

Lab Report No. 09

Lab Report Name: Implementation of Priority Scheduling algorithm .

Objectives:

- i. What is Priority Scheduling algorithm.
- ii. How to implementation in C

Theory:

Priority scheduling is one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with the highest priority is to be executed first and so on. Processes with the same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

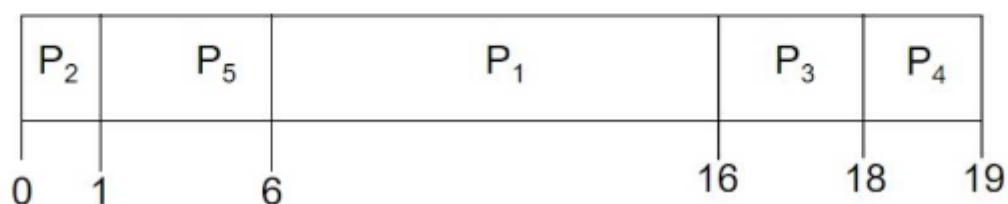
Implementation :

- First input the processes with their burst time and priority
- Sort the processes, burst time and priority according to the priority
- Now simply apply **FCFS** algorithm.

Example:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

Gantt Chart



C program to implement priority scheduling:

```
#include<stdio.h>

int main()
{
    int burst[20], p[20], waiting[20], tat[20], priority[20],i,j,n,total=0,process,temp;
    double avg_wait,avg_turn;
    printf("Enter Total Number of Process:");
    scanf("%d",&n);

    for(i=0; i<n; i++)
    { printf("\nEnter Burst Time and Priority for process %d : \n",i+1);

        scanf("%d",&burst[i]);

        scanf("%d",&priority[i]);
        p[i]=i+1;        //contains process number
    }

    for(i=0; i<n; i++)
    {
        process=i;
        for(j=i+1; j<n; j++)
        {
            if(priority[j]<priority[process])
                process=j;
        }

        temp=priority[i];
        priority[i]=priority[process];
```

```

priority[process]=temp;

temp=brust[i];
brust[i]=brust[process];
brust[process]=temp;

temp=p[i];
p[i]=p[process];
p[process]=temp;
}

waiting[0]=0; //waiting time for first process is zero

//calculate waiting time
for(i=1; i<n; i++)
{
    waiting[i]=0;
    for(j=0; j<i; j++)
        waiting[i]+=brust[j];

    total+=waiting[i];
}

avg_wait=(float)total/(float)n; //average waiting time
total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
for(i=0; i<n; i++)
{
    tat[i]=brust[i]+waiting[i]; //calculate turnaround time
    total+=tat[i];
}

```

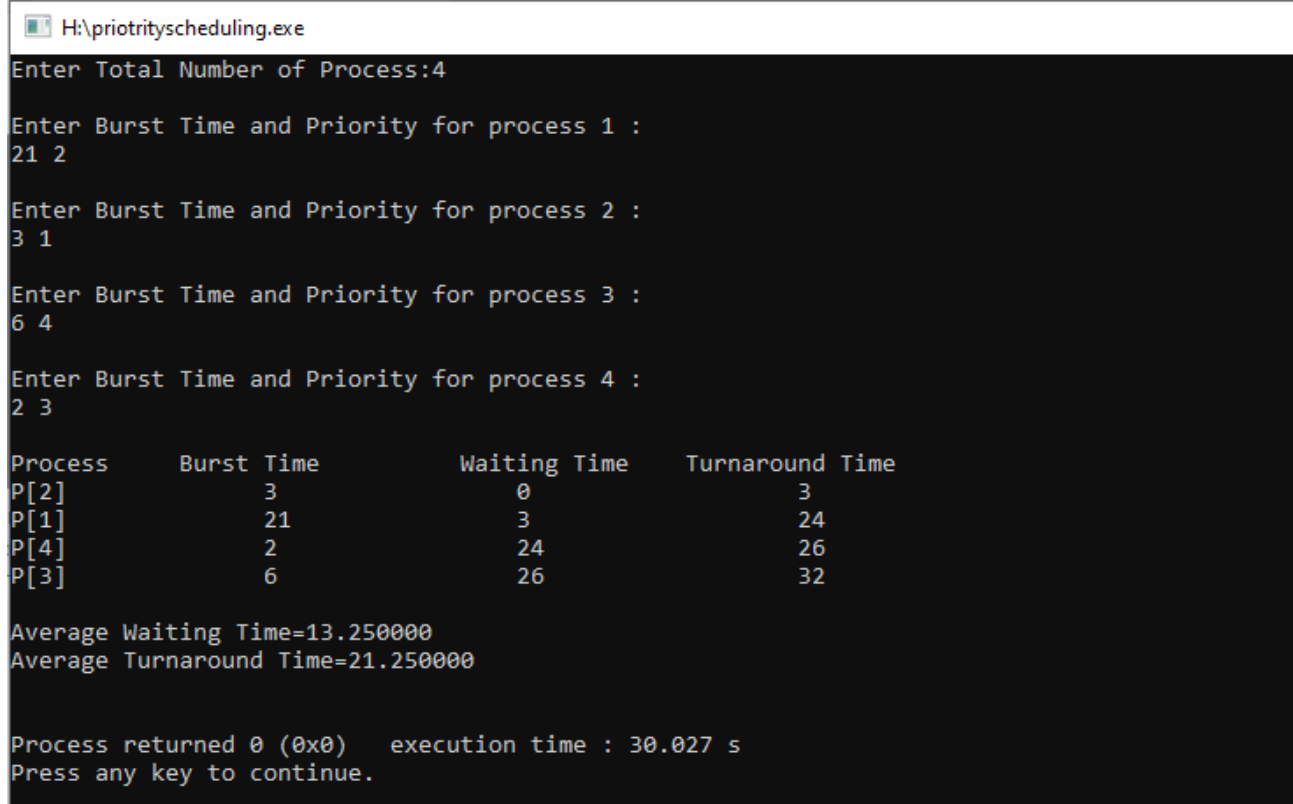
```

        printf("\nP[%d]\t\t %d\t\t %d\t\t\t%d",p[i],brust[i],waiting[i],tat[i]);
    }

    double avg_tat= (double)total/(double)n;    //average turnaround time
    printf("\n\nAverage Waiting Time=%f",avg_wait);
    printf("\nAverage Turnaround Time=%f\n",avg_tat);
    printf("\n");
    return 0;
}

```

outputs:



```

H:\priorityscheduling.exe
Enter Total Number of Process:4

Enter Burst Time and Priority for process 1 :
21 2

Enter Burst Time and Priority for process 2 :
3 1

Enter Burst Time and Priority for process 3 :
6 4

Enter Burst Time and Priority for process 4 :
2 3

Process      Burst Time      Waiting Time      Turnaround Time
P[2]          3                0                 3
P[1]         21                3                24
P[4]          2                24               26
P[3]          6                26               32

Average Waiting Time=13.250000
Average Turnaround Time=21.250000

Process returned 0 (0x0)   execution time : 30.027 s
Press any key to continue.

```