LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

Course : Operating System (CSE316)

- Student Name : Rohit Kumar
- Student ID : 11810740
- Email Address : rs2001289@gmail.com
- GitHub Link : https://github.com/Rohitszx/Operating-System-Project
- Roll No : 29
- Section : K18SP

1.1 Problem Statement :

Design a scheduling program that is capable of scheduling many processes that comes in at some time interval and are allocated the CPU not more that 10-time units. CPU must schedule processes having short execution time first. CPU is idle for 3-time units and does not entertain any process prior this time. Scheduler must maintain a queue that keeps the order of execution of all the processes. Compute average waiting and turnaround time.

1.2 Description :

This program is based on the Shortest Remaining Time First (SRTF). It is the pre-emptive version of Shortest Job Next (SJN) algorithm, where the processor is allocated to the job closest to completion. This algorithm requires advanced concept and knowledge of CPU time required to process the job in an interactive system, and hence can’t be implemented there. But, in a batch system where it is desirable to give preference to short jobs, SRT algorithm is used.

However, SRT involves more overheads than SJN, as the OS is required to frequently monitor the CPU time of the jobs in the READY queue and perform context switching.

If a new process arrives with CPU burst length less than remaining time of current executing process, pre-empt. This scheme is known as the Shortest-Remaining-Time-First (SRTF). If a new process arrives with a shorter burst time than remaining of current process then schedule new process. Further reduces average waiting time and average response time.

1.3 Algorithm used:

Shortest Remaining Job First:

1. Sort all the process according to the arrival time.
2. Then select that process which has minimum arrival time and minimum Burst time.
3. After completion of process make a pool of process which after till the completion of previous process and select that process among the pool which is having minimum Burst time.

1.4 Implementation of Algorithm-

```
1- Traverse until all process gets completely
   executed.
   a) Find process with minimum remaining time at
     every single time lap.
   b) Reduce its time by 1.
   c) Check if its remaining time becomes 0
   d) Increment the counter of process completion.
   e) Completion time of current process =  current_time +1;
   e) Calculate waiting time for each completed process.
   wt[i]= Completion time - arrival_time-burst_time
   f)Increment time lap by one.
2- Find turnaround time (waiting_time+burst_time).
```

1.5 Overall Complexity

Among all the available processes, the process with smallest burst time has to be selected.

Min heap is a suitable data structure where root element contains the process with least burst time.

In min heap, each process will be added and deleted exactly once.

Adding an element takes log(n) time and deleting an element takes log(n) time.

Thus, for n processes, **time complexity = n x 2log(n) = nlog(n)**

1.6 Code Snippet :

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
int main()
{
 int count,i,j,m=0,n,y=0;
  int time,remain=0,min,flag=0;
  int wait_time=0,turn_a_time=0,a_time[10],b_time[10],p[10],z[10];
  float k=0,x=0;
  printf("NOTE: ");
  printf("\n   1. Arrival time should be greater than 2 as CPU remains idle for  first 
3 secs.\n");
  printf("   2. Burst time should be less tha 10\n");
  printf("\nEnter number of Process: ");
  scanf("%d",&n);
  for(count=0;count<n;count++)
  {
    printf("\nEnter Arrival Time and Burst Time for Process  %d :",count+1);
    scanf("%d",&a_time[count]);
    scanf("%d",&b_time[count]);
  }
```

```
for(i=0;i<n;i++)
{
        if(a_time[i]==0)
        {
                printf("\nINVALID ARRIVAL TIME\n");
                getch();
                exit(1);
        }
}
for(i=0;i<n;i++)
{
        if(a_time[i]<3)
        {
                printf("\nINVALID Arrival Time it should be greater than 3\n");
                getch();
                exit(1);
        }
}
  printf("\n\nProcess\t\tTurnaround Time\t\tWaiting Time\n");
  printf("-------------------------------------------------------\n");
```

```
for(i=0;i<n;i++)
{
        m=m+b_time[i];
}
min=m;
time=m;
for(i=0;i<n;i++)
{
        if(a_time[i]<time)
        {
                time=a_time[i];
        }
}
for(i=time;i<=m;i=i+b_time[j])
{
        min=m;
        remain=0;
        flag=0;

        for(count=0;count<n;count++)
        {
```

```
    if(a_time[count]<=i)
    {
        if(b_time[count]<min)
        {
            min = b_time[count];
            j=count;
            flag=1;
        }
        remain=1;
    }
}
if(flag==1&&remain==1)
{
    wait_time=i-a_time[j];
    turn_a_time=wait_time+b_time[j];
    printf("P[%d]\t\t%d\t\t%d\n\n",j+1,turn_a_time,wait_time);
    k=k+wait_time;
    x=x+turn_a_time;
    a_time[j]=m+1;
    p[y]=j+1;
    z[y]=i;
    y++;
```

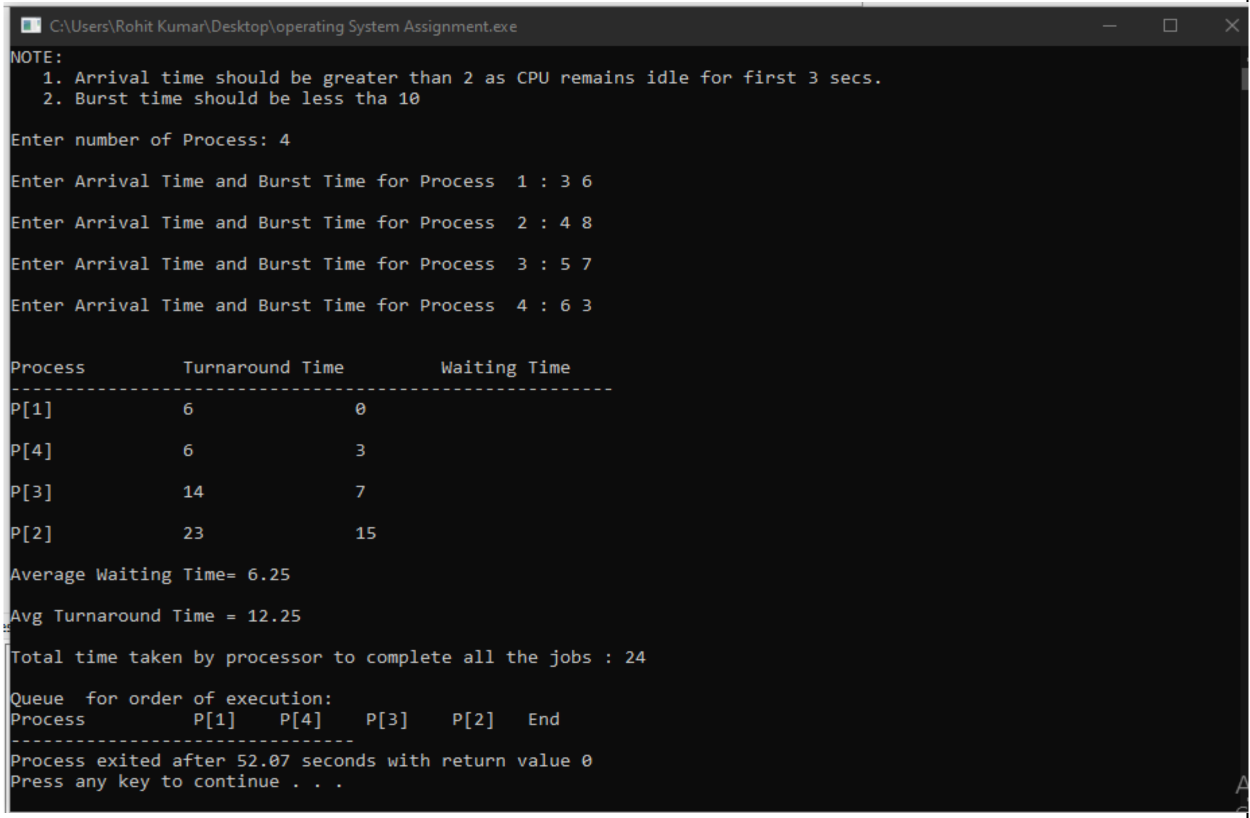
```
        }
}
printf("Average Waiting Time= %.2f\n\n",k/n);
printf("Avg Turnaround Time = %.2f\n\n",x/n);
printf("Total time taken by processor to complete all the jobs : %d\n\n",m);
printf("Queue  for order of execution:\n");
printf("Process                  ");
for(i=0;i<n;i++)
{
        printf(" P[%d]   ",p[i]);
        if(i==(n-1))
        {
                printf("End");
        }
}
  return 0;
}
```

1.7 - Test Case :

1. Process Details:-

Process Name	Arrival Time	Burst Time
P1	3	6
P2	4	8
P3	5	7
P4	6	3

Output of program :



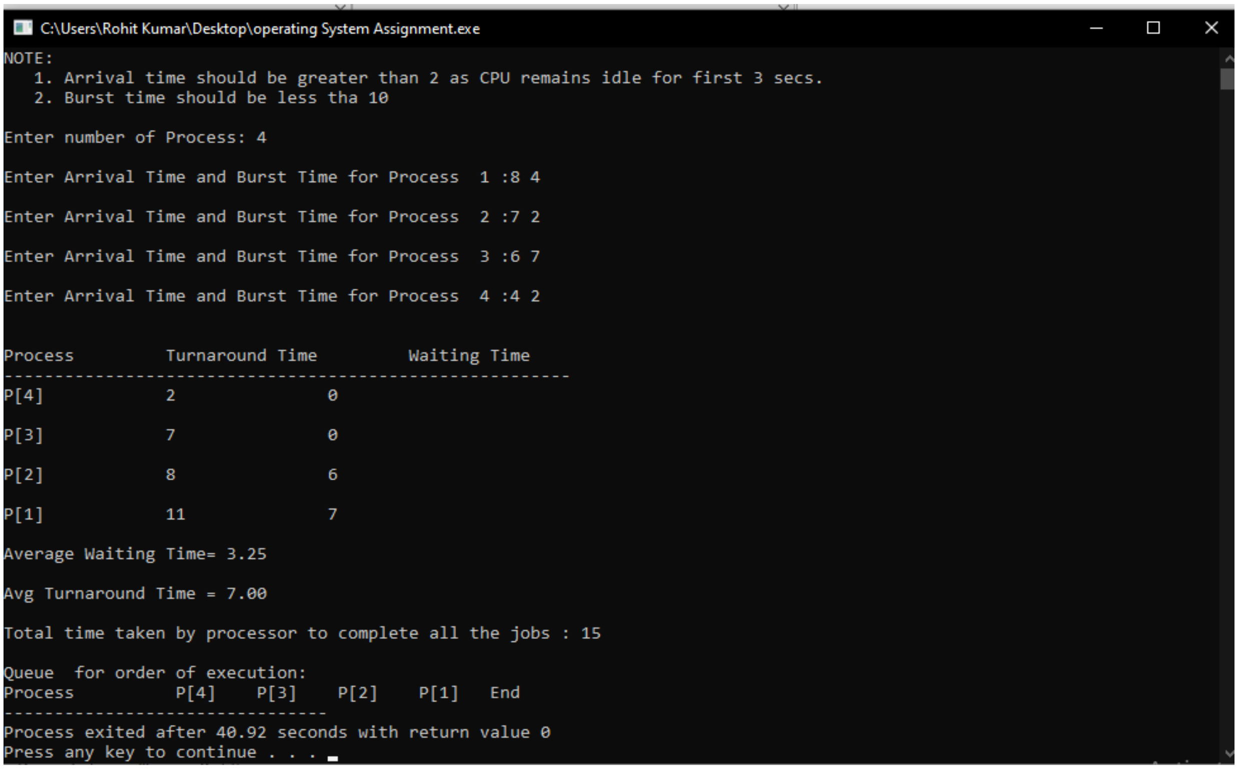
C:\Users\Rohit Kumar\Desktop\operating System Assignment.exe

```
NOTE:
   1. Arrival time should be greater than 2 as CPU remains idle for first 3 secs.
   2. Burst time should be less tha 10

Enter number of Process: 4

Enter Arrival Time and Burst Time for Process  1 : 3 6

Enter Arrival Time and Burst Time for Process  2 : 4 8

Enter Arrival Time and Burst Time for Process  3 : 5 7

Enter Arrival Time and Burst Time for Process  4 : 6 3


Process         Turnaround Time         Waiting Time
--------------------------------------------------------
P[1]            6               0

P[4]            6               3

P[3]            14              7

P[2]            23              15

Average Waiting Time= 6.25

Avg Turnaround Time = 12.25

Total time taken by processor to complete all the jobs : 24

Queue  for order of execution:
Process          P[1]    P[4]    P[3]    P[2]   End
--------------------------------
Process exited after 52.07 seconds with return value 0
Press any key to continue . . .
```

2. Process Details:-

Process Name	Arrival Time	Burst Time
P1	8	4
P2	7	2
P3	6	7
P4	4	2

Output of program :

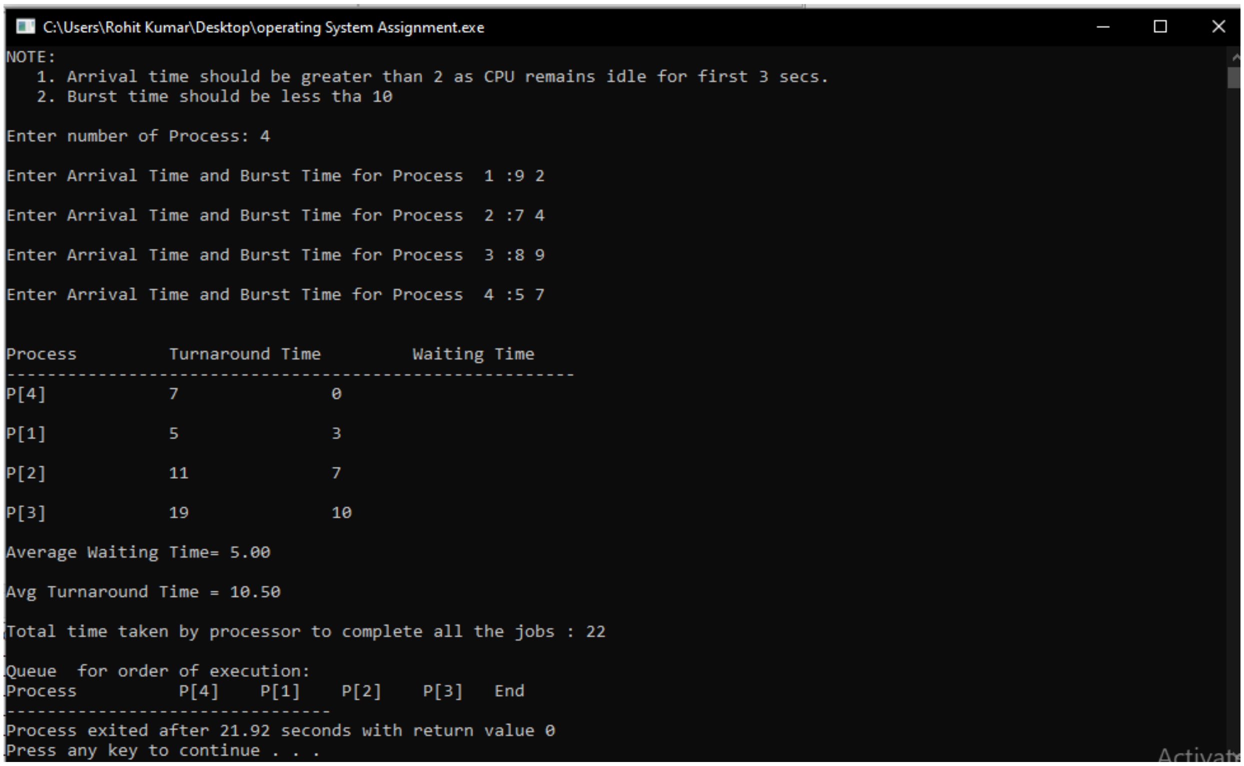

C:\Users\Rohit Kumar\Desktop\operating System Assignment.exe

```
NOTE:
   1. Arrival time should be greater than 2 as CPU remains idle for first 3 secs.
   2. Burst time should be less tha 10

Enter number of Process: 4

Enter Arrival Time and Burst Time for Process  1 :8 4

Enter Arrival Time and Burst Time for Process  2 :7 2

Enter Arrival Time and Burst Time for Process  3 :6 7

Enter Arrival Time and Burst Time for Process  4 :4 2


Process         Turnaround Time         Waiting Time
--------------------------------------------------------
P[4]            2               0

P[3]            7               0

P[2]            8               6

P[1]            11              7

Average Waiting Time= 3.25

Avg Turnaround Time = 7.00

Total time taken by processor to complete all the jobs : 15

Queue  for order of execution:
Process          P[4]    P[3]    P[2]    P[1]   End
--------------------------------
Process exited after 40.92 seconds with return value 0
Press any key to continue . . .
```

3. Process Details:-

Process Name	Arrival Time	Burst Time
P1	9	2
P2	7	4
P3	8	9
P4	5	7

Output of program :

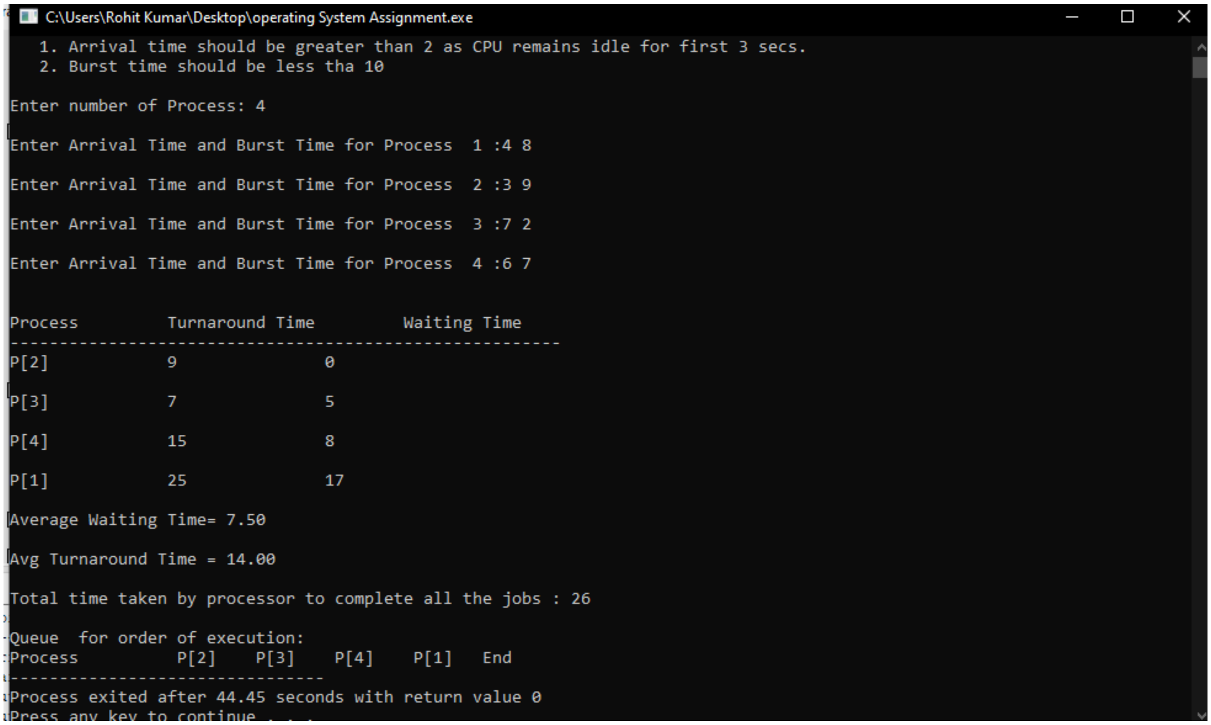

C:\Users\Rohit Kumar\Desktop\operating System Assignment.exe

```
NOTE:
   1. Arrival time should be greater than 2 as CPU remains idle for first 3 secs.
   2. Burst time should be less tha 10

Enter number of Process: 4

Enter Arrival Time and Burst Time for Process  1 :9 2

Enter Arrival Time and Burst Time for Process  2 :7 4

Enter Arrival Time and Burst Time for Process  3 :8 9

Enter Arrival Time and Burst Time for Process  4 :5 7


Process         Turnaround Time         Waiting Time
--------------------------------------------------------
P[4]            7               0

P[1]            5               3

P[2]            11              7

P[3]            19              10

Average Waiting Time= 5.00

Avg Turnaround Time = 10.50

Total time taken by processor to complete all the jobs : 22

Queue  for order of execution:
Process          P[4]    P[1]    P[2]    P[3]   End
--------------------------------
Process exited after 21.92 seconds with return value 0
Press any key to continue . . .
```

4. Process Details:-

Process Name	Arrival Time	Burst Time
P1	4	8
P2	3	9
P3	7	2
P4	6	7

Output of program :



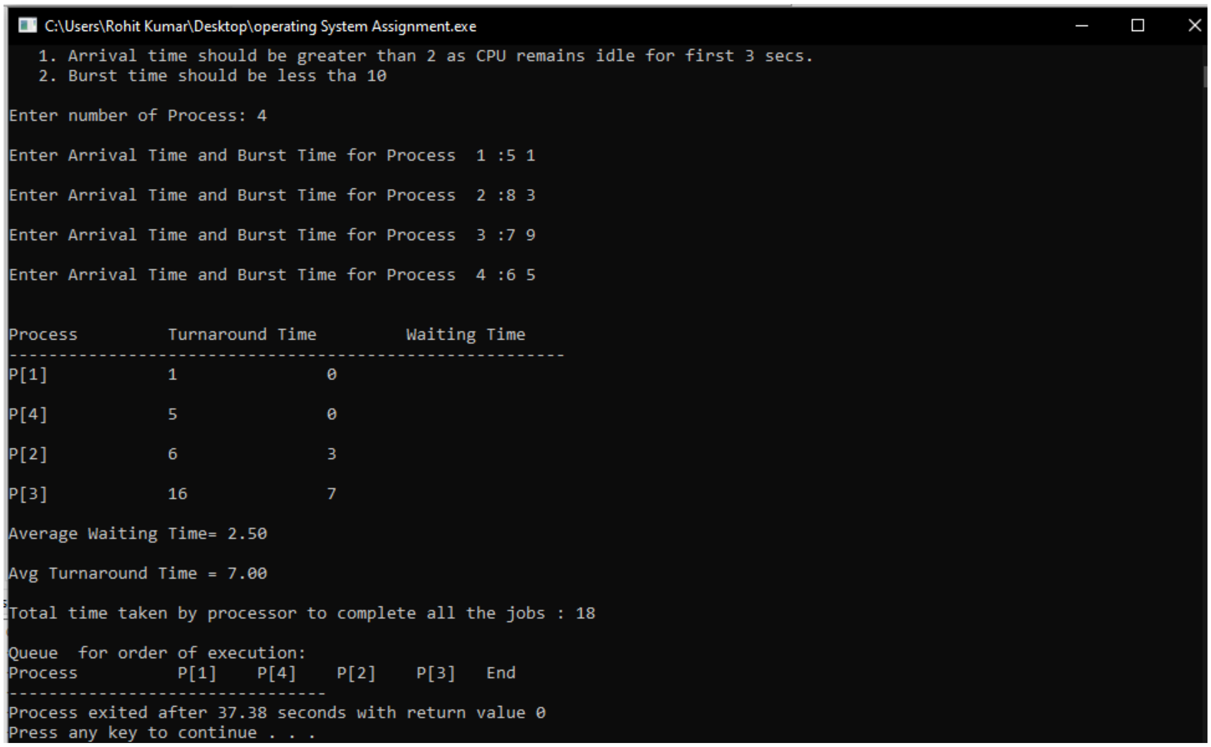
C:\Users\Rohit Kumar\Desktop\operating System Assignment.exe

```
   1. Arrival time should be greater than 2 as CPU remains idle for first 3 secs.
   2. Burst time should be less tha 10

Enter number of Process: 4

Enter Arrival Time and Burst Time for Process  1 :4 8

Enter Arrival Time and Burst Time for Process  2 :3 9

Enter Arrival Time and Burst Time for Process  3 :7 2

Enter Arrival Time and Burst Time for Process  4 :6 7


Process         Turnaround Time         Waiting Time
--------------------------------------------------------
P[2]            9               0

P[3]            7               5

P[4]            15              8

P[1]            25              17

Average Waiting Time= 7.50

Avg Turnaround Time = 14.00

Total time taken by processor to complete all the jobs : 26

Queue  for order of execution:
Process          P[2]    P[3]    P[4]    P[1]   End
--------------------------------
Process exited after 44.45 seconds with return value 0
Press any key to continue . . .
```

5. Process Details:-

Process Name	Arrival Time	Burst Time
P1	5	1
P2	8	3
P3	7	9
P4	6	5

Output of program :


C:\Users\Rohit Kumar\Desktop\operating System Assignment.exe

```
   1. Arrival time should be greater than 2 as CPU remains idle for first 3 secs.
   2. Burst time should be less tha 10

Enter number of Process: 4

Enter Arrival Time and Burst Time for Process  1 :5 1

Enter Arrival Time and Burst Time for Process  2 :8 3

Enter Arrival Time and Burst Time for Process  3 :7 9

Enter Arrival Time and Burst Time for Process  4 :6 5


Process         Turnaround Time         Waiting Time
--------------------------------------------------------
P[1]            1               0

P[4]            5               0

P[2]            6               3

P[3]            16              7

Average Waiting Time= 2.50

Avg Turnaround Time = 7.00

Total time taken by processor to complete all the jobs : 18

Queue  for order of execution:
Process          P[1]    P[4]    P[2]    P[3]   End
--------------------------------
Process exited after 37.38 seconds with return value 0
Press any key to continue . . .
```

1.8 Github Link :

□ https://github.com/Rohitszx/Operating-System-Project