MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Santosh,Tangail-1902

LAB REPORT

Department of : Information & Communication Technology

Lab Report No : 07

Lab Report On : **Implementation of SJF scheduling algorithm**

Course Title : Operating Systems Lab

Course Code : ICT - 3110

Submitted by,	Submitted to,
Name : Zahid Hasan Chowdhury	Nazrul Islam
Student ID : IT-18017	Assistant Professor
Session : 2017-18	Dept. of ICT, MBSTU
Year: 3rd Semester : 1st	
Dept. of ICT MBSTU	

<u>Experiment No</u> : 08

<u>Experiment Name</u> : Implementation of SJF Scheduling Algorithm.

<u>Theory</u> :

Shortest job first(SJF) is a scheduling algorithm, that is used to schedule processes in an operating system. It is a very important topic in Scheduling when compared to round-robin and FCFS Scheduling.

There are two types of SJF

Pre-emptive SJF

Non-Preemptive SJF

These algorithms schedule processes in the order in which the shortest job is

done first. It has a minimum average waiting time.

There are 3 factors to consider while solving SJF, they are

1. BURST Time
2. Average waiting time
3. Average turn around time

Non-Preemptive Shortest Job First

Working Process :

Code for non-preemptive SJF scheduling Algorithm -

```
#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 processes :");
    scanf("%d",&n);

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

   //sorting of burst times
    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\t%d",p[i],bt[i],wt[i],tat[i]);
    }

    avg_tat=(float)total/n;
    printf("\n\nAverage Waiting Time=%f",avg_wt);
    printf("\nAverage Turnaround Time=%f\n",avg_tat);
}
```

Output :

```
C:\Users\ASUS\Desktop\algo2.exe
Enter number of processes :5

Enter Burst Time: p1:3
p2:2
p3:8
p4:5
p5:4

Process         Burst Time              Waiting Time    Turnaround Time
p2               2                      0                       2
p1               3                      2                       5
p5               4                      5                       9
p4               5                      9                       14
p3               8                      14                      22

Average Waiting Time=6.000000
Average Turnaround Time=10.400000

Process returned 0 (0x0)   execution time : 22.858 s
Press any key to continue.
```

Code for preemptive SJF Scheduling Algorithm :

```
#include <stdio.h>

int main()
{
    int arrival_time[10], burst_time[10], temp[10];
    int i, smallest, count = 0, time, limit;
    double wait_time = 0, turnaround_time = 0, end;
    float average_waiting_time, average_turnaround_time;
    printf("\nEnter the Total Number of Processes:\t");
    scanf("%d", &limit);
    printf("\nEnter Details of %d Processes\n", limit);
    for(i = 0; i < limit; i++)
    {
        printf("\nEnter Arrival Time:\t");
        scanf("%d", &arrival_time[i]);
        printf("Enter Burst Time:\t");
        scanf("%d", &burst_time[i]);
        temp[i] = burst_time[i];
    }
    burst_time[9] = 9999;
    for(time = 0; count != limit; time++)
    {
        smallest = 9;
        for(i = 0; i < limit; i++)
        {
            if(arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i]
> 0)
            {
                smallest = i;
            }
        }
        burst_time[smallest]--;
        if(burst_time[smallest] == 0)
        {
            count++;
            end = time + 1;
            wait_time = wait_time + end - arrival_time[smallest] - temp[smallest];
            turnaround_time = turnaround_time + end - arrival_time[smallest];
        }
```

```
    }
    average_waiting_time = wait_time / limit;
    average_turnaround_time = turnaround_time / limit;
    printf("\n\nAverage Waiting Time:\t%lf\n", average_waiting_time);
    printf("Average Turnaround Time:\t%lf\n",
    average_turnaround_time); return 0;
}
```

Output :

```
C:\Users\ASUS\Desktop\algo.exe

Enter the Total Number of Processes:    4

Enter Details of 4 Processes

Enter Arrival Time:     1
Enter Burst Time:       3

Enter Arrival Time:     2
Enter Burst Time:       3

Enter Arrival Time:     3
Enter Burst Time:       3

Enter Arrival Time:     4
Enter Burst Time:       3


Average Waiting Time:   3.000000
Average Turnaround Time:        6.000000

Process returned 0 (0x0)   execution time : 54.392 s
Press any key to continue.
```

Discussion :

This lab helps to learn non-preemptive and preemptive SJF (Smallest Job First) scheduling algorithm. We have implemented this algorithm using C language. It returns the result accurately.