



**ASSIGNMENT-19.03.2021**

# **COURSE CODE :** 19CSE103

# **COURSE NAME:** OPERATING SYSTEMS

# **TEAM MEMBERS :**

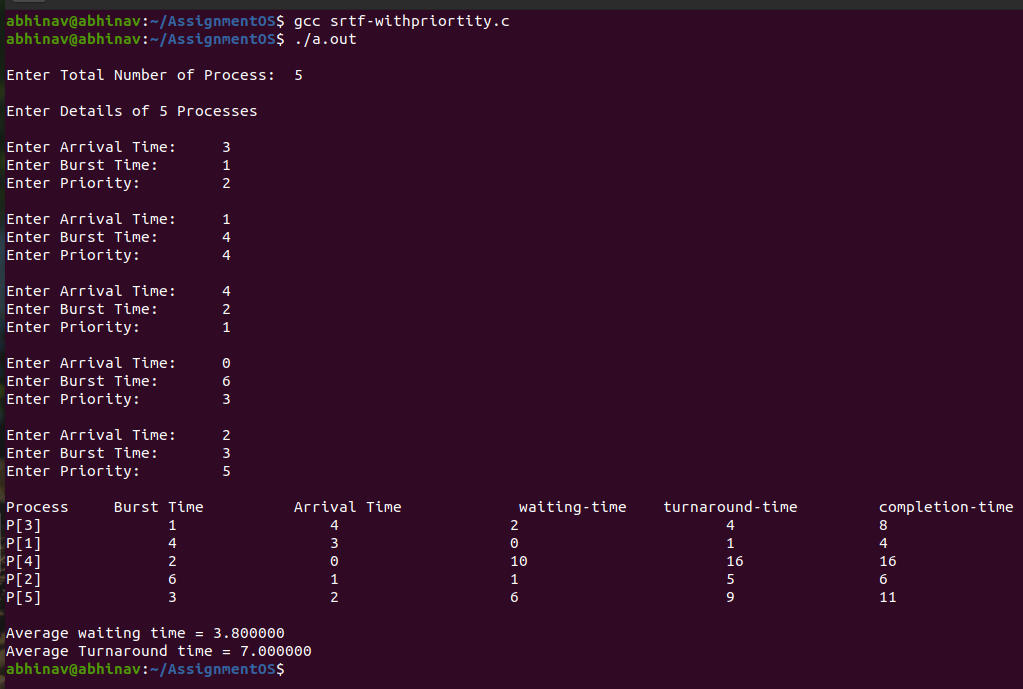
|  |  |  |
| --- | --- | --- |
| S.No | Name of the Student | Roll No. |
| 1. | **RAVELLA ABHINAV** | **CB.EN.U4CSE19453** |
| 2. | **PENUGONDA KOUSHIK** | **CB.EN.U4CSE19449** |
| 3. | **SINGADI SHANTHAN REDDY** | **CB.EN.U4CSE19459** |
| 4. | **NUSUM KARTHIK** | **CB.EN.U4CSE19444** |

**SRTF WITH PRIORITY**

**Code :**

#include <stdio.h>  
int main()  
{  
 int a[10],b[10],x[10],pr[10],p[10];  
 int i,j,smallest,count=0,time,n,pos,temp;  
 double avg=0,tt=0,end;  
  
 printf("\nEnter Total Number of Process:\t");  
 scanf("%d",&n);  
  
 printf("\nEnter Details of %d Processes\n", n);  
 for(i=0;i<n;i++)  
 {  
 printf("\nEnter Arrival Time:\t");  
 scanf("%d",&a[i]);  
 printf("Enter Burst Time:\t");  
 scanf("%d",&b[i]);  
 printf("Enter Priority:\t\t");  
 scanf("%d",&pr[i]);  
 p[i]=i+1;  
 }  
  
 b[9]=9999;  
  
 for(i=0;i<n;i++)  
 {  
 pos=i;  
 for(j=i+1;j<n;j++)  
 {  
 if(pr[j]<pr[pos])  
 pos=j;  
 }  
  
 temp=pr[i];  
 pr[i]=pr[pos];  
 pr[pos]=temp;  
  
 temp=b[i];  
 b[i]=b[pos];  
 b[pos]=temp;  
  
 temp=a[i];  
 a[i]=a[pos];  
 a[pos]=temp;  
  
 temp=p[i];  
 p[i]=p[pos];  
 p[pos]=temp;  
 }  
  
 printf("\nProcess\t Burst Time \tArrival Time\t\tPriority ");  
 for(i=0;i<n;i++)  
 {  
 printf("\nP[%d]\t\t %d\t\t %d\t\t\t%d",p[i],b[i],a[i],pr[i]);  
 }  
 for(time=0;count!=n;time++)  
 {  
 smallest=9;  
 for(i=0;i<n;i++)  
 {  
 if(a[i]<=time && b[i]<b[smallest] && b[i]>0 )  
 smallest=i;  
 }  
 b[smallest]--;  
 if(b[smallest]==0)  
 {  
 count++;  
 end=time+1;  
 avg=avg+end-a[smallest]-x[smallest];  
 tt= tt+end-a[smallest];  
 }  
 }  
 printf("\n\nAverage waiting time = %lf\n",avg/n);  
 printf("Average Turnaround time = %lf",tt/n/n);  
 return 0;  
}

**Output :**

****

**ROUND ROBIN**

**Code :**

#include<stdio.h>   
   
int main()   
{   
 int i, limit, total = 0, x, counter = 0, time\_quantum;   
 int wait\_time = 0, turnaround\_time = 0, arrival\_time[10], burst\_time[10], temp[10];   
 float average\_wait\_time, average\_turnaround\_time;  
 printf("\nEnter Total Number of Processes:\t");   
 scanf("%d", &limit);   
 x = limit;   
 for(i = 0; i < limit; i++)   
 {  
 printf("\nEnter Details of Process[%d]\n", i + 1);  
 printf("Arrival Time:\t");  
 scanf("%d", &arrival\_time[i]);  
 printf("Burst Time:\t");  
 scanf("%d", &burst\_time[i]);   
 temp[i] = burst\_time[i];  
 }   
 printf("\nEnter Time Quantum:\t");   
 scanf("%d", &time\_quantum);   
 printf("\nProcess ID\t\tBurst Time\t Arrival Time\t Turnaround Time\t Waiting Time\n");  
 for(total = 0, i = 0; x != 0;)   
 {   
 if(temp[i] <= time\_quantum && temp[i] > 0)   
 {   
 total = total + temp[i];   
 temp[i] = 0;   
 counter = 1;   
 }   
 else if(temp[i] > 0)   
 {   
 temp[i] = temp[i] - time\_quantum;   
 total = total + time\_quantum;   
 }   
 if(temp[i] == 0 && counter == 1)   
 {   
 x--;   
 printf("\nProcess[%d]\t\t%d\t\t%d\t\t%d\t\t\t\t %d", i + 1, burst\_time[i],arrival\_time[i], total - arrival\_time[i], total - arrival\_time[i] - burst\_time[i]);  
 wait\_time = wait\_time + total - arrival\_time[i] - burst\_time[i];   
 turnaround\_time = turnaround\_time + total - arrival\_time[i];   
 counter = 0;   
 }   
 if(i == limit - 1)   
 {  
 i = 0;   
 }  
 else if(arrival\_time[i + 1] <= total)   
 {  
 i++;  
 }  
 else   
 {  
 i = 0;  
 }  
 }   
 average\_wait\_time = wait\_time \* 1.0 / limit;  
 average\_turnaround\_time = turnaround\_time \* 1.0 / limit;  
 printf("\n\nAverage Waiting Time:\t%f", average\_wait\_time);   
 printf("\nAvg Turnaround Time:\t%f\n", average\_turnaround\_time);   
 return 0;   
}

**Output :**

