

## Contents

USCSP301:USCS303-OperatingSystem(OS) Practical-02

Practical-02:Shortest Job First Scheduling Algorithm

Practical Date: 24 <sup>th</sup> July,2021.....	1
Practical Aim:Implement SJF (with no preemption) scheduling algorithm in Java.....	2
Algorithm .....	3
ExampleofSJF .....	4
GnattChart .....	5
Implementation.....	6
Input .....	7
Output .....	8
SampleOutput .....	9

## Algorithm:

CPU scheduling algorithm are used for scheduling different process present in the ready queue with available resource in an optimal way so that each and every process get execute by CPU

Scheduling algorithm are broadly classified into two main types namely preemptive and non-preemptive.

FIRST COME FIRST OUT (FCFS) is also known as FIRST IN FIRST OUT (FIFO) SCHEDUAL algorithm. It is the simplest CPU .

A process scheduling different processes to be assigned to the CPU based on particular scheduling algorithm. There are six popular process scheduling algorithm which we are going to discuss in this chapter FIRST COME FIRST OUT (FCFS) scheduling.

**Example1:** Consider the following example contain five processes.

ProcessId	BurstTime
P0	6
P1	3
P2	8
P3	3
P4	4

**Step1:** Processes get execute according to their lowest burst time first.

ProcessId	BurstTime
P0	6
P1	3
P2	8
P3	3
P4	4

**Step 2: Following shows the scheduling and execution of**

**processes**  
**Step 2.1:** At start P1 shortest execution time which is 0-3 second.

Systemtime	0
Processesschedulingfinishtime	P1
Finishtime	$0+3=3$
Watingtime	$3-3=0$
TurnAroundtime	$3-0=3$

**Step 2.2:** next shortest execution time is for process P3 for duration 3-6 second.

Systemtime	6
Processesschedulingfinishtime	P1,p3
Finishtime	$3+3=6$
Watingtime	$6-3=3$
TurnAroundtime	$6-0=6$

**Step2.3:**NextjobwithshortestexecutiontimeisP4foraduration6-10second.

Systemtime	10
Processesschedulingfinishtime	P1,p3'p4
Finishtime	6+4=10
Watingtime	10-4=6
TurnAroundtime	10-0=10

**Step2.4:**Nextjobwith shortestexecutiontimeisp0fordurationof10-16second.

Systemtime	10
Processesschedulingfinishtime	P1,p3,p3,p4,p0
Finishtime	10+6=16
Watingtime	16-6=10
TurnAroundtime	16-0=16

**Step2.5:**SimilarlynextjobwithshortestexecutiontimeisP2fordurationof16-24second.

Systemtime	16
Processesschedulingfinishtime	P1,p3,p3,p4,p0,p2

Finishtime 16+8=24

Watingtime 24-8=16

TurnAroundtime

$$24-0=24$$

**Step 3:** Calculate average wating time and average turn around

$$\begin{aligned} \text{time.Averagewatingtime} &= (0+3+6+10+16)/5 \\ &= 35/5 \\ &= 7 \end{aligned}$$

$$\begin{aligned} \text{Averageturnaroundtime} &= (3+6+10+16+24)/5 \\ &= 59/5 \\ &= 11.8 \end{aligned}$$

## GnattChart

**Step4:** Afterschedulingof allprovidedprocesses.

Processid	Bursttime	Arrivaltime	Finishtime	TurnAround time	Watingtime
P1	3	0	0+3=3	3-0=3	3-3=0
P3	3	0	3+3=6	6-0=6	6-3=3
P4	4	0	6+4=10	10-0=10	10-4=6
<del>P0</del>	<del>6</del>	<del>0</del>	<del>10+6=16</del>	<del>16-0=16</del>	<del>16-6=10</del>
Batch 01 Shraddha Sawant					
P2	8	0	16+8=24	24-0=24	24-8=16

P1	P3	P4	P0	P2
----	----	----	----	----

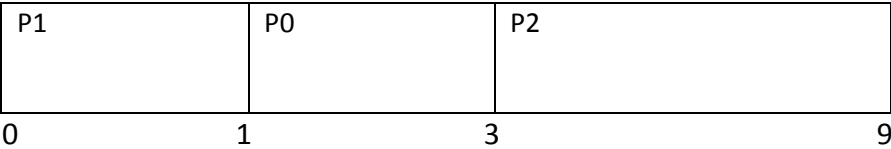
0                      3                      6                      10                      16                      24

**Example2:** Consider the following example containing five processes arrive at same time.

ProcessesID	BurstTime
P0	2
P1	1
P2	6

**GnattChart**

Processid	Bursttime	Arrivaltime	Finishtime	TurnAr oundti me	Watingtime
P1	1	0	1	1	1
P0	2	0	3	3	3
P2	6	0	9	9	9
Average				4.33333	1.33333





**Example3:** Consider the following example contain five processes arrive at same time.

Process ID	Bursttime				
P0	25				
P1	15				
P2	10				
P3	25				
P4	10				
P5	25				

Processid	Bursttime	Arrivaltime	Finishtime	Turn Aroundtim e	Watingtime
P2	10	0	10	10	0
P4	10	0	20	20	10
P1	15	0	35	35	20
P0	25	0	60	60	35
P3	25	0	85	85	60

Gnattchart:

Processid	Bursttime	Arrivaltime	Finishtime	TurnA round time	Watingtime
P2	10	0	10	10	0
P4	10	0	20	20	10
P1	15	0	35	35	20
P0	25	0	60	60	35
P3	25	0	85	85	60
P5	25	0	110	110	85
Average				53.3333	35.000000

P2	P4	P1	P0	P3	P5
0	10	20	35	60	85
					110

**Example4:** Consider the following example contain five processes arrive at same time.

ProcessId	BurstTime
P0	7
P1	3

P2	2
P3	10
P4	8

**Step4:** Afterschedulingof allprovidedprocesses.

Processid	Bursttime	Arrivaltime	Finishtime	TurnAr oundti me	Watingtime
P2	2	0	2	2	0
P1	3	0	5	5	2
P0	7	0	12	12	5
P4	8	0	20	20	12
P3	10	0	30	30	20

# SMT.Chandibai Himathmal Mansukhani College

---

Average				13.80000	7.800000
---------	--	--	--	----------	----------

```
//Name:Shraddha Sawant
//Batch:B1
//PRN:2020016400773862
//Date:24th July,2021
//Prac-02:SJF (with no preemption) Algorithm
```

```
import java.util.Scanner;

public class P2_SJF_SS {
    int burstTime[];
    int arrivalTime[] = {0};
    String[] processId;
    int numberOfProcess;

    void getProcessData(Scanner input) {
        System.out.print("Enter the number of Process for Scheduling: ");
        int inputNumberOfProcess = input.nextInt();
        numberOfProcess = inputNumberOfProcess;
        burstTime = new int[numberOfProcess];
        arrivalTime = new int[numberOfProcess];
        processId = new String[numberOfProcess];
        String st = "P";
        for (int i = 0; i < numberOfProcess; i++) {
            processId[i] = st.concat(Integer.toString(i));
            System.out.print("Enter the burst time for Process - " + (i+1) + " : ");
            burstTime[i] = input.nextInt();
        }
    }

    void sortAccordingBurstTime(int[] at,int[] bt,String[] pid){
        boolean swapped;
        int temp;
        String stemp;
        for (int i = 0; i < numberOfProcess; i++) {
            swapped = false;
            for (int j = 0; j < numberOfProcess - i - 1; j++){
                if (bt[j] > bt[j + 1]) {
                    temp = bt[j];
                    bt[j] = bt[j + 1];
                    bt[j + 1] = temp;
                    temp = at[j];
                    at[j] = at[j + 1];
                    at[j + 1] = temp;
                    stemp = pid[j];
                    pid[j] = pid[j + 1];
                    pid[j + 1] = stemp;
                    swapped = true;
                }
            }
        }
    }
}
```

```
if(swapped == false){
    break;
}
}
}

void shortestJobFirstNPAlgorithm() {
    int finishTime[] = new int [numberOfProcess];
    int bt[] = burstTime.clone();
    int at[] = arrivalTime.clone();
    String pid[] = processId.clone();
    int waitingTime[] = new int[numberOfProcess];
    int turnAroundTime[] = new int[numberOfProcess];
    sortAccordingBurstTime(at,bt,pid);
    finishTime[0] = at[0]+ bt[0];
    turnAroundTime[0] = finishTime[0] - at[0];
    waitingTime[0] = turnAroundTime[0] - bt[0];
    for (int i = 1; i < numberOfProcess; i++) {
        finishTime[i] = bt[i]+ finishTime[i - 1];
        turnAroundTime[i] = finishTime[i] - at[i];
        waitingTime[i] = turnAroundTime[i] - bt[i];
    }
    float sum = 0;
    for (int n: waitingTime){
        sum += n;
    }
    float averageWaitingTime = sum/ numberOfProcess;
    sum = 0;
    for (int n : turnAroundTime) {
        sum += n;
    }
    float averageTurnAroundTime = sum/ numberOfProcess;
    System.out.println("SJF (with no preemption) Scheduling Algorithm: ");
    System.out.format("%20s%20s%20s%20s%20s%20s\n", "ProcessId", "BurstTime", "ArrivalTime",
"FinishTime", "TurnAroundTime", "WaitingTime");
    for (int i = 0; i < numberOfProcess; i++) {
        System.out.format("%20s%20d%20d%20d%20d%20d\n", pid[i], bt[i], at[i],
finishTime[i],turnAroundTime[i],waitingTime[i]);
    }
    System.out.format("%80s%20f%20f\n", "Average", averageTurnAroundTime,averageWaitingTime);
}

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    P2_SJF_SS obj = new P2_SJF_SS();
    obj.getProcessData(input);
    obj.shortestJobFirstNPAlgorithm();
}
}
```

# SMT.Chandibai Himathmal Mansukhani College

---

## Sample Output1

```
Command Prompt
D:\os\p2>javac P2_S3F_SS.java

D:\os\p2>java P2_S3F_SS
enter the number of process for Scheduling:
3
enter the burst time for process-0:2
enter the burst time for process-1:1
enter the burst time for process-2:6
S3F (with no preemption) Scheduling Algorithm :
  ProcessId      BurstTime  ArrivallTime  FinishTime  TurnAroundTime  WatingTime
    p1           1           0                1             1             0
    p0           2           0                3             3             1
    p2           6           0                9             9             3
              Average      4.333333      1.333333

D:\os\p2>_
```

## Sample Output 2

```
Command Prompt
p4      10      0      20      20      10
p1      15      0      35      35      20
p0      25      0      60      60      35
p3      25      0      85      85      60
p5      25      0      110     110     85
Average 53.333332 35.000000

D:\os\p2>java P2_SJF_SS
enter the number of process for Scheduling:
5
enter the burst time for process-0:6
enter the burst time for process-1:3
enter the burst time for process-2:8
enter the burst time for process-3:3
enter the burst time for process-4:4
SJF (with no preemption) Scheduling Algorithm :
ProcessId    BurstTime    ArrivalTime    FinishTime    TurnAroundTime    WaitingTime
p1            3            0              3              3              0
p3            3            0              6              6              3
p4            4            0             10             10              6
p0            6            0             16             16             10
p2            8            0             24             24             16
Average      11.800000     7.000000

D:\os\p2>
```



## Sample Output 3

```
Command Prompt
p1      3      0      3      3      8
p3      3      0      6      6      3
p4      4      0      10     10     6
p0      6      0      16     16     10
p2      8      0      24     24     16
                        Average 11.800000 7.000000

D:\os\p2>java P2_SJF_SS
enter the number of process for Scheduling:
5
enter the burst time for process-0:7
enter the burst time for process-1:3
enter the burst time for process-2:2
enter the burst time for process-3:10
enter the burst time for process-4:8
SJF (with no preemption) Scheduling Algorithm :
ProcessId  BurstTime  ArrivalTime  FinishTime  TurnAroundTime  WaitingTime
p2         2         0           2           2           0
p1         3         0           5           5           2
p0         7         0          12          12           5
p4         8         0          20          20          12
p3        10         0          30          30          20
                        Average 13.800000 7.000000

D:\os\p2>
```

## Sample Output4

```
Command Prompt

Average          4.333333          1.333333

D:\os\p2>java P2_SJF_55
enter the number of process for Scheduling:
6
enter the burst time for process-0:25
enter the burst time for process-1:15
enter the burst time for process-2:10
enter the burst time for process-3:25
enter the burst time for process-4:10
enter the burst time for process-5:25
SJF (with no preemption) Scheduling Algorithm :
  ProcessId      BurstTime  ArrivalTime  FinishTime  TurnAroundTime  WatingTime
    p2           10         0             10          10             0
    p4           10         0             20          20             10
    p1           15         0             35          35             20
    p0           25         0             60          60             35
    p3           25         0             85          85             60
    p5           25         0            110         110             85
                        Average  53.333332    35.000000

D:\os\p2>
```

