Mawlana Bhashani Scienceand Technology University

Lab-Report

Report No: 08

Course code:ICT- 3110

Course title:Operating Systems Lab

Date of Performance:

Date of Submission:

Submitted by

Name:Sayma Akter Shumi

ID:IT-18032

3rd year 1stsemester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Reference: Nusrat Jahan Jui(IT-18039)

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 :

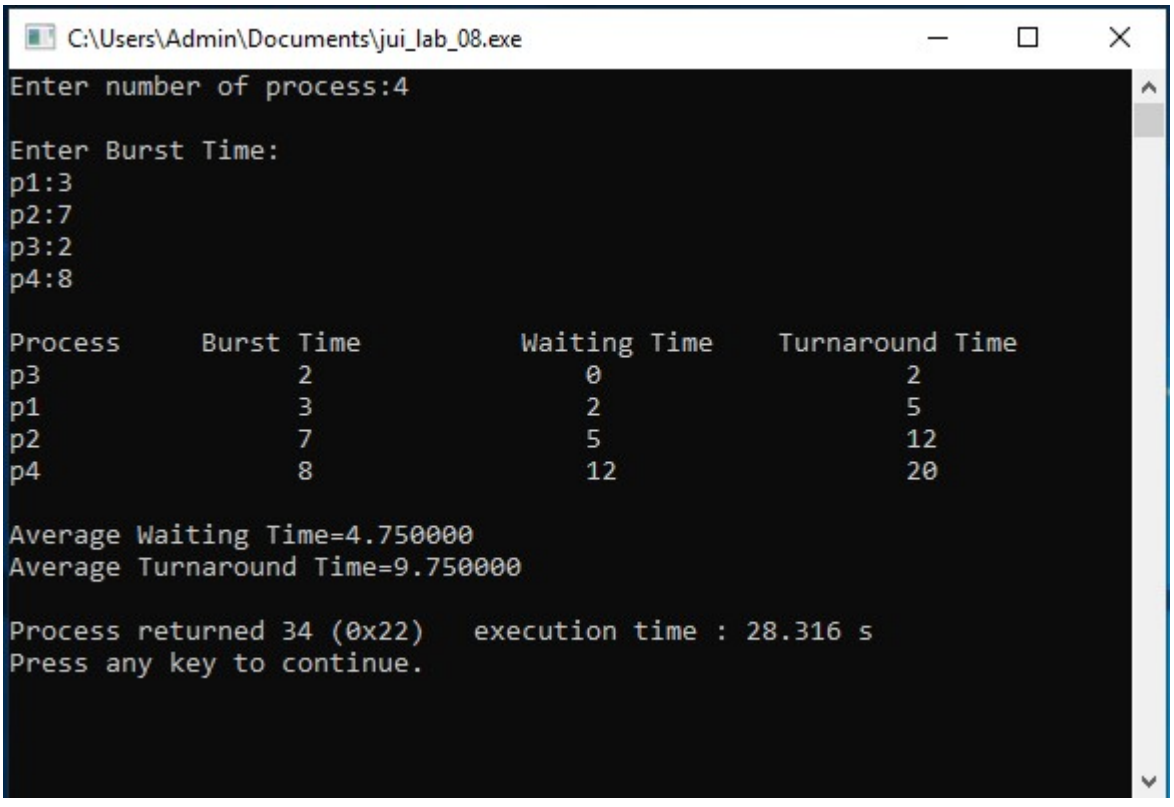
Code for SJF Scheduling Algorithm–

```
#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\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 :



Discussion :

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

avg_wt=(float)total/n;

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