Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 08

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:

Date of Submission: 09/09/2020

Submitted by

Name: Shakhera khanom

ID: IT-18033

3rd year 1st semester

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.
Objectives:

- ➢ What is SJF Scheduling Algorithm?
- ➢ How to implementation in C?

❖ **SJF Scheduling Algorithm :**

SJF is a scheduling policy that selects the waiting process with the smallest execution time to execute next

Aim : To write a c program to Implement the CPU scheduling algorithm Shortest job first (Non- Preemption)

Description:
To calculate the average waiting time in the shortest job first algorithm the sorting of the process based on their burst time in ascending order then calculate the waiting time of each process as the sum of the bursting times of all the process previous or before to that process

Algorithm:

Step 1: Start the process
Step 2: Accept the number of processes in the ready Queue
Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time
Step 4: Start the Ready Q according the shortest Burst time by sorting according to lowest to highest burst time.
Step 5: Set the waiting time of the first process as_0‘ and its turnaround time as its burst time.
Step 6: Sort the processes names based on their Burt time

Step 7: For each process in the ready queue, calculate

a) Waiting time(n)= waiting time (n-1) + Burst time (n-1)

b) Turnaround time (n)= waiting time(n)+Burst time(n)

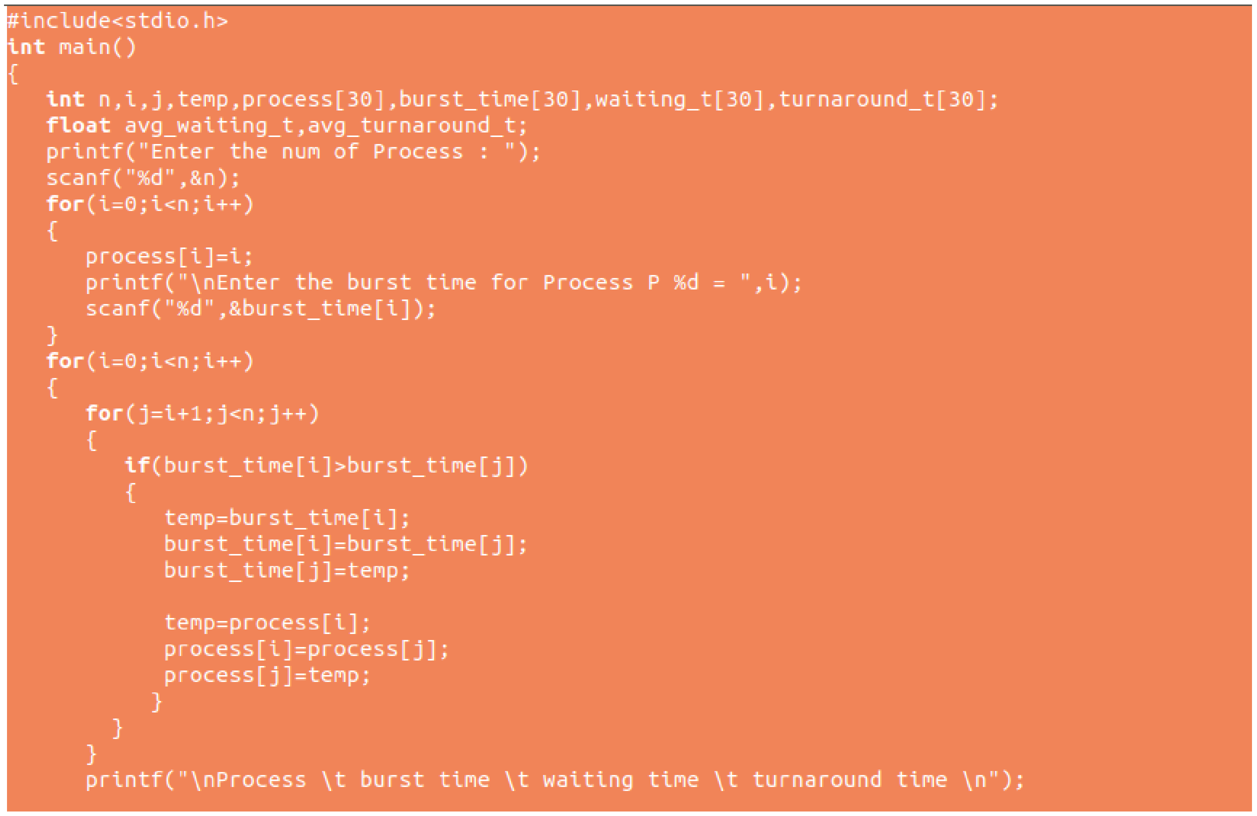
Step 8: Calculate

a) Average waiting time = Total waiting Time / Number of process

b) Average Turnaround time = Total Turnaround Time / Number of process

Step 9: Stop the process

Corresponding Code:



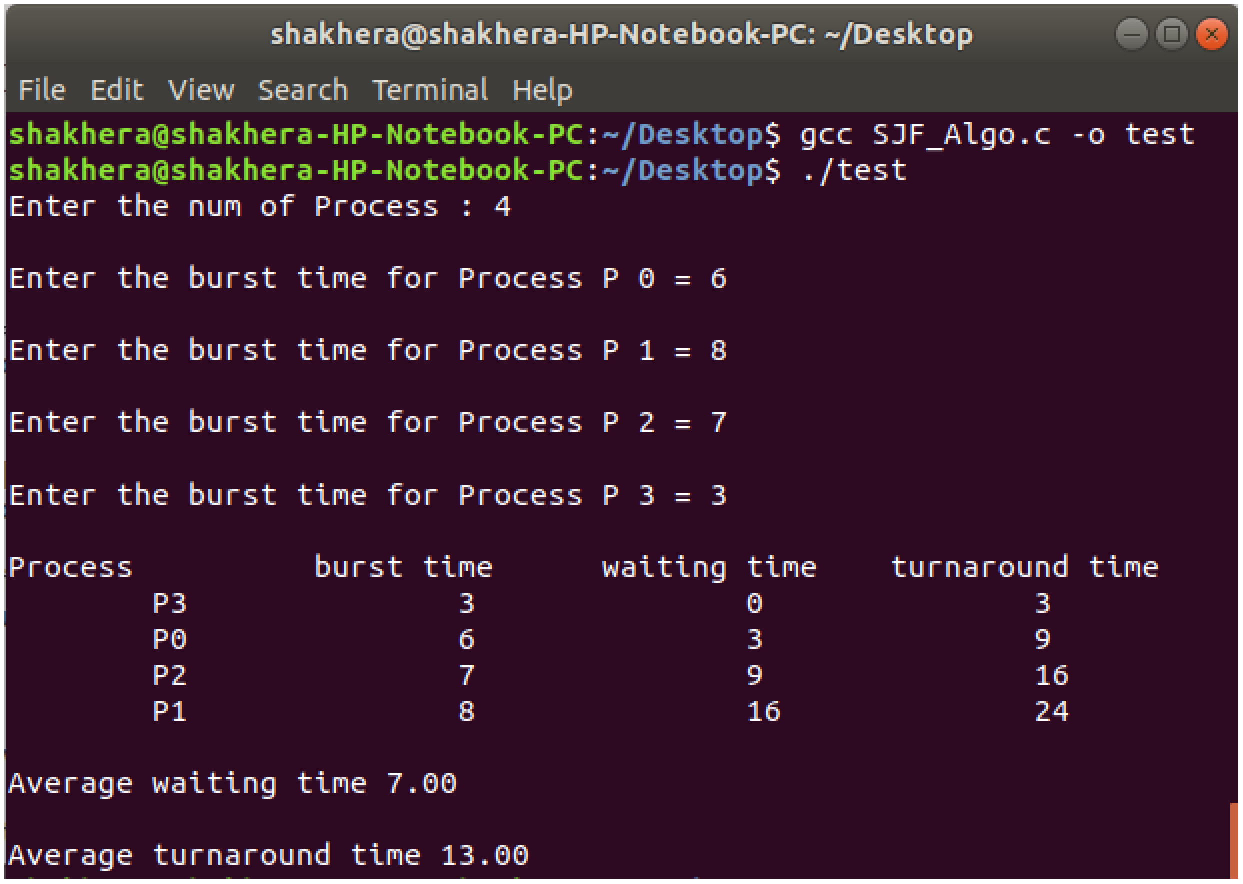
```
#include<stdio.h>
int main()
{
   int n,i,j,temp,process[30],burst_time[30],waiting_t[30],turnaround_t[30];
   float avg_waiting_t,avg_turnaround_t;
   printf("Enter the num of Process : ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      process[i]=i;
      printf("\nEnter the burst time for Process P %d = ",i);
      scanf("%d",&burst_time[i]);
   }
   for(i=0;i<n;i++)
   {
      for(j=i+1;j<n;j++)
      {
         if(burst_time[i]>burst_time[j])
         {
            temp=burst_time[i];
            burst_time[i]=burst_time[j];
            burst_time[j]=temp;

            temp=process[i];
            process[i]=process[j];
            process[j]=temp;
           }
        }
      }
      printf("\nProcess \t burst time \t waiting time \t turnaround time \n");
```

```c
    waiting_t[0]=avg_waiting_t=0;
    turnaround_t[0]=avg_turnaround_t=burst_time[0];
    for(i=1;i<n;i++)
    {
        waiting_t[i]=waiting_t[i-1]+burst_time[i-1];
        turnaround_t[i]=turnaround_t[i-1]+burst_time[i];

        avg_waiting_t=avg_waiting_t+waiting_t[i];
        avg_turnaround_t=avg_turnaround_t+turnaround_t[i];
    }
    for(i=0;i<n;i++)
    {
      printf("\tP%d\t\t%d\t\t%d\t\t%d\n",process[i],burst_time[i],waiting_t[i],turnaround_t[i]);
    }
    avg_waiting_t=avg_waiting_t/n;
    avg_turnaround_t=avg_turnaround_t/n;
    printf("\nAverage waiting time %.2f\n",avg_waiting_t);
    printf("\nAverage turnaround time %.2f\n",avg_turnaround_t);
    return 0;
}
```

Output:



```
shakhera@shakhera-HP-Notebook-PC:~/Desktop$ gcc SJF_Algo.c -o test
shakhera@shakhera-HP-Notebook-PC:~/Desktop$ ./test
Enter the num of Process : 4

Enter the burst time for Process P 0 = 6

Enter the burst time for Process P 1 = 8

Enter the burst time for Process P 2 = 7

Enter the burst time for Process P 3 = 3

Process         burst time      waiting time    turnaround time
        P3              3               0               3
        P0              6               3               9
        P2              7               9               16
        P1              8               16              24

Average waiting time 7.00

Average turnaround time 13.00
```

Discussion:

This lab helps to learn Shortest job first. Using this algorithm we can find waiting time and turnaround time . it is an algorithm in which the process having the smallest execution time is chosen for the next execution.