**Student Name: PODDUTURI AKHILESH**

**Student ID: 11306256**

**Email Address: gunaaky152@gmail.com**

**GitHub Link: https://github.com/PRMARBLES**

**CPU scheduling algorithms**

.

**ROUND ROBIN CPU SCHEDULING ALGORITHM** For round robin scheduling algorithm, read the number of processes in system of CPU burst times and the size of time. Time slices are assigned to each process equal portions and in circular order, handling processes execution. Every process to get an equal chance. Calculate the waiting time and turnaround time of each of the processes accordingly.

**PRIORITY CPU SCHEDULING ALGORITHM** For priority scheduling algorithm, read the number of processes in system of CPU burst times, and the priorities. Arrange all the jobs in order with respect to their priorities. There may be two jobs of queue with the same priority, and then FCFS approach is performed. Each process will be executed according to its priority. Calculate the waiting time and turnaround time of each of the processes accordingly.

**FCFS CPU SCHEDULING ALGORITHM** For FCFS scheduling algorithm, read the number of processes in the system of CPU burst times. The scheduling is mostly performed on basis of arrival time of the processes irrespective of other parameters. Each process will executed according to the arrival time. Calculate the waiting time and turnaround time of each of the processes accordingly

**ROUND ROBIN** **PROGRAM:**

1. Start the process

2. Accept the number of processes in the ready Queue and time quantum (or) time slice

3. For each process in the ready Q, assign the process id and accept the CPU burst time

4. Calculate the no. of time slices for each process where

No. of time slice for process (n) = burst time process (n)/time slice

5. If the burst time is less than the time slice then the no. of time slices =1.

6. Consider the ready queue is a circular Q, calculate

(a) Waiting time for process (n) = waiting time of process (n-1) + burst time of

Process (n-1) + the time difference in getting the CPU from process (n-1)

(b) Turnaround time for process (n) = waiting time of process (n) + burst time of

Process (n)+ the time difference in getting CPU from process(n).

7. Calculate

(a) Average waiting time = Total waiting Time / Number of process

(b) Average Turnaround time = Total Turnaround Time / Number of process

8. Stop the process

**PRIORITY PROGRAM:**

1. Start the process

2. Accept the number of processes in the ready Queue

3. For each process in the ready Q, assign the process id and accept the CPU burst time

4. Sort the ready queue according to the priority number.

5. Set the waiting of the first process as „0‟ and its burst time as its turnaround time

6. For each process in the Ready Q calculate

(e) Waiting time for process (n) = waiting time of process (n-1) + Burst time of process (n-1)

(f) Turnaround time for Process (n) = waiting time of Process (n) + Burst time for process (n)

7. Calculate

(g) Average waiting time = Total waiting Time / Number of process

(h) Average Turnaround time = Total Turnaround Time / Number of process

8. Stop the process

**FCFS PROGRAM:**

1. Start the process

2. Accept the number of processes in the ready Queue

3. For each process in the ready Q, assign the process id and accept the CPU burst time

4. Set the waiting of the first process as „0‟ and its burst time as its turnaround time

5. For each process in the Ready Q calculate

(c) Waiting time for process (n) = waiting time of process (n-1) + Burst time of process (n-1)

(d) Turnaround time for Process (n) = waiting time of Process (n) + Burst time for process (n)

6. Calculate

(e) Average waiting time = Total waiting Time / Number of process

(f) Average Turnaround time = Total Turnaround Time / Number of process

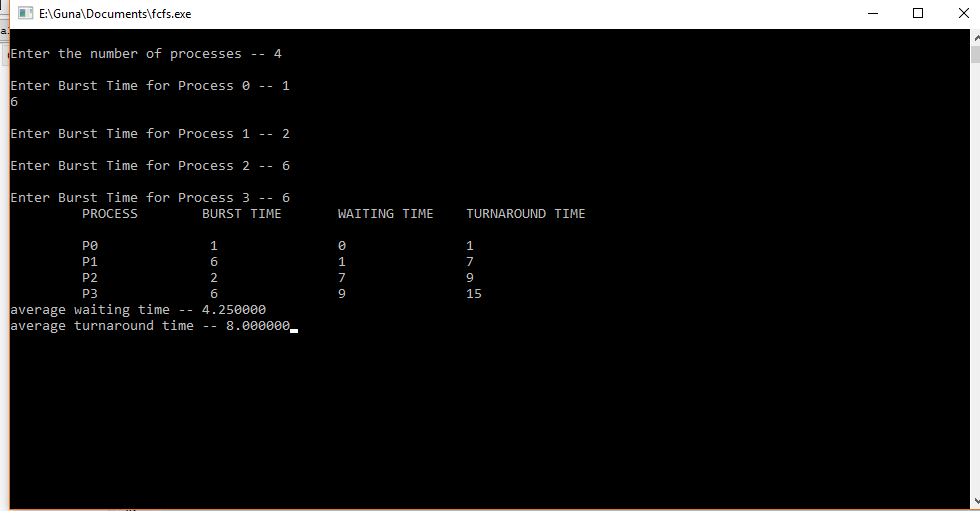
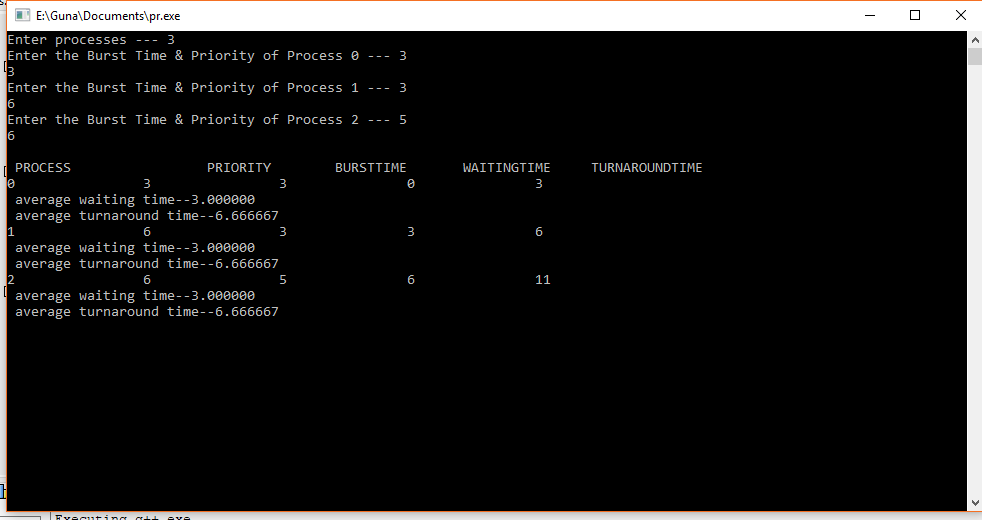
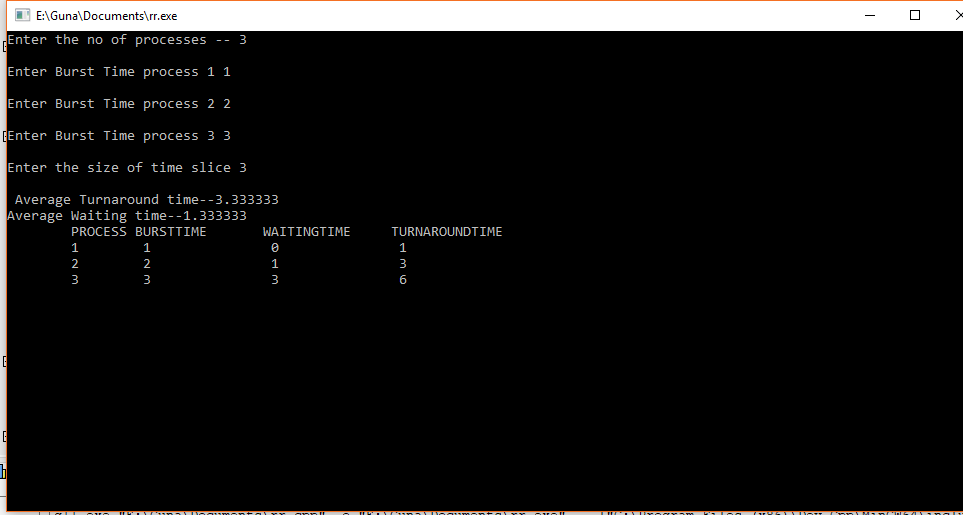
7. Stop the process

**Line complexity:**

**Round Robin: O(1)**

**Priority: O(log N)**

**Fcfs: O(log N)**

**Overall Complexity:  **

**Boundary Conditions :**

**Round Robin**: Processors sharing CPU equally

**Priority**: Starvation

**FCFS**: Convoy Effect