Name: Devanshu Surana

Roll No.: 12

Panel: C**, Batch:** C1

OS Lab 3 Code:

Title: CPU Scheduling Algorithms (FCFS and Round Robin)

FCFS (Non-Preemptive):

```
#include <stdio.h>
int main()
{
int n,AT[20],BT[20],ST[20],CT[20],WT[20],TAT[20];
float total_tt =0;
float total_wt =0;
printf("How many process to be
created?\n"); scanf("%d", &n);
printf("Enter the arrival time and burst time of processes\n");
for(int i=0; i<n; i++)
{
 printf("Arrival time of process %d: ",
 i); scanf("%d", &AT[i]);
```

```
 printf("Burst time of process %d: ", i);
scanf("%d", &BT[i]); }


int i;
ST[0]=AT[0];
```

```
for(i=0;i<n;i++)
{
 CT[i]=ST[i]+BT[i];
 ST[i+1]=CT[i];
 TAT[i]=CT[i]-AT[i];
 WT[i]=TAT[i]-BT[i];
}
 for(i=0;i<n;i++)
 {
 printf("AT\tBT\tCT\tTAT\tWT\t\n");
 for(i=0;i<n;i++)
 {
 printf("%d\t%d\t%d\t%d\t%d\n",AT[i],BT[i],CT[i],TAT[i],WT[i]);
 total_tt = total_tt + TAT[i];
 total_wt = total_wt + WT[i];
 }
 float avg_tt = total_tt/n;
 float avg_wt = total_wt/n;
 printf("\nAverage waiting time: %f\n", avg_wt);
 printf("Average turnaround time: %f\n", avg_tt);
 return 0;
 }
 return 0;
}
```

OUTPUT:

```
Output
/tmp/iROo2xibUM.o
How many process to be created?
5
Enter the arrival time and burst time of processes
Arrival time of process 0: 0
Burst time of process 0: 2
Arrival time of process 1: 1
Burst time of process 1: 6
Arrival time of process 2: 2
Burst time of process 2: 4
Arrival time of process 3: 3
Burst time of process 3: 9
Arrival time of process 4: 4
Burst time of process 4: 12
AT  BT  CT  TAT WT
0   2   2   2   0
1   6   8   7   1
2   4   12  10  6
3   9   21  18  9
4   12  33  29  17

Average waiting time: 6.600000
Average turnaround time: 13.200000
```

Round Robin (Preemptive):

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
int main()
```

```
{
int count,j,n,time1,remain,flag=0,time_quantum;
```

```
int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
printf("Enter Total Process:\n "); scanf("%d",&n);

remain=n;
for(count=0;count<n;count++)
{
printf("Enter Arrival Time Process Process Number %d :",count+1);
scanf("%d",&at[count]);

printf("Enter Burst Time for Process Process Number %d :",count+1);
scanf("%d",&bt[count]);
rt[count]=bt[count];
}
printf("Enter Time Quantum:\n");
scanf("%d",&time_quantum);
printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
for(time1=0,count=0;remain!=0;) {

if(rt[count]<=time_quantum && rt[count]>0)
{
time1+=rt[count];
rt[count]=0;
flag=1;
}
else if(rt[count]>0)
{
```

```
rt[count]-=time_quantum;
```

```
time1+=time_quantum;
}
if(rt[count]==0 && flag==1)
{
remain--;
printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time1-at[count],time1-at[count]-
bt[count]);
wait_time+=time1-at[count]-bt[count];
turnaround_time+=time1-at[count];
flag=0;
}
if(count==n-1)
count=0;
else if(at[count+1]<=time1)
count++;
else
count=0;
}
printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
printf("Avg Turnaround Time = %f",turnaround_time*1.0/n);
return 0;
}
```

OUTPUT:

```
Output
/tmp/uK4y7u1cvV.o
Enter Total Process:
 4
 Enter Arrival Time Process Process Number 1 :0
 Enter Burst Time for Process Process Number 1 :7
 Enter Arrival Time Process Process Number 2 :2
 Enter Burst Time for Process Process Number 2 :4
 Enter Arrival Time Process Process Number 3 :4
 Enter Burst Time for Process Process Number 3 :1
 Enter Arrival Time Process Process Number 4 :5
 Enter Burst Time for Process Process Number 4 :4
 Enter Time Quantum:
2
Process |Turnaround Time|Waiting Time

P[3]    |   1   |   0
P[2]    |   9   |   5
P[4]    |   8   |   4
P[1]    |   16  |   9

Average Waiting Time= 4.500000
Avg Turnaround Time = 8.500000
```