



**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

**Santosh, Tangail-1902**

# LAB REPORT

Lab Report No : 08  
Lab Report name : Implementation of SJF Scheduling algorithm .  
Course Title : Operating System Lab  
Course Code : ICT-3110  
Date of Performance : 20-08-2020  
Date of Submission : 28-08-2020

Submitted by,  
Student Name : Tanvir Ahmed  
Student ID : IT-18043  
Session : 2017-18  
3<sup>rd</sup> Year 1<sup>st</sup> semester  
Dept. of ICT

Submitted to,  
Nazrul Islam  
Assistant Professor  
Dept. of ICT,  
MBSTU.

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 :** Shortest-Job-First (SJF) is a **non-preemptive** discipline in which waiting job (or process) with the smallest estimated run-time-to-completion is run next. In other words, when CPU is available, it is assigned to the process that has smallest next CPU burst. The SJF scheduling is especially appropriate for batch jobs for which the run times are known in advance. Since the SJF scheduling algorithm gives the minimum average time for a given set of processes, it is probably optimal. The SJF algorithm favors short jobs (or processors) at the expense of longer ones.

The obvious problem with SJF scheme is that it requires precise knowledge of how long a job or process will run, and this information is not usually available. The best SJF algorithm can do is to rely on user estimates of run times.

### Characteristics of SJF scheduling :

1. It is associated with each job as a unit of time to complete.
2. This algorithm method is helpful for batch-type processing, where waiting for jobs to complete is not critical.
3. It can improve process throughput by making sure that shorter jobs are executed first, hence possibly have a short turnaround time.
4. It improves job output by offering shorter jobs, which should be executed first, which mostly have a shorter turnaround time.

### Advantages of SJF

1. Maximum throughput
2. Minimum average waiting and turnaround time

### Disadvantages of SJF

1. May suffer with the problem of starvation
2. It is not implementable because the exact Burst time for a process can't be known in advance.

## Corresponding Code:

```
#include<stdio.h>

int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;

    float avg_wt,avg_tat;

    printf("Enter number of process:");

    scanf("%d",&n);

    printf("\nEnter Burst Time:\n");

    for(i=0;i<n;i++)
    {
        printf("p%d: ",i+1);

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

        p[i]=i+1;
    }

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

        for(j=i+1;j<n;j++)
        {
            if(bt[j]<bt[pos])

                pos=j;
        }

        temp=bt[i];

        bt[i]=bt[pos];
```

```

    bt[pos]=temp;

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

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/n;
total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
{
    tat[i]=bt[i]+wt[i];
    total+=tat[i];

    printf("\n%d\t\t %d\t\t\t %d\t\t\t %d",p[i],bt[i],wt[i],tat[i]);
}

```

```
    avg_tat=(float)total/n;    //average turnaround time

    printf("\n\nAverage Waiting Time=%f",avg_wt);

    printf("\nAverage Turnaround Time=%f\n",avg_tat);

    printf("\n");

    return 0;

}
```

## Output:

```
tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx: ~/CodePractice/C_programming
File Edit View Search Terminal Help
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~/CodePractice/C_programming$ ./SJF
Enter number of process:5

Enter Burst Time:
p1: 5
p2: 2
p3: 4
p4: 9
p5: 8

Process    Burst Time    Waiting Time    Turnaround Time
p2          2             0               2
p3          4             2               6
p1          5             6              11
p5          8            11              19
p4          9            19              28

Average Waiting Time=7.600000
Average Turnaround Time=13.200000

(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~/CodePractice/C_programming$
```

**Conclusion:** The SJF means Shortest Job first algorithm. Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms. SJF scheduling algorithm, schedules the processes according to their burst time. 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. It is a Greedy Algorithm.

Here we got 5 process of burst time 5,2,4,9,8 . By SJF they are sorted according to their burst time like 2,4,5,8,9. Here p2 executed first. Then p3,p1,p5,p4. Here we see that the process of shortest burst time 2 has the priority to execute first. After that rest of them are also organized by SJF algorithm.

SJF may cause starvation if shorter processes keep coming. It is not implementable because the exact Burst time for a process can't be known in advance.