Lab Report No. 08

Lab Report Name: Implementation of SJF Scheduling algorithm .

Objectives:

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

Theory:

In SJF scheduling, the process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next.

However, it is very difficult to predict the burst time needed for a process hence this algorithm is very difficult to implement in the system.

It is of two types:

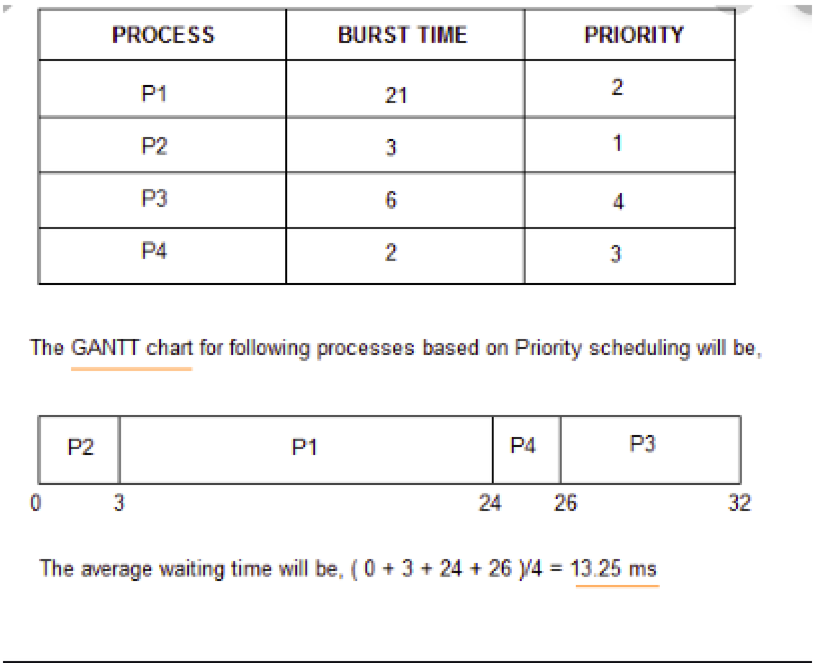
1.Non Pre-emptive

2.Pre-emptive

Algorithm:

- Sort all the process according to the arrival time.
- Then select that process which has minimum arrival time and minimum Burst time.
- After completion of process make a pool of process which after till the completion of previous process and select that process among the pool which is having minimum Burst time.

Example:



PROCESS	BURST TIME	PRIORITY
P1	21	2
P2	3	1
P3	6	4
P4	2	3

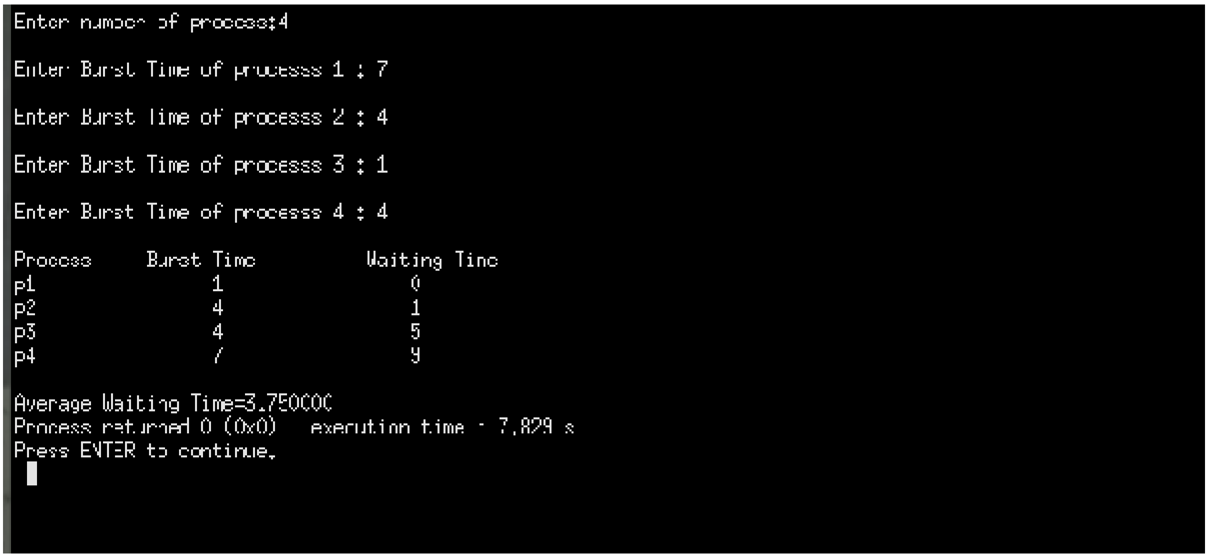
The GANTT chart for following processes based on Priority scheduling will be,

The average waiting time will be, (0 + 3 + 24 + 26)/4 = 13.25 ms

C program for implementing for implementing shorted job first algorithms:

```
#include<bits/stdc++.h>
using namespace std;
main()
{
    int n;
    float avg_wt,avg_tat;
    printf("Enter number of process:");
    scanf("%d",&n);
     int bt[n],p[n],wt[n],i,j,total=0,pos,temp;

    for(i=0; i<n; i++)
    {
         printf("\nEnter Burst Time of processs %d : ",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;
    }
    sort(bt,bt+n);

    wt[0]=0;
    for(i=1; i<n; i++)
    {
        wt[i]=0;
        for(j=0; j<i; j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }

    avg_wt=(float)total/(float)n;
    printf("\nProcess\t    Burst Time    \tWaiting Time");
    for(i=0; i<n; i++)
    {
        printf("\np%d\t\t  %d\t\t    %d",p[i],bt[i],wt[i]);
    }
    printf("\n\nAverage Waiting Time=%f",avg_wt);
    return 0;
}
```

outputs:



```
Enter number of process:4

Enter Burst Time of processs 1 : 7

Enter Burst Time of processs 2 : 4

Enter Burst Time of processs 3 : 1

Enter Burst Time of processs 4 : 4

Process     Burst Time          Waiting Time
p1                1                 0
p2                4                 1
p3                4                 5
p4                7                 9

Average Waiting Time=3.750000
Process returned 0 (0x0)   execution time : 7.829 s
Press ENTER to continue.
```