

Department of Computer Science and Engineering

**Course Code :** CSE -334

**Course Title :** Operating System Lab.

**Report :** 02.

**Report Name :** Implementation of SJF Algorithm.

**Performance Date :** 08-08-2015.

**Submission Date :** 17-08-2015.

**REMARKS**

**NAME : Md. Jahidul alam**

**ID :** **12-023-1-02-00419**

**DEPT :** **CSE**

**YEAR :** **2015**

**SEM. :** **6th**

**Introduction:** Other name of this algorithm is Shortest-Process-Next (SPN). SJF is a non-preemptive discipline in which waiting job (or process) with the smallest estimated run-time-to-completion is run next.

**Objective:** Implementation of SJF algorithm.

**Source Code:**

#include<stdio.h>

main()

{

int temp,temp1,i,j,x,bt[20],pt[20];

scanf("%d",&x);

for(i=0;i<x;i++)

{

scanf("%d %d",&pt[i],&bt[i]);

}

for(i=0;i<x;i++)

for(j=0;j<x;j++)

{

if(bt[i]>bt[i+1])

{

temp=bt[i];

bt[i]=bt[i+1];

bt[i+1]=temp;

temp1=pt[i];

pt[i]=pt[i+1];

pt[i+1]=temp1;

}

}

for(i=0;i<x;i++)

{

for(j=0;j<bt[i];j++)

{

printf("P%d",pt[i]);

}

}

}

**Input:** 3

1 5

2 3

3 2

**Output:**

P2P2P2P3P3P1P1P1P1P1

Process returned 3 (0x3) execution time : 16.590 s

Press any key to continue.

**Discussion:**

1. There are no problems of compilation.
2. The problem was successfully executed.