COMSATS UNIVERSITY ISLAMABAD
ATTOCK CAMPUS

Lab Report 7 : Operating System

Submitted to : Sir Fayyaz Ali

Group Members	Muaaz Shoaib FA20-BCS-074 Shahzeb Shaheen FA20-BCS-040

Rubrics Assessment Sheet for Operating System

Lab #:	**Lab no 7**
Lab Title:	**Scheduling Algorithms**

Submitted by:

Names	**Registration**
Muaaz Shoaib	**FA20-BCS-074**
Shahzeb Shaheen	**FA20-BCS-040**

Rubrics name & number		Marks	
		ln-Lab	Post lab
Engineering Knowledge	*R2:* Use of Engineering Knowledge and follow Experiment Procedures: *Ability to follow experimental procedure,control variables,and record Procedural steps on lab report.*		
Problem Analysis	*R6: Experimental Data Analysis :* *Ability to interept findings,compare them to values in the literature,identify weaknesses and limitations*		
Design	*RS: Best Coding Staudards:* *Ability lo follow the* coding standards and programming practices		
Modem Tools Usage	*R9: Undcrsta11d Tools: Ability to describe and explain the principles behind applicability of engineering tools.*		
Individual and Teamwork	*R9:Management of Team Work:* *Ability to appreciate, understand and work multidisciplinary team members*		

Rubrics #	R2	R6	RS	R9	Rl3
Jn -Lab					
Post- Lab					

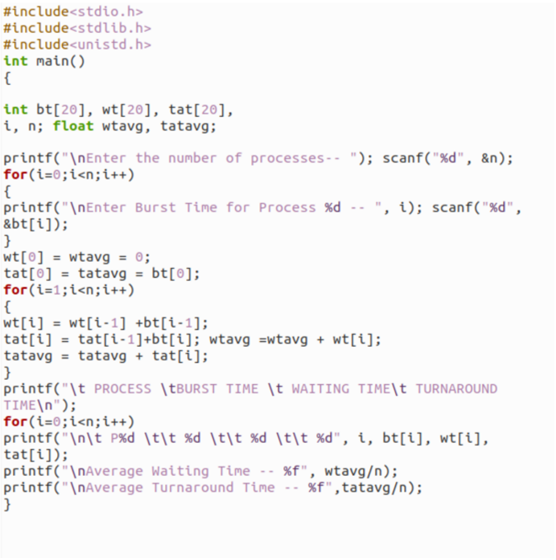
Description :

Scheduling Algorithms :

It tells us how much CPU time is allocated to the processes.

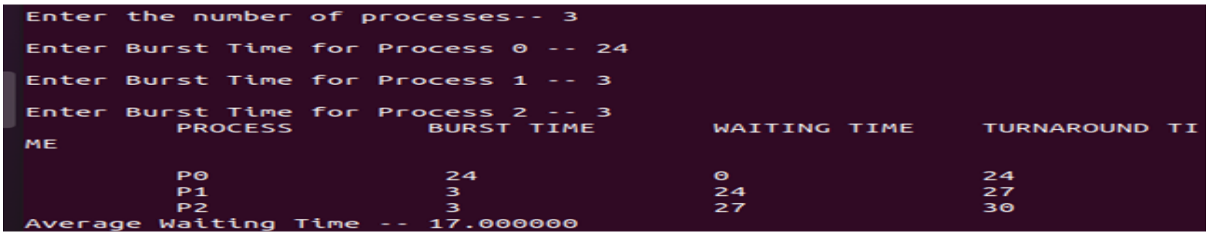
Q: Write a C program to simulate the following non-preemptive CPU Scheduling Algorithms to find turnaround time and waiting time for above problems.

1 FCFS Algorithm

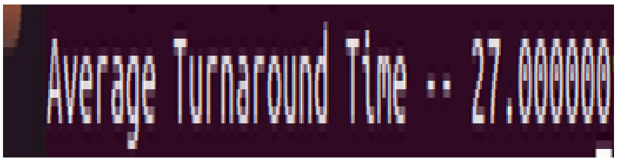


```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main()
{

int bt[20], wt[20], tat[20],
i, n; float wtavg, tatavg;

printf("\nEnter the number of processes-- "); scanf("%d", &n);
for(i=0;i<n;i++)
{
printf("\nEnter Burst Time for Process %d -- ", i); scanf("%d",
&bt[i]);
}
wt[0] = wtavg = 0;
tat[0] = tatavg = bt[0];
for(i=1;i<n;i++)
{
wt[i] = wt[i-1] +bt[i-1];
tat[i] = tat[i-1]+bt[i]; wtavg =wtavg + wt[i];
tatavg = tatavg + tat[i];
}
printf("\t PROCESS \tBURST TIME \t WAITING TIME\t TURNAROUND
TIME\n");
for(i=0;i<n;i++)
printf("\n\t P%d \t\t %d \t\t %d \t\t %d", i, bt[i], wt[i],
tat[i]);
printf("\nAverage Waiting Time -- %f", wtavg/n);
printf("\nAverage Turnaround Time -- %f",tatavg/n);
}
```

Output

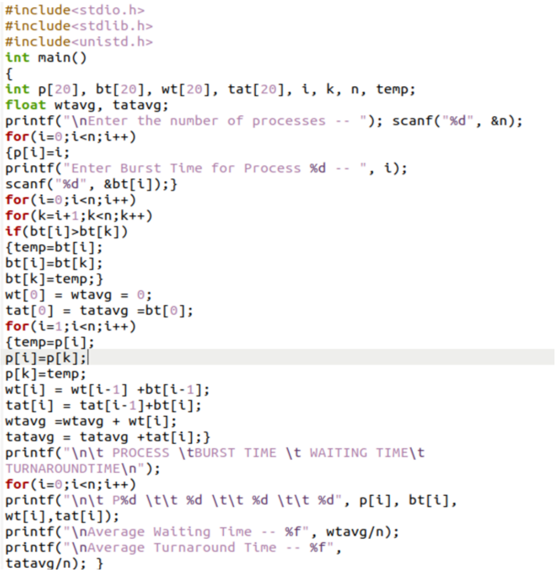


```
Enter the number of processes-- 3
Enter Burst Time for Process 0 -- 24
Enter Burst Time for Process 1 -- 3
Enter Burst Time for Process 2 -- 3
          PROCESS          BURST TIME         WAITING TIME      TURNAROUND TI
ME

          P0                24                0                 24
          P1                3                 24                27
          P2                3                 27                30
Average Waiting Time -- 17.000000
```



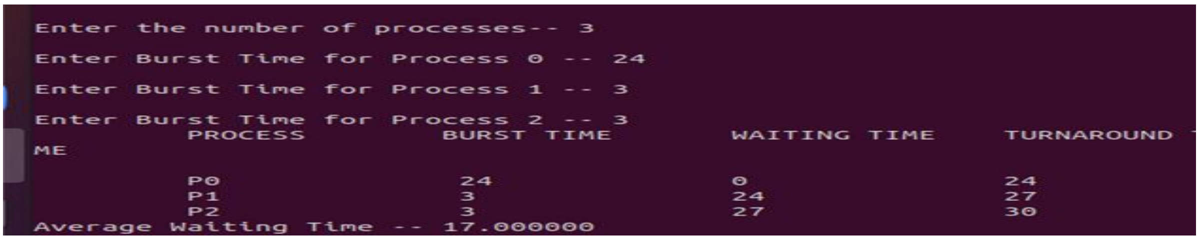
```
Average Turnaround Time -- 27.000000
```

2 SJF Algorithm

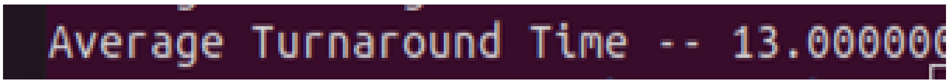


```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main()
{
int p[20], bt[20], wt[20], tat[20], i, k, n, temp;
float wtavg, tatavg;
printf("\nEnter the number of processes -- "); scanf("%d", &n);
for(i=0;i<n;i++)
{p[i]=i;
printf("Enter Burst Time for Process %d -- ", i);
scanf("%d", &bt[i]);}
for(i=0;i<n;i++)
for(k=i+1;k<n;k++)
if(bt[i]>bt[k])
{temp=bt[i];
bt[i]=bt[k];
bt[k]=temp;}
wt[0] = wtavg = 0;
tat[0] = tatavg =bt[0];
for(i=1;i<n;i++)
{temp=p[i];
p[i]=p[k];
p[k]=temp;
wt[i] = wt[i-1] +bt[i-1];
tat[i] = tat[i-1]+bt[i];
wtavg =wtavg + wt[i];
tatavg = tatavg +tat[i];}
printf("\n\t PROCESS \tBURST TIME \t WAITING TIME\t
TURNAROUNDTIME\n");
for(i=0;i<n;i++)
printf("\n\t P%d \t\t %d \t\t %d \t\t %d", p[i], bt[i],
wt[i],tat[i]);
printf("\nAverage Waiting Time -- %f", wtavg/n);
printf("\nAverage Turnaround Time -- %f",
tatavg/n); }
```

Output :

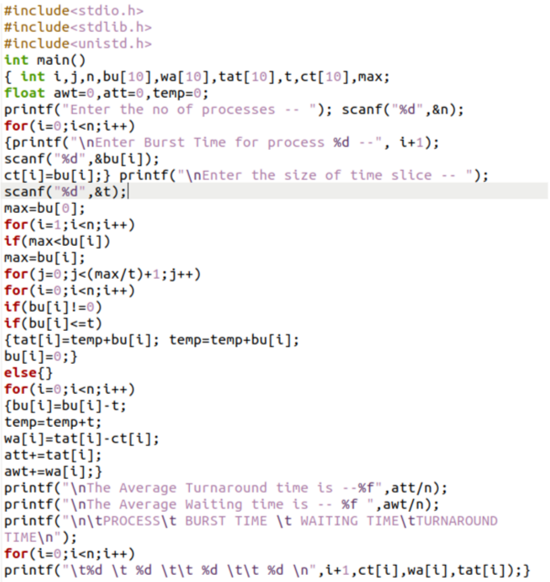


```
Enter the number of processes-- 3
Enter Burst Time for Process 0 -- 24
Enter Burst Time for Process 1 -- 3
Enter Burst Time for Process 2 -- 3
          PROCESS          BURST TIME        WAITING TIME      TURNAROUND T
ME

          P0                24                0                 24
          P1                3                 24                27
          P2                3                 27                30
Average Waiting Time -- 17.000000
```



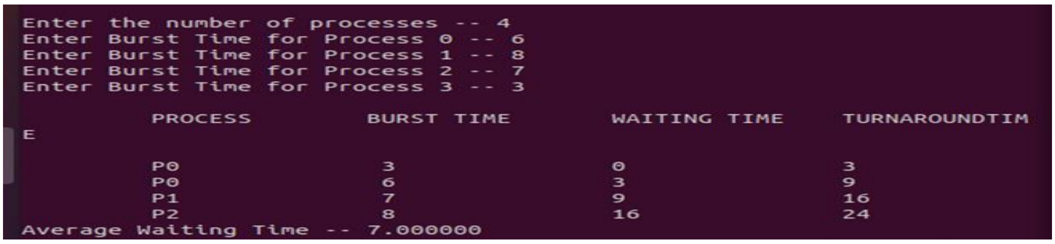
```
Average Turnaround Time -- 13.00000
```

3 Round Robin Algorithm



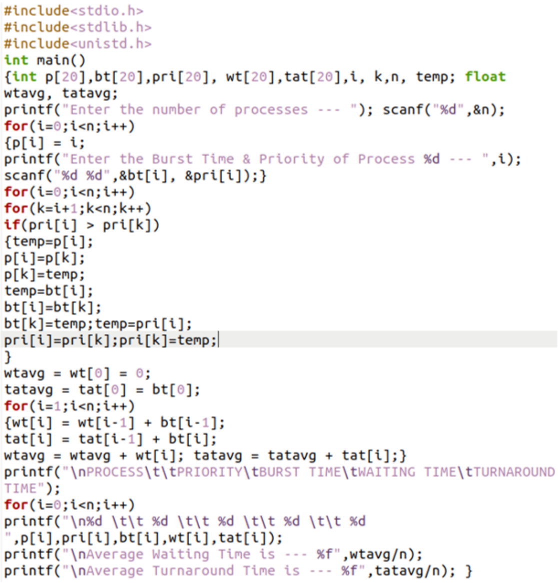
```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main()
{ int i,j,n,bu[10],wa[10],tat[10],t,ct[10],max;
float awt=0,att=0,temp=0;
printf("Enter the no of processes -- "); scanf("%d",&n);
for(i=0;i<n;i++)
{printf("\nEnter Burst Time for process %d --", i+1);
scanf("%d",&bu[i]);
ct[i]=bu[i];} printf("\nEnter the size of time slice -- ");
scanf("%d",&t);
max=bu[0];
for(i=1;i<n;i++)
if(max<bu[i])
max=bu[i];
for(j=0;j<(max/t)+1;j++)
for(i=0;i<n;i++)
if(bu[i]!=0)
if(bu[i]<=t)
{tat[i]=temp+bu[i]; temp=temp+bu[i];
bu[i]=0;}
else{}
for(i=0;i<n;i++)
{bu[i]=bu[i]-t;
temp=temp+t;
wa[i]=tat[i]-ct[i];
att+=tat[i];
awt+=wa[i];}
printf("\nThe Average Turnaround time is --%f",att/n);
printf("\nThe Average Waiting time is -- %f ",awt/n);
printf("\n\tPROCESS\t BURST TIME \t WAITING TIME\tTURNAROUND
TIME\n");
for(i=0;i<n;i++)
printf("\t%d \t %d \t\t %d \t\t %d \n",i+1,ct[i],wa[i],tat[i]);}
```

Output :



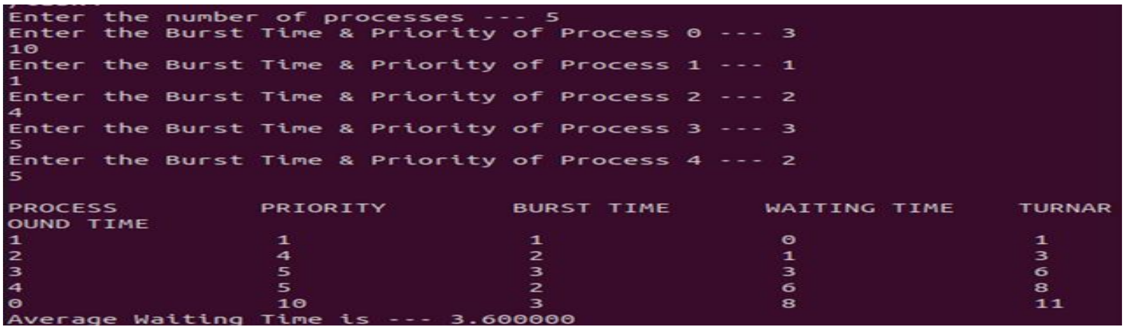
```
Enter the number of processes -- 4
Enter Burst Time for Process 0 -- 6
Enter Burst Time for Process 1 -- 8
Enter Burst Time for Process 2 -- 7
Enter Burst Time for Process 3 -- 3

          PROCESS          BURST TIME          WAITING TIME     TURNAROUNDTIM
E

          P0                3                  0                3
          P0                6                  3                9
          P1                7                  9                16
          P2                8                  16               24
Average Waiting Time -- 7.000000
```

4 Priority Algorithm



```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main()
{int p[20],bt[20],pri[20], wt[20],tat[20],i, k,n, temp; float
wtavg, tatavg;
printf("Enter the number of processes --- "); scanf("%d",&n);
for(i=0;i<n;i++)
{p[i] = i;
printf("Enter the Burst Time & Priority of Process %d --- ",i);
scanf("%d %d",&bt[i], &pri[i]);}
for(i=0;i<n;i++)
for(k=i+1;k<n;k++)
if(pri[i] > pri[k])
{temp=p[i];
p[i]=p[k];
p[k]=temp;
temp=bt[i];
bt[i]=bt[k];
bt[k]=temp;temp=pri[i];
pri[i]=pri[k];pri[k]=temp;
}
wtavg = wt[0] = 0;
tatavg = tat[0] = bt[0];
for(i=1;i<n;i++)
{wt[i] = wt[i-1] + bt[i-1];
tat[i] = tat[i-1] + bt[i];
wtavg = wtavg + wt[i]; tatavg = tatavg + tat[i];}
printf("\nPROCESS\t\tPRIORITY\tBURST TIME\tWAITING TIME\tTURNAROUND
TIME");
for(i=0;i<n;i++)
printf("\n%d \t\t %d \t\t %d \t\t %d \t\t %d
",p[i],pri[i],bt[i],wt[i],tat[i]);
printf("\nAverage Waiting Time is --- %f",wtavg/n);
printf("\nAverage Turnaround Time is --- %f",tatavg/n); }
```

Output :



```
Enter the number of processes --- 5
Enter the Burst Time & Priority of Process 0 --- 3
10
Enter the Burst Time & Priority of Process 1 --- 1
1
Enter the Burst Time & Priority of Process 2 --- 2
4
Enter the Burst Time & Priority of Process 3 --- 3
5
Enter the Burst Time & Priority of Process 4 --- 2
5

PROCESS         PRIORITY        BURST TIME      WAITING TIME    TURNAR
OUND TIME
1                1               1               0               1
2                4               2               1               3
3                5               3               3               6
4                5               2               6               8
0                10              3               8               11
Average Waiting Time is --- 3.600000
```