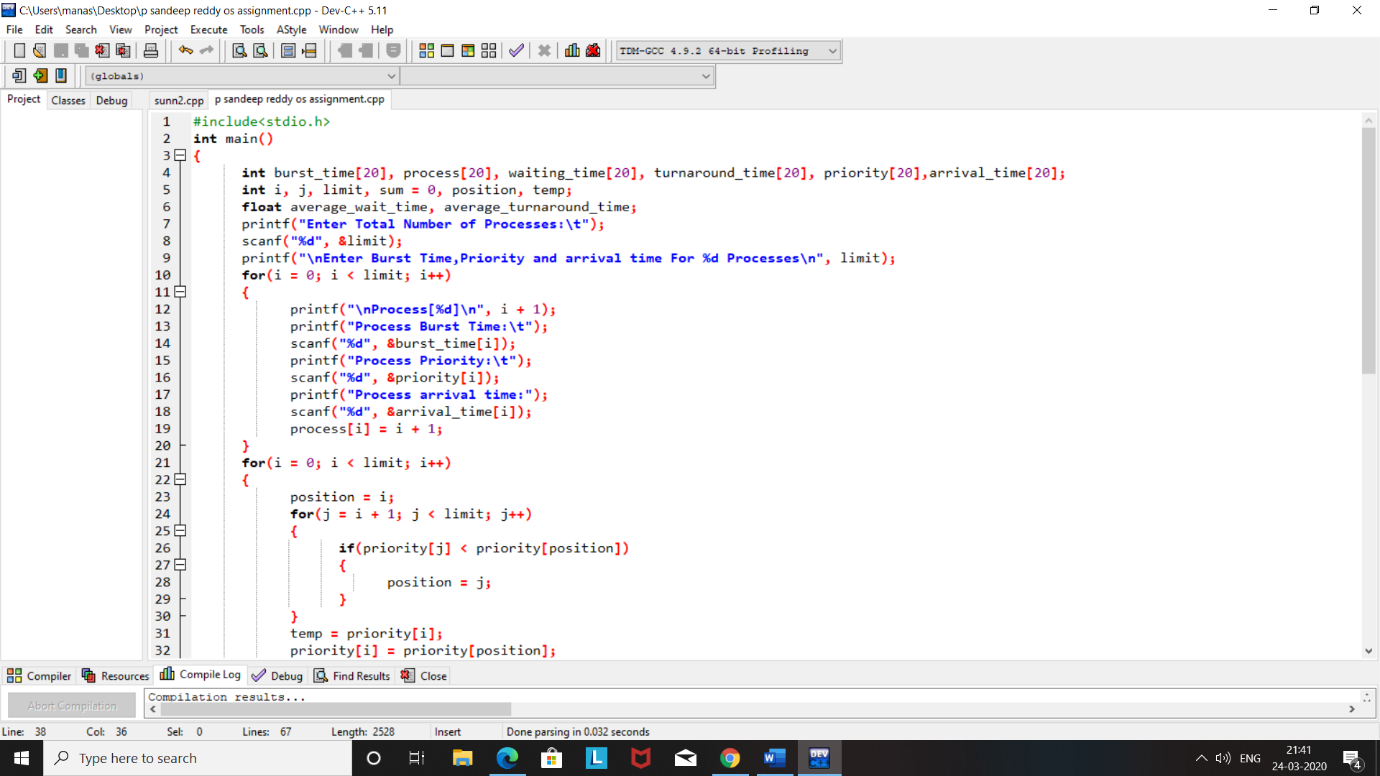
**STEP3:-** CPU executes the process with the highest priority that is the lowest integer value given for that process.

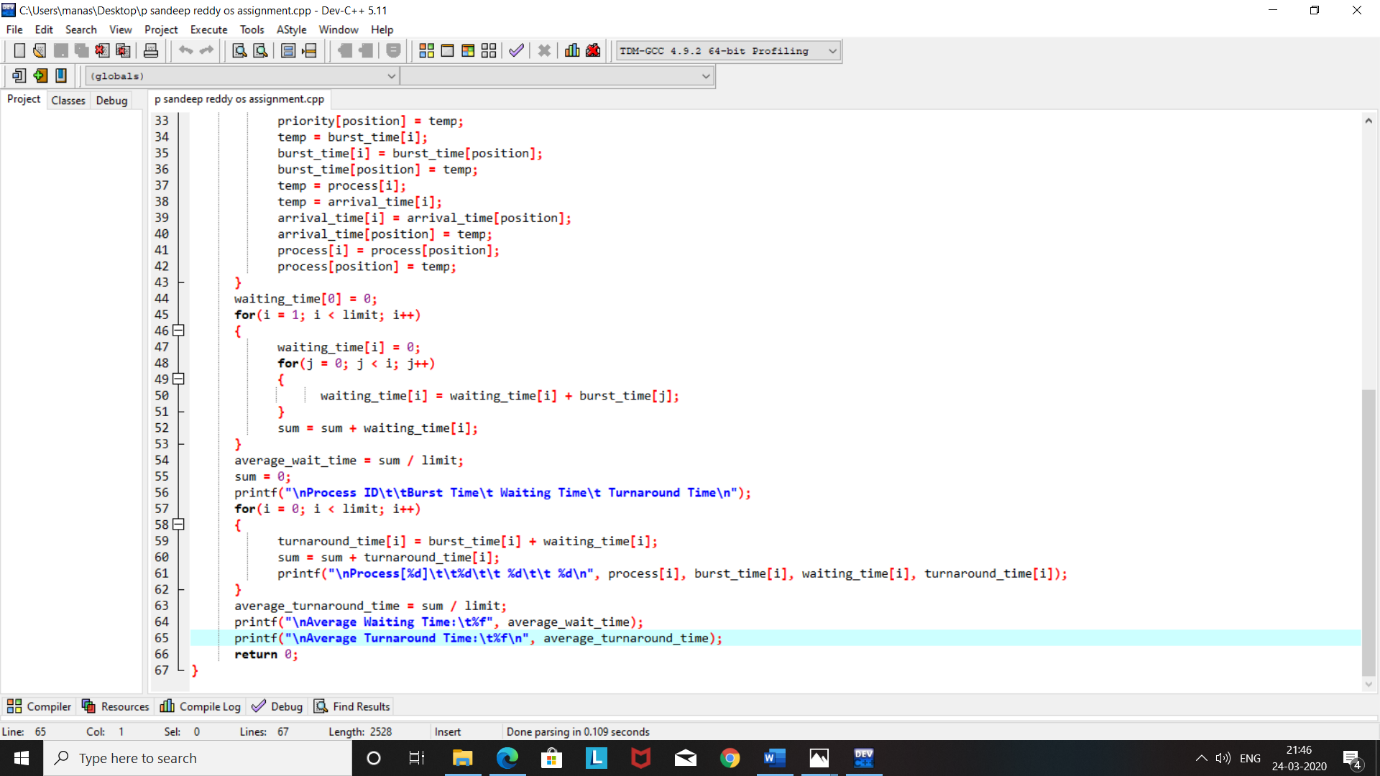
**STEP4:-** The CPU then moves to the next process and the same method is followed.

**STEP5:-** Same steps are repeated till all the processes are over.

**STEP6:-** At last it calculates the average waiting time and turn around time f for all the given processes.

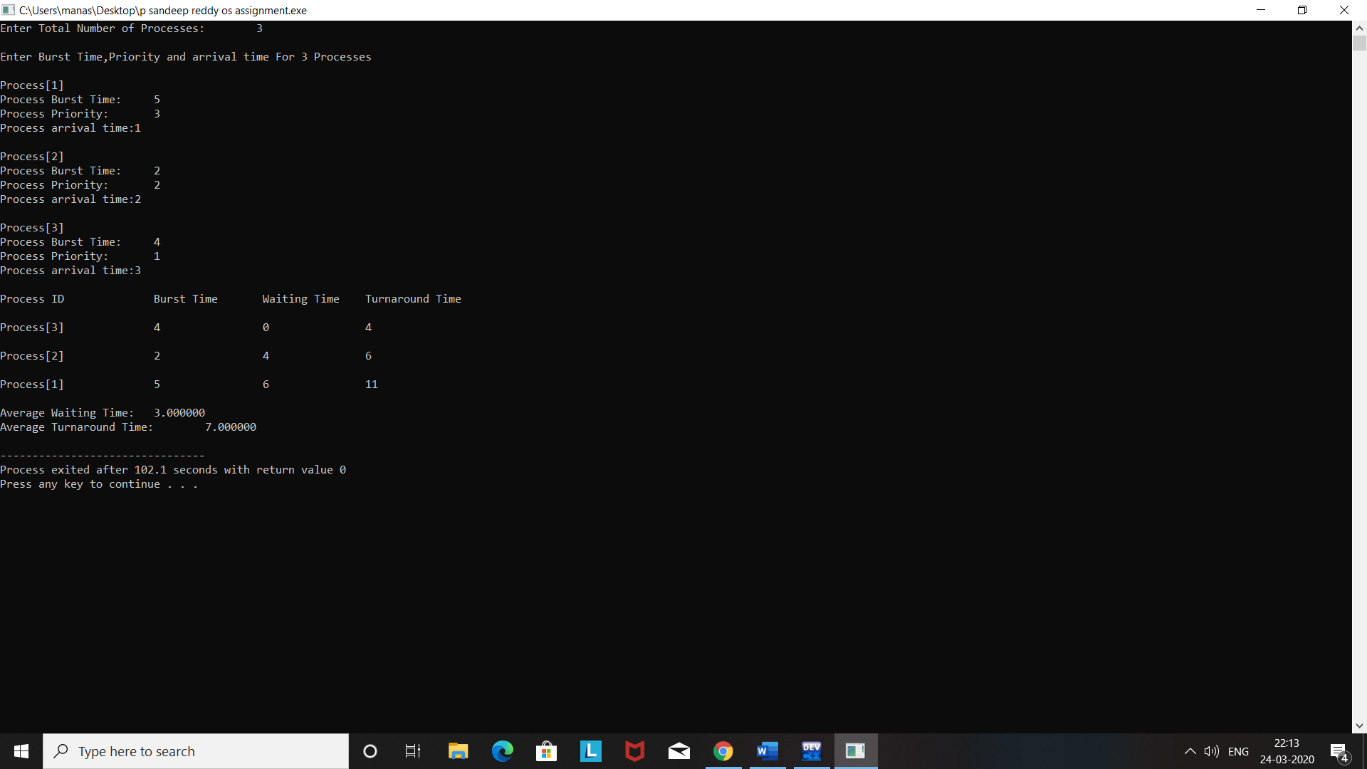
**CODE SNIPPET**

****

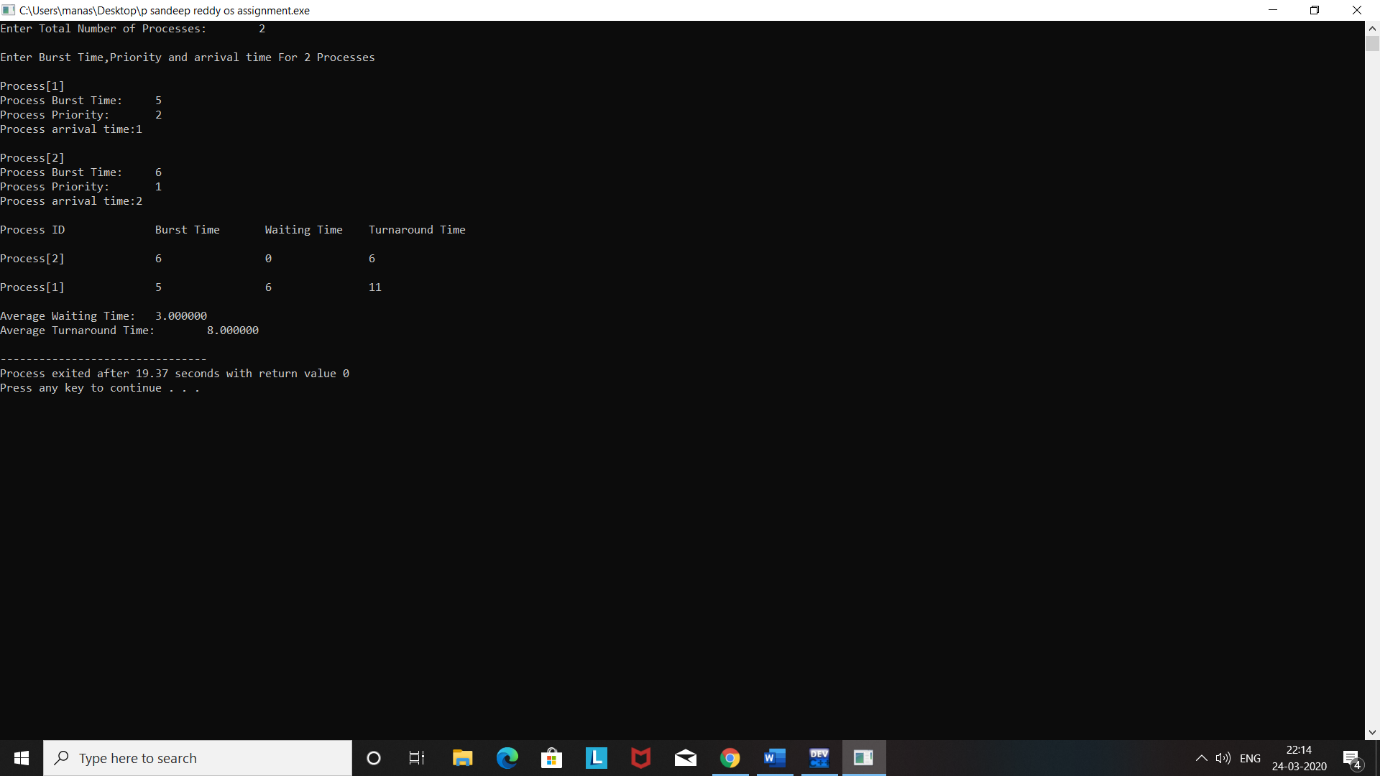
****

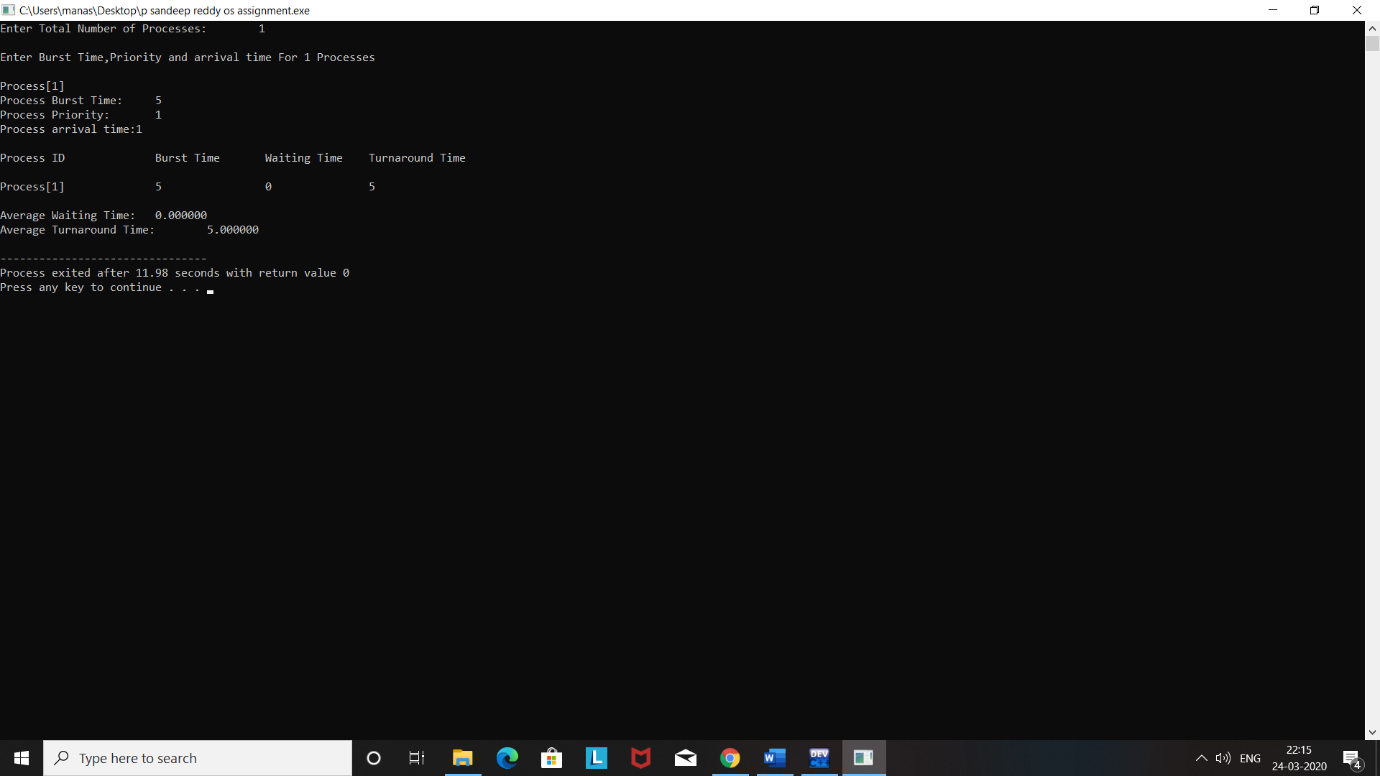
**TEST CASE**

**Test case-1:**

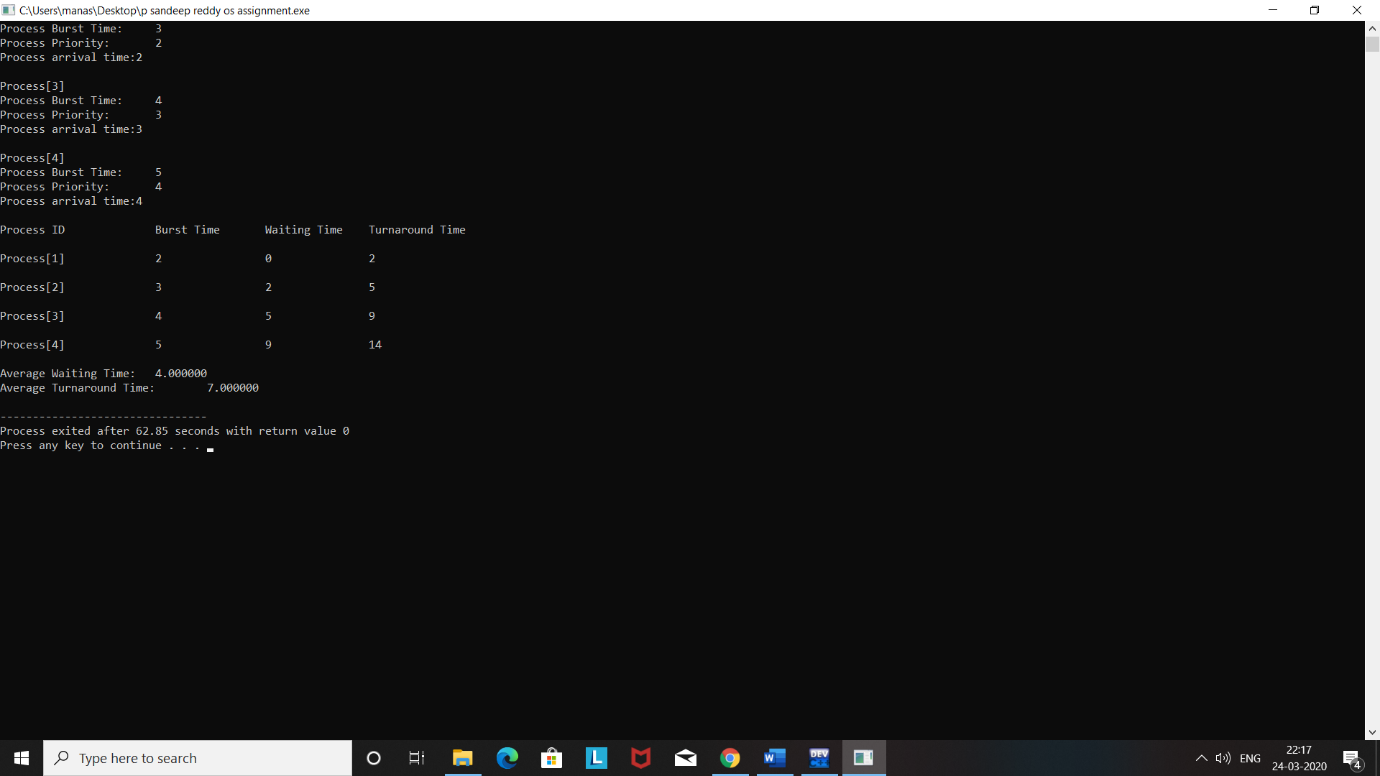


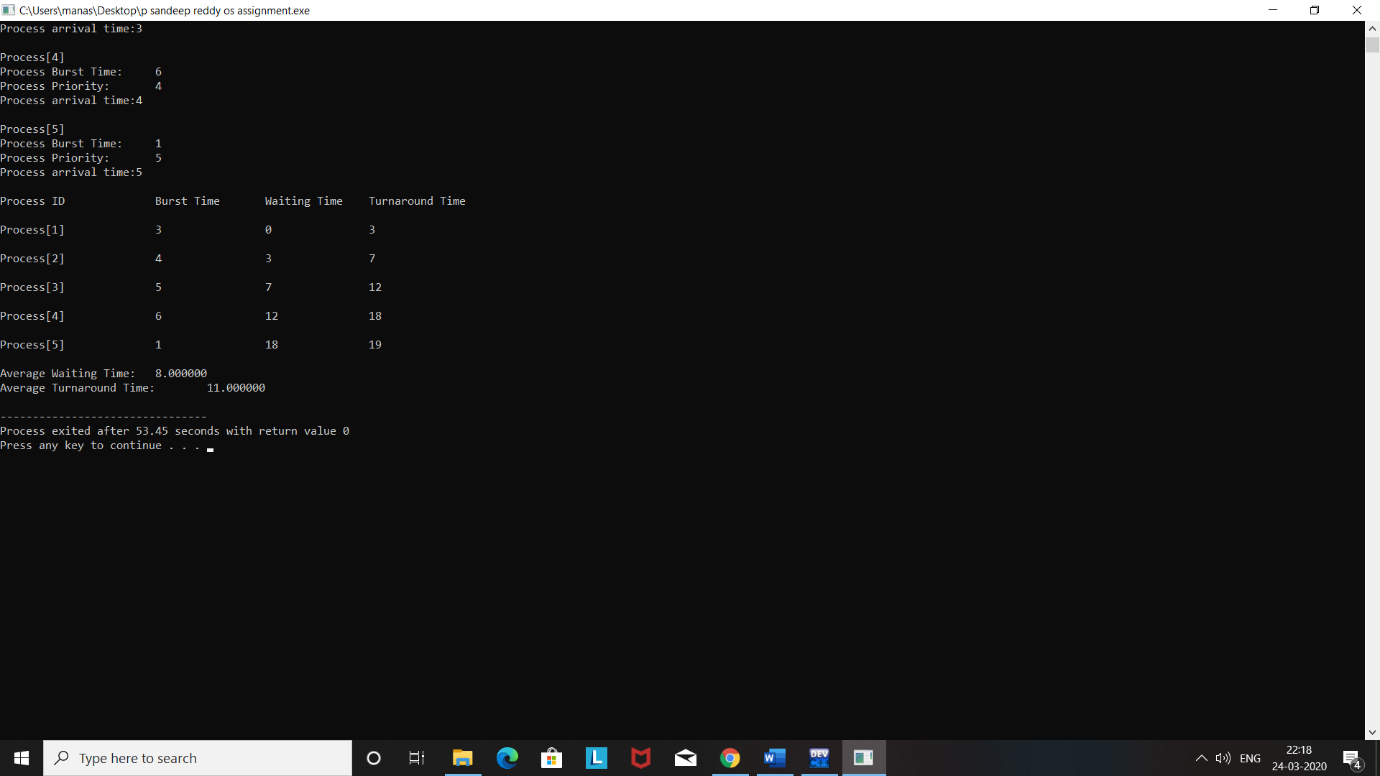
Test case-2:



Test case-3:

Test case-4:



Test case-5:

**GITHUB REPOSITORY**

**CONCLUSION**

In Priority scheduling, there is a priority number assigned to each process. In some systems, the lower the number, the higher the priority. While, in the others, the higher the number, the higher will be the priority. The Process with the higher priority among the available processes is given the CPU. There are two types of priority scheduling algorithm exists. One is **Preemptive** priority scheduling while the other is **Non Preemptive** Priority scheduling.