



Mawlana Bhashani Science and Technology University

Lab-Report

Lab Report No: 08

Lab Report Name: Implementation of SJF Scheduling algorithm .

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 16-09-2020

Date of Submission:

Submitted by

Name: Md Masud Rana

ID: IT-18002

3rd year 1st semester

Session: 2017-18

Dept. of ICT.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

Experiment No: 08

Experiment Name: Implementation of SJF Scheduling algorithm .

Theory: Shortest job first scheduling is the job or process scheduling algorithm that follows the non-preemptive scheduling discipline. In this, scheduler selects the process from the waiting queue with the least completion time and allocate the CPU to that job or process. Shortest Job First is more desirable than FIFO algorithm because SJF is more optimal as it reduces average wait time which will increase the throughput.

- **Completion Time** is the time required by the process to complete its execution
- **Turnaround Time** is the time interval between the submission of a process and its completion.

Turnaround Time = completion of a process – submission of a process

- **Waiting Time** is the difference between turnaround time and burst time

Waiting Time = turnaround time – burst time.

Code-

fcfs.c

```
#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("\np%d\t\t %d\t\t %d\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-

```

masud@masud-VirtualBox:~/program$ ./sjf.out
Enter number of process:4

Enter Burst Time:
p1: 10
p2: 20
p3: 30
p4: 40

Process   Burst Time   Waiting Time   Turnaround Time
p1         10           0              10
p2         20          10              30
p3         30          30              60
p4         40          60             100

Average Waiting Time=25.000000
Average Turnaround Time=50.000000

masud@masud-VirtualBox:~/program$

```

Discussion: From this lab we learn SJF scheduling algorithm. For implementation we have used C language. The major advantage of this algorithm is that it gives the minimum waiting time for a given set of processes and thus reduces the average waiting time. This lab helps us to realize about the SJF scheduling algorithm.