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:
Date of Submission: 29/09/2020

Submitted by

Name: Ali Ashadullah Arif
ID:IT-18031
3rd Year 1st Semester
Session: 2017-2018
Dept. of ICT
MBSTU.

Submitted To

Nazrul Islam
Assistant Professor
Dept. of ICT
MBSTU.

Lab Report No: 08

Name of the Lab Report: Implementation of SJF Scheduling Algorithm

Objective: SJF algorithm Definition & executable code in C.

1. What is SJF Scheduling Algorithm?

Answer: Shortest job first is a scheduling algorithm in which the process with the smallest execution time is selected for execution next. 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.

2. How to implemented in C?

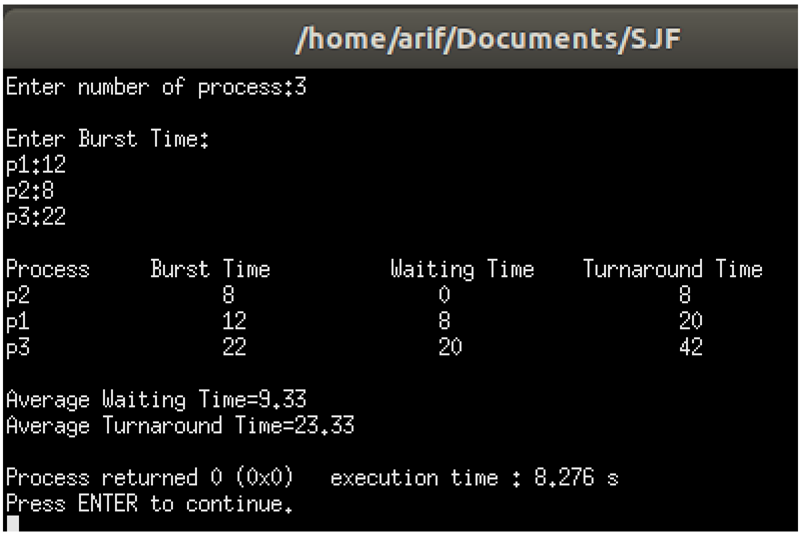
Answer:

Source Code of SJF Algorithm:

```
#include<stdio.h>
int main()
{
                int BuT[31],Store[31],WaT[31],TaT[31],i,j,n,total=0,pos,temp;
                float Avgwt,AvgTaT;
                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",&BuT[i]);
                    Store[i]=i+1;
                }
                for(i=0; i<n; i++) {
                    pos=i;
                    for(j=i+1; j<n; j++) {
                        if(BuT[j]<BuT[pos])
                            pos=j;
                    }
```

```
        temp=BuT[i];
        BuT[i]=BuT[pos];
        BuT[pos]=temp;

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

    WaT[0]=0;
    for(i=1; i<n; i++) {
        WaT[i]=0;
        for(j=0; j<i; j++)
            WaT[i]+=BuT[j];

        total+=WaT[i];
    }

    Avgwt=(float)total/n;
    total=0;
    printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");
    for(i=0; i<n; i++) {
        TaT[i]=BuT[i]+WaT[i];
        total+=TaT[i];
        printf("\np%d\t\t  %d\t\t    %d\t\t\t%d",Store[i],BuT[i],WaT[i],TaT[i]);
    }
    AvgTaT=(float)total/n;
    printf("\n\nAverage Waiting Time=%.2f",Avgwt);
    printf("\nAverage Turnaround Time=%.2f\n",AvgTaT);
    return 0;
}
```

Output:



```
Enter number of process:3

Enter Burst Time:
p1:12
p2:8
p3:22

Process     Burst Time          Waiting Time    Turnaround Time
p2              8                   0                   8
p1              12                  8                   20
p3              22                  20                  42

Average Waiting Time=9.33
Average Turnaround Time=23.33

Process returned 0 (0x0)   execution time : 8.276 s
Press ENTER to continue.
```

Conclusion: In this lab, we learnt about SJF algorithm. We are implementing this algorithm with C programming language. The main advantage of this algorithm is if we give the processes and burst time value, then it returns the average waiting time and average turnaround time.