

Lab-Report

Report No: 08

Course code: ICT- ICT-3110

Course title: Operating Systems Lab

Date of Performance:11-09-20

Date of Submission:27-09-20

Submitted by

Name: Nusrat Jahan Jui

ID:IT-18039

3rd year 1stsemester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment no : 08

Experiment Name : Implementation of SJF Scheduling Algorithm.

Theory :

Shortest job first (SJF) scheduling algorithm, schedules the processes according to their burst time. Shortest job first can be either preemptive or non-preemptive. Owing to its simple nature, shortest job first is considered optimal. It also reduces the average waiting time for other processes awaiting execution.

Implementation :

- Step-1: Sort all the process according to the arrival time.
- Step-2: Then select that process which has minimum arrival time and minimum Burst time.
- Step-3: 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.

Working Process :

Code for SJF Scheduling Algorithm–

```
#include<stdio.h>
void 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;
printf("\n\nAverage Waiting Time=%f",avg_wt);
printf("\n\nAverage Turnaround Time=%f\n",avg_tat);
}

```

Output :

```

C:\Users\Admin\Documents\jui_lab_08.exe
Enter number of process:4
Enter Burst Time:
p1:3
p2:7
p3:2
p4:8

Process      Burst Time      Waiting Time      Turnaround Time
p3           2              0                2
p1           3              2                5
p2           7              5               12
p4           8             12               20

Average Waiting Time=4.750000
Average Turnaround Time=9.750000

Process returned 34 (0x22)   execution time : 28.316 s
Press any key to continue.

```

Discussion :

The above algorithm has been implemented using C language. In the above program, we calculate the average waiting and average turn around times of the jobs. The average waiting time is calculated:

$\text{avg_wt} = (\text{float}) \text{total} / n;$

Then, the turnaround time is calculated by adding the burst time and the waiting time.