**Ex.No: 5 CPU SCHEDULING**

**Date : 19.03.2021**

**Aim**

To do CPU Scheduling using the following:

* **First Come First Serve (FCFS) scheduling**
* **Shortest Job First (SJF) scheduling**
* **Priority scheduling**
* **Round Robin scheduling**

**1. Write a C program to perform CPU Scheduling using FCFS.**

**Algorithm:**

**1.** Start

**2.** Create a file using vi command with filename.c

**3**. Include the header file <stdio.h> for I/O operations.

**4**. Declare variables n,i,j,sum of integer data type and also declare 3 integer arrays to store the burst time,waiting time and turn around time of the processes.

**5**. Get the no. of processes (n) from the user.

**6**. Using for loop, which runs upto value of n, get the burst time for n processes from the user and store it in the array.

**7**. Assign waiting time of index 0th process as 0.

**8**. The using for loop ,calculate the waiting time of remaining processes by summing up the burst time of 0 to the current index process.

**9**. To calculate turn around time of each process , sum up the waiting time and burst time of the correseponding process.

**10**. Finally, calculate the average Waiting time and average turn around time and print all these as output.

**11.** Stop.

**Program:**

#include<stdio.h>

void main()

{

int n,i,j,sum=0,avg=0;

float awt,atat;

printf("Enter the no. of processes:");

scanf("%d",&n);

int p\_bt[n],p\_wt[n],p\_tt[n];

printf("\nEnter the Burst time for all the processes:");

for(i=0;i<n;i++)

{

printf("\nProcess %d:",(i+1));

scanf("%d",&p\_bt[i]);

}

p\_wt[0]=0;

for(i=1,j=0;i<n;i++,j++)

{

sum+=p\_bt[j];

p\_wt[i]=sum;

avg+=p\_wt[i];

}

awt=(float)avg/n;

avg=0;

for(i=0;i<n;i++)

{

p\_tt[i]=p\_bt[i]+p\_wt[i];

avg+=p\_tt[i];

}

atat=(float)avg/n;

printf("\nProcess Burst\_Time Waiting\_Time Turn\_Around\_Time");

for(i=0;i<n;i++)

{

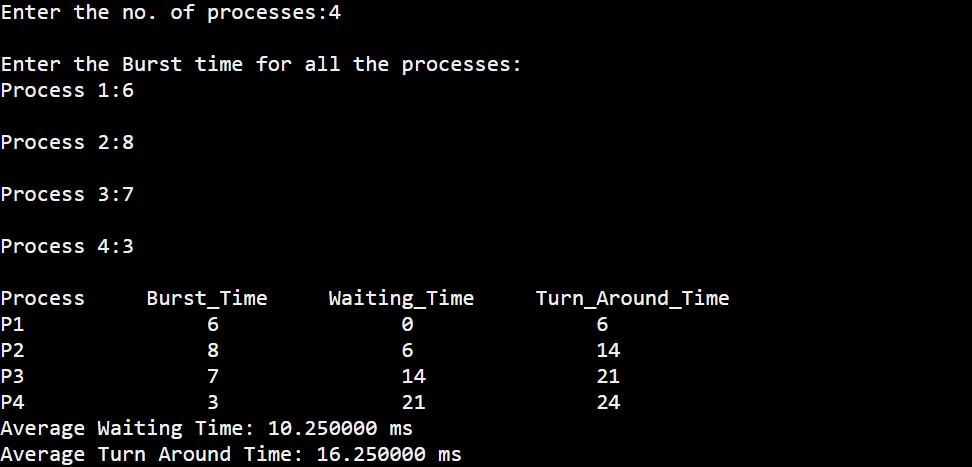
printf("\nP%d \t\t %d \t\t %d \t\t %d",(i+1),p\_bt[i],p\_wt[i],p\_tt[i]);

}

printf("\nAverage Waiting Time: %f ms\nAverage Turn Around Time: %f ms\n",awt,atat);

}

**Sample Output:**



**2. Write a C program to perform CPU Scheduling using SJF.**

**Algorithm:**

**1.** Start

**2.** Create a file using vi command with filename.c

**3**. Include the header file <stdio.h> for I/O operations.

**4**. Declare variables n,i,j,sum of integer data type and also declare 4 integer arrays to store the process id,burst time,waiting time and turn around time of the processes.

**5**. Get the no. of processes (n) from the user.

**6**. Using for loop, which runs upto value of n, get the burst time for n processes from the user and store it in the array.

**7**. The using selection sort, sort the burst time along with its process id in an ascending order.

**8**. Assign waiting time of first process as 0.

**9**. The using for loop ,calculate the waiting time of remaining processes by summing up the burst time of 0 to the current index process.

**10**. To calculate turn around time of each process , sum up the waiting time and burst time of the correseponding process.

**11**. Finally, calculate the average Waiting time and average turn around time and print all these as output.

**12.** Stop.

**Program:**

#include<stdio.h>

void main()

{ int i,j,temp,n,p[10],p\_bt[10],p\_wt[10],p\_tt[10],sum=0,avg=0;

float awt,atat;

printf("Enter the no. of processes:");

scanf("%d",&n);

printf("\nEnter the Burst time for all processes:");

for(i=0;i<n;i++)

{

p[i]=i+1;

printf("\nProcess %d: ",p[i]);

scanf("%d",&p\_bt[i]);

}

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(p\_bt[i]>p\_bt[j])

{

temp=p\_bt[i];

p\_bt[i]=p\_bt[j];

p\_bt[j]=temp;

temp=p[i];

p[i]=p[j];

p[j]=temp;

}

}

}

p\_wt[0]=0;

for(i=1,j=0;i<n;i++,j++)

{

sum+=p\_bt[j];

p\_wt[i]=sum;

avg+=p\_wt[i];

}

awt=(float)avg/n;

avg=0;

for(i=0;i<n;i++)

{

p\_tt[i]=p\_bt[i]+p\_wt[i];

avg+=p\_tt[i];

}

atat=(float)avg/n;

printf("\nProcess Burst\_Time Waiting\_Time Turn\_Around\_Time");

for(i=0;i<n;i++)

{

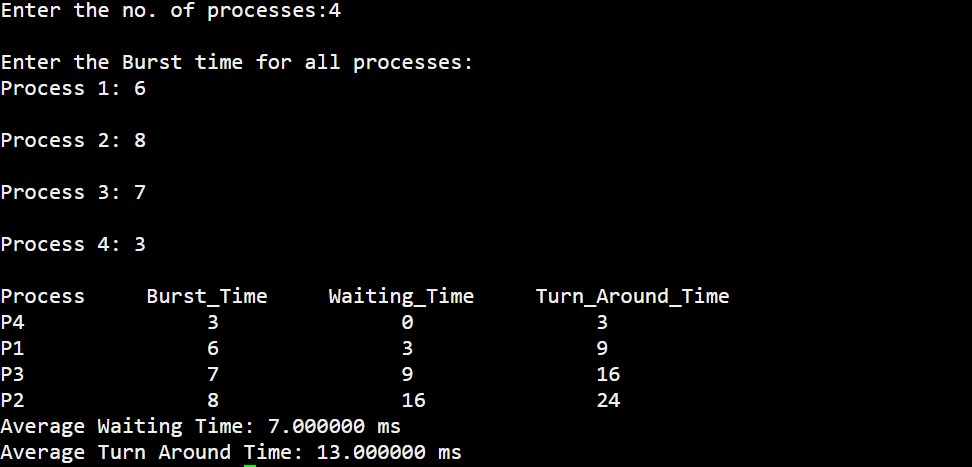
printf("\nP%d \t\t %d \t\t %d \t\t %d",p[i],p\_bt[i],p\_wt[i],p\_tt[i]);

}

printf("\nAverage Waiting Time: %f ms\nAverage Turn Around Time: %f ms\n",awt,atat);

}

**Sample Output:**

****

**3. Write a C program to perform CPU Scheduling using Priority scheduling.**

**Algorithm:**

**1.** Start

**2.** Create a file using vi command with filename.c

**3**. Include the header file <stdio.h> for I/O operations.

**4**. Declare variables n,i,j,sum,avg,temp of integer data type and also declare 5 integer arrays to store the process id,burst time,priority number,waiting time and turn around time of the processes.

**5**. Get the no. of processes (n) from the user.

**6**. Using for loop, which runs upto value of n, get the burst time and priority no. for n processes from the user and store it in the corresponding arrays.

**7**. The using selection sort, sort the priority no. along with its process id and burst time in an ascending order.

**8**. Assign waiting time of first process as 0.

**9**. The using for loop ,calculate the waiting time of remaining processes by summing up the burst time of 0 to the current index process.

**10**. To calculate turn around time of each process , sum up the waiting time and burst time of the correseponding process.

**11**. Finally, calculate the average Waiting time and average turn around time and print all these as output.

**12.** Stop.

**Program:**

#include<stdio.h>

void main()

{

int i,j,temp,n,p[10],p\_no[10],p\_bt[10],p\_wt[10],p\_tt[10],sum=0,avg=0;

float awt,atat;

printf("Enter the no. of processes:");

scanf("%d",&n);

printf("\nEnter the Burst time & Priority number for all processes:");

for(i=0;i<n;i++)

{

p[i]=i+1;

printf("\nProcess %d: ",p[i]);

scanf("%d %d",&p\_bt[i],&p\_no[i]);

}

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(p\_no[i]>p\_no[j])

{

temp=p\_no[i];

p\_no[i]=p\_no[j];

p\_no[j]=temp;

temp=p\_bt[i];

p\_bt[i]=p\_bt[j];

p\_bt[j]=temp;

temp=p[i];

p[i]=p[j];

p[j]=temp;

}

}

}

p\_wt[0]=0;

for(i=1,j=0;i<n;i++,j++)

{

sum+=p\_bt[j];

p\_wt[i]=sum;

avg+=p\_wt[i];

}

awt=(float)avg/n;

avg=0;

for(i=0;i<n;i++)

{

p\_tt[i]=p\_bt[i]+p\_wt[i];

avg+=p\_tt[i];

}

atat=(float)avg/n;

avg=0;

printf("\nProcess Burst\_Time Priority\_no. Waiting\_Time Turn\_Aro

und\_Time");

for(i=0;i<n;i++)

{

printf("\nP%d \t\t %d \t\t %d \t\t %d \t\t %d",p[i],p\_bt[i],p\_no[i],p\_wt[i],p\_

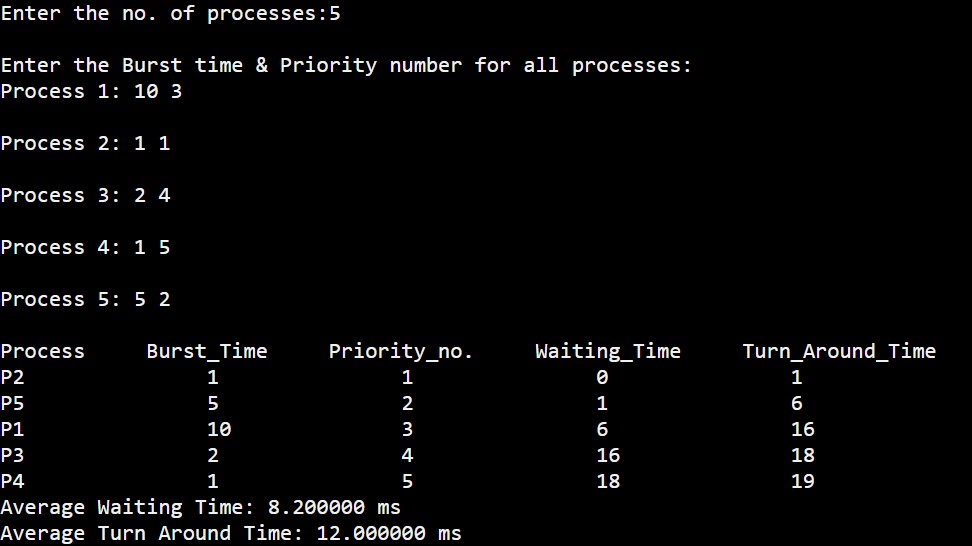
tt[i]);

}

printf("\nAverage Waiting Time: %f ms\nAverage Turn Around Time: %f ms\n",awt,atat);

}

**Sample Output:**



**4. Write a C program to perform CPU Scheduling using Round Robin scheduling.**

**Algorithm:**

**1.** Start

**2.** Create a file using vi command with filename.c

**3**. Include the header files <stdio.h> for I/O operations.

**4**. Declare variables n,i,time\_slice,s,awt,atat,count of integer data type and also declare 4 integer arrays to store the burst time ,waiting time, remaining burst time and turn around time of the processes.

**5**. Get the no. of processes (n) and time\_slice from the user.

**6**. Using for loop, which runs upto value of n, get the burst time for n processes from the user and store it in the array.

**7**. Then execute the processes each time upto the time\_slice and maintain the remaining burst time to be run in next slot.

**8**. Continue this process until all the processes complete their burst time.

**9**. Assign waiting time of first process as 0.

**10**. The using for loop ,calculate the waiting time of remaining processes by summing up the burst time of 0 to the current index process.

**11**. To calculate turn around time of each process , sum up the waiting time and burst time of the correseponding process.

**12**. Finally, calculate the average Waiting time and average turn around time and print all these as output.

**13.** Stop.

**Program:**

#include<stdio.h>

void main()

{

int n,i,time\_slice,count=0,temp,s=0,p\_bt[10],p\_wt[10],p\_tt[10],bal\_bt[10];

float awt,atat;

printf("Enter the no. of processes:");

scanf("%d",&n);

printf("Enter the Burst time for all the processes:");

for(i=0;i<n;i++)

{

printf("\nProcess %d:",(i+1));

scanf("%d",&p\_bt[i]);

bal\_bt[i]=p\_bt[i];

}

printf("Enter Time\_slice:");

scanf("%d",&time\_slice);

while(1)

{

for(i=0,count=0;i<n;i++)

{

temp=time\_slice;

if(bal\_bt[i]==0)

{

count++;

continue;

}

if(bal\_bt[i]>time\_slice)

bal\_bt[i]=bal\_bt[i]-time\_slice;

else

if(bal\_bt[i]>=0)

{

temp=bal\_bt[i];

bal\_bt[i]=0;

}

s+=temp;

p\_tt[i]=s;

}

if(n==count)

break;

}

printf("\nProcess Burst\_Time Waiting\_Time Turn\_Around\_Time");

for(i=0;i<n;i++)

{

p\_wt[i]=p\_tt[i]-p\_bt[i];

awt+=p\_wt[i];

atat+=p\_tt[i];

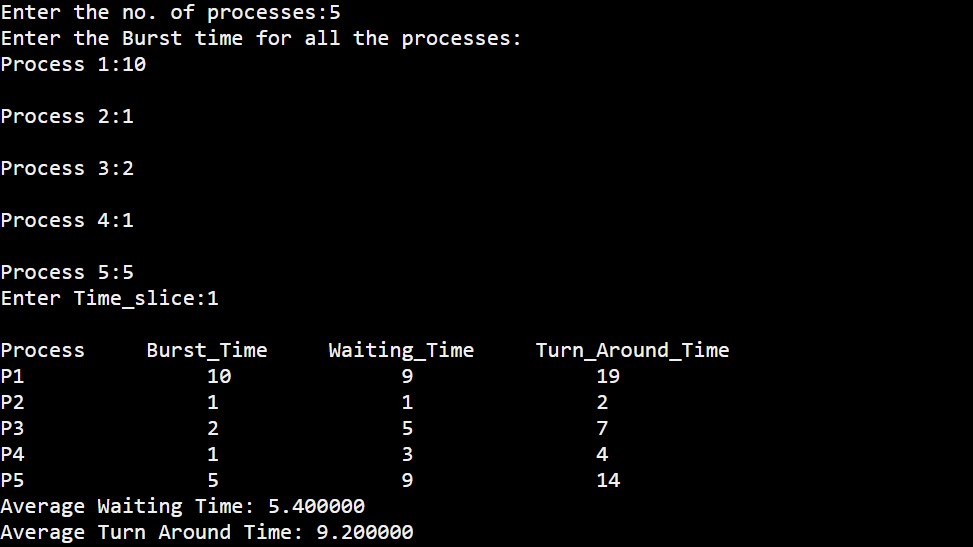
printf("\nP%d \t\t %d \t\t %d \t\t %d",(i+1),p\_bt[i],p\_wt[i],p\_tt[i]);

}

printf("\nAverage Waiting Time: %f\nAverage Turn Around Time: %f\n",awt/n,atat/n);

}

**Sample Output:**

****

|  |  |
| --- | --- |
| **Observation(20)** |  |
| **Record(5)** |  |
| **Total(25)** |  |
| **Initial** |  |

**Result:**

Thus the CPU scheduling is done successfully using different types of scheduling.