Smt. Chandibai Himathmal Mansukhani College

USCS3PO1: USCS303 – Operating System (0S) Practical -01

First Come First Serve (FCFS) Algorithm

Contents

Batch : B2 Name: Sahil Jadhav

- **Algorithm**

First Come First Serve (FCFS) is also known as First In First Out (FIFO) Scheduling Algorithm is the easiest and simplest CPU scheduling algorithm where the process which arrives first in the ready queue is executed first by the CPU .New process is executed only when the current process is executed fully by the CPU.

Step 1: Input the number of processes required to be scheduled using FCFS , burst time for each process and its arrival time .

s**tep 2:** Using enhanced bubble sort technique ,sort the all given process in ascending order according to arrival time in a ready queue.

Step 3: Calculate the Finish Time ,Turn Around Time and Waiting Time for each process which in turn help to calculate Average Waiting Time and Average Turn Around Time required by CPU to schedule given set of process using FCFS.

Step 3.1: for i= 0, Finish Time T_0 = Arrival Time T_0 + Burst Time T_0

Step 3.2: for i >= 1, Finish Time T_i = Arrival Time T_i + Burst Time T_{i-1}

Step 3.3: for i = 0, Turn Around Time T_0 =Finish Time T_0 – Arrival Time T_0

Step 3.4: for i > = 1, Turn Around Time T_i =Finish Time T_i – Arrival Time T_i

Step 3.5: for i = 0, Waiting Time T_0 =Turn Around Time T_0 – Burst Time T_0

Step 3.6: for i >=1, Waiting Time T_i =Turn Around Time T_i – Burst Time T_{i-1}

Step 4: Process with less arrival time comes first and gets schedule first by the CPU.

Step 5: Calculate the Average Waiting Time and Average Turn Around Time .

Step 6: Stop.

- **Solved Example**

1. Consider following example containing five process arrive at same time.

Process ID	Burst Time
P0	6
P1	3
P2	8
P3	3
P4	4

Step 1: Processes get executed according to their arrival time.

Step 2: Following shows the scheduling and execution of processes.

Step 2.1: At start P0 arrives and get executed for 6 (i.e , 0-6) seconds.

System Time :0

Process Schedule :P0

Turn Around Time :6 – 0 = 6

Waiting Time :6 – 6 = 0

Step 2.2:P1 arrives after completion of P0, P1, is executed for 3 (i.e , 6 - 9) seconds.

System Time :6

Process Schedule :P0,P1

Turn Around Time :9 – 0 = 9

Waiting Time :9 – 3 = 6

Step 2.3: P2 arrives after complete execution for process P1, for 8(i.e; 9-17) seconds.

System Time :9

Process Schedule :P0,P1,P2

Turn Around Time :17 – 0 = 17

Waiting Time :17 – 8 = 9

Step 2.4:P3 arrives and gets executed for 3(i.e. 17- 20) seconds.

System Time :17

Process Schedule :P0,P1,P2,P3

Turn Around Time :20 – 0 = 20

Waiting Time :20 – 3 = 17

Step 2.5: Similarly , P4 arrives and gets executed for 4(i.e. 20 - 24) seconds.

System Time :20

Process Schedule :P0,P1,P2,P3,P4

Turn Around Time :24 – 0 = 24

Waiting Time :24 – 4 = 20

Step 3: Calculate the Average Waiting Time and Average Turn Around Time.

Average Waiting Time =(0 + 6 + 9 + 17 + 20)/5

=52/5

=10.4

Average Turn Around Time =(6 + 9 + 17 + 20 + 24)/5

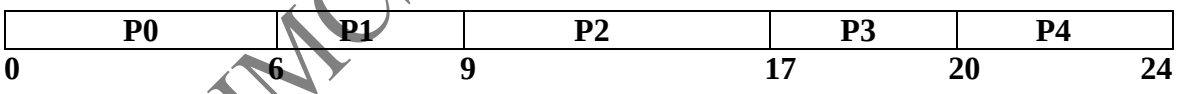
=76/5

=15.2

Step 4: After Scheduling of all provided processes :

Process ID	Burst Time	Arrival Time	Finish Time [Prev. Finish Time + Burst Time]	Turn Around Time [FinishTime-Arrival Time]	Waiting Time [TurnAround Time -Burst Time]
P0	6	0	(-+6=)6	(6-0=)6	(6-6=)0
P1	3	0	(6+3=)9	(9-0=)9	(9-3=)6
P2	8	0	(9+8=)17	(17-0=)17	(17-8=)9
P3	3	0	(17+3=)20	(20-0=)20	(20-3=)17
P4	4	0	(20+4=)24	(24-0=)24	(24-4=)20
Average				15.20000000	10.40000000

Gnatt Chart



2. Consider following example containing five process arrive at same time.

Process ID	Burst Time	Arrival Time
P0	6	2
P1	3	5
P2	8	1
P3	3	0
P4	4	4

Step 1: Processes get executed according to their arrival time.

Step 2: Following shows the scheduling and execution of processes.

Step 2.1: At start P3 arrives and get executed because its arrival time is 0 (i.e , 0-3) seconds.

System Time :0

Process Schedule :P3

Turn Around Time :3 – 3 = 0

Waiting Time :3– 0 = 3

Step 2.2:P2 arrives at time 1 sec during which CPU was busy with Process P3. After completion of P3 ,P2 is executed for (3 -11)seconds.

System Time :3

Process Schedule :P3,P2

Turn Around Time :10 – 8 = 2

Waiting Time :11 – 1 = 10

Step 2.3: P0 arrives at time 2 sec but its execution started at 11th sec after complete execution for process P2, for (11-17) seconds.

System Time :11

Process Schedule :P3,P2,P0

Turn Around Time :15 – 6 = 9

Waiting Time :17 – 2 = 15

Step 2.4:P4 arrives at time 4 sec but gets resources of CPU at 17th second and gets executed for (17- 21) seconds.

System Time :17

Process Schedule :P3,P2,P0,P4

Turn Around Time :17 – 4 = 13

Waiting Time :21 – 4 = 17

Step 2.5: Similarly , P1 arrives at time 5 sec but its execution gets started at time 21st second and lasts for 4(21 - 24) seconds.

System Time :21

Process Schedule :P3,P2,P0,P4,P1

Turn Around Time :19 – 3 = 16

Waiting Time :24 – 5 = 19

Step 3: Calculate the Average Waiting Time and Average Turn Around Time.

Average Waiting Time =(0 + 2 + 9 + 13 + 16)/5

=40/5

=8

Average Turn Around Time =(3 + 10 + 15 + 17 + 19)/5

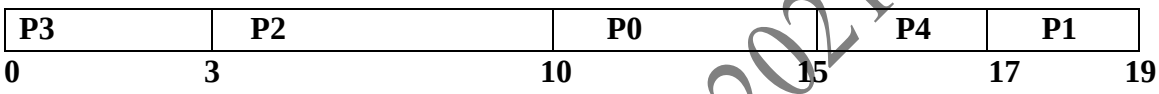
=64/5

=12.8

Step 4: After Scheduling of all provided processes :

Process ID	Burst Time	Arrival Time	Finish Time [Prev. Finish Time + Burst Time]	Turn Around Time [FinishTime-Arrival Time]	Waiting Time [TurnAround Time -Burst Time]
P3	3	0	(-+3=)3	(3-0=)3	(3-3=)0
P2	8	1	(3+8=)11	(11-1=)10	(10-8=)2
P0	6	2	(11+6=)17	(17-2=)15	(15-6=)9
P4	4	4	(17+4=)21	(21-4=)17	(17-4=)13
P1	3	5	(21+3=)24	(24-5=)19	(19-3=)16
Average				12.80000000	8.0000000

Gnatt Chart



3. Consider following example containing three process arrive at same time

Process ID	Burst Time
P0	2
P1	1
P2	6

Gnatt Chart :

Process ID	Burst Time	Arrival Time	Finish Time [Prev. Finish Time + Burst Time]	Turn Around Time [FinishTime-Arrival Time]	Waiting Time [TurnAround Time -Burst Time]
P0	2	0	(-+2=)2	(2-0=)2	(2-2=)0
P1	1	0	(2+1=)3	(3-0=)3	(3-1=)2
P2	6	0	(3+6=)9	(9-0=)9	(9-6=)3
Average				4.6666667	1.6666667

P0	P1	P2

0 2 3 9

4. Consider following example containing five process with varried arrival time.

Process ID	Burst Time	Arrival Time
P0	4	3
P1	3	5
P2	2	0
P3	1	5
P4	3	4

Gnatt Chart:

Process ID	Burst Time	Arrival Time	Finish Time [Prev. Finish Time + Burst Time]	Turn Around Time [FinishTime-Arrival Time]	Waiting Time [TurnAround Time -Burst Time]
P2	2	0	(-+2=)2	(2-0=)2	(2-2=)0
P0	4	3	(2+4=)6	(6-3=)3	(3-4=)-1
P4	3	4	(6+3=)9	(9-4=)5	(5-3=)2
P1	3	5	(9+3=)12	(12-5=)7	(7-3=)4
P3	1	5	(12+1=)13	(13-5=)8	(8-1=)7
Average				5.00000000	2.40000000

P2	P0	P4	P1	P3
0	2	3	5	7 8

Implementation

```
//Name:Jadhav Sahil
//Batch: B2
//PRN :2020016400783091
//Date :16th July ,2021

import java.util.Scanner;

public class P1_FCFS_SJ{


    int burstTime[];
    int arrivalTime[];
    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)+":");
      burstTime[i]=input.nextInt();
        System.out.print("Enter the arrival time for Process"+(i)+" :");
        arrivalTime[i]=input.nextInt();
    }
  }
void sortAccordingArrivalTime(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(at[j]>at[j+1]){
                  temp = at[j];
                  at[j] =at[j+1];
                  at[j+1]=temp;



              temp = bt[j];
                  bt[j] =bt[j+1];
                  bt[j+1]=temp;



              stemp = pid[j];
                   pid[j]=pid[j+1];
                   pid[j+1]=stemp;
```

```
            swapped=true;
           }
          }
        if(swapped==false){
            break;
            }
       }
   }
void firstComeFirstServeAlgorithm(){
    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];
    sortAccordingArrivalTime(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("FCFS Schedulling 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);
        P1_FCFS_SJ obj = new P1_FCFS_SJ();
        obj.getProcessData(input);
        obj.firstComeFirstServeAlgorithm();
     }
}
```

- **Input:**

Enter the number of process for scheduling:5

Enter the burst time for Process0:

Enter the arrival time for Process0:

Enter the burst time for Process1:

Enter the arrival time for Process1:

Enter the burst time for Process2:

Enter the arrival time for Process2:

Enter the burst time for Process3:

Enter the arrival time for Process3:

Enter the burst time for Process4:

Enter the arrival time for Process4

- **Output:**

Process ID	Burst Time	Arrival Time	Finish Time	Turn Around Time	Waiting Time
P3	3	0	(-+3=)3	(3-0=)3	(3-3=)0
P2	8	1	(3+8=)11	(11-1=)10	(10-8=)2
P0	6	2	(11+6=)17	(17-2=)15	(15-6=)9
P4	4	4	(17+4=)21	(21-4=)17	(17-4=)13
P1	3	5	(21+3=)24	(24-5=)19	(19-3=)16
Average				12.80000000	8.0000000

- **Sample OutPut:**

Question 1:

```
C:\Users\SAHIL\OneDrive\Documents\OSpracs>java P1_FCFS_sj.java
Enter the number of process for Scheduling:5
Enter the burst time for Process0:6
Enter the arrival time for Process0 :0
Enter the burst time for Process1:3
Enter the arrival time for Process1 :0
Enter the burst time for Process2:8
Enter the arrival time for Process2 :0
Enter the burst time for Process3:3
Enter the arrival time for Process3 :0
Enter the burst time for Process4:4
Enter the arrival time for Process4 :0
FCFS Schedulling Algorithm :
          ProcessId          BurstTime        ArrivalTime         FinishTime      TurnAroundTime        WaitingTime
                P0                  6                  0                  6                  6                  0
                P1                  3                  0                  9                  9                  6
                P2                  8                  0                 17                 17                  9
                P3                  3                  0                 20                 20                 17
                P4                  4                  0                 24                 24                 20
                                                                    Average          15.200000          10.400000

C:\Users\SAHIL\OneDrive\Documents\OSpracs>
```

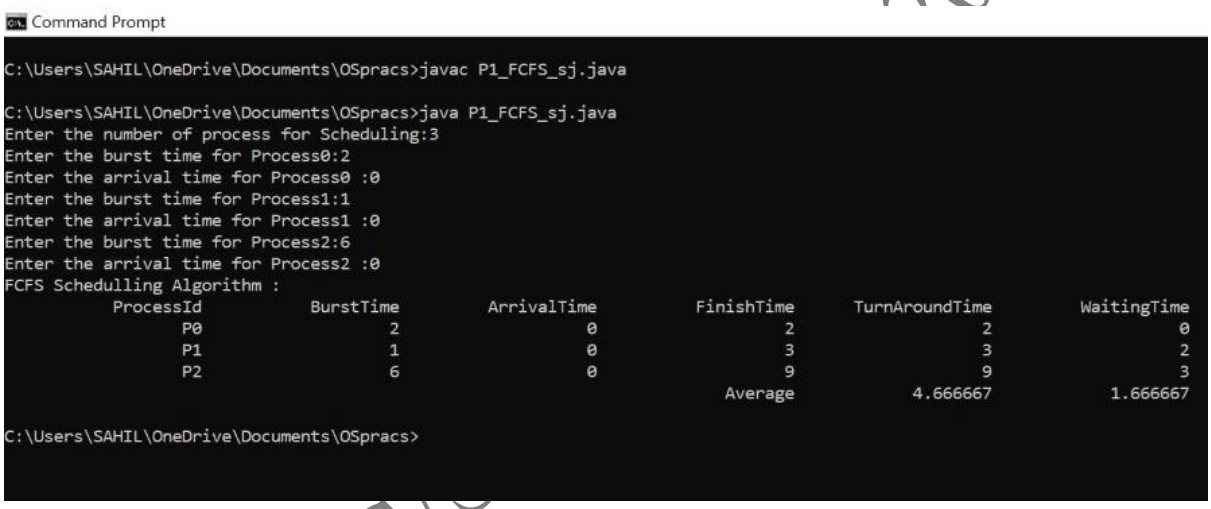
Question 2:

Command Prompt

```
C:\Users\SAHIL\OneDrive\Documents\OSpracs>java P1_FCFS_sj.java
Enter the number of process for Scheduling:5
Enter the burst time for Process0:6
Enter the arrival time for Process0 :2
Enter the burst time for Process1:3
Enter the arrival time for Process1 :5
Enter the burst time for Process2:8
Enter the arrival time for Process2 :1
Enter the burst time for Process3:3
Enter the arrival time for Process3 :0
Enter the burst time for Process4:4
Enter the arrival time for Process4 :4
FCFS Schedulling Algorithm :
          ProcessId          BurstTime        ArrivalTime         FinishTime      TurnAroundTime        WaitingTime
                P3                  3                  0                  3                  3                  0
                P2                  8                  1                 11                 10                  2
                P0                  6                  2                 17                 15                  9
                P4                  4                  4                 21                 17                 13
                P1                  3                  5                 24                 19                 16
                                                                    Average          12.800000           8.000000

C:\Users\SAHIL\OneDrive\Documents\OSpracs>
```

Question 3:



Command Prompt

```
C:\Users\SAHIL\OneDrive\Documents\OSpracs>javac P1_FCFS_sj.java

C:\Users\SAHIL\OneDrive\Documents\OSpracs>java P1_FCFS_sj.java
Enter the number of process for Scheduling:3
Enter the burst time for Process0:2
Enter the arrival time for Process0 :0
Enter the burst time for Process1:1
Enter the arrival time for Process1 :0
Enter the burst time for Process2:6
Enter the arrival time for Process2 :0
FCFS Schedulling Algorithm :
          ProcessId          BurstTime        ArrivalTime         FinishTime      TurnAroundTime        WaitingTime
                P0                  2                  0                  2                  2                  0
                P1                  1                  0                  3                  3                  2
                P2                  6                  0                  9                  9                  3
                                                                     Average          4.666667           1.666667

C:\Users\SAHIL\OneDrive\Documents\OSpracs>
```


Question 4:

```
C:\Users\SAHIL\OneDrive\Documents\OSpracs>java P1_FCFS_sj.java
Enter the number of process for Scheduling:5
Enter the burst time for Process0:4
Enter the arrival time for Process0 :3
Enter the burst time for Process1:3
Enter the arrival time for Process1 :5
Enter the burst time for Process2:2
Enter the arrival time for Process2 :0
Enter the burst time for Process3:1
Enter the arrival time for Process3 :5
Enter the burst time for Process4:3
Enter the arrival time for Process4 :4
FCFS Schedulling Algorithm :
          ProcessId          BurstTime        ArrivalTime         FinishTime      TurnAroundTime        WaitingTime
                 P2                  2                  0                  2                  2                  0
                 P0                  4                  3                  6                  3                 -1
                 P4                  3                  4                  9                  5                  2
                 P1                  3                  5                 12                  7                  4
                 P3                  1                  5                 13                  8                  7
                                                                    Average          5.000000           2.400000

C:\Users\SAHIL\OneDrive\Documents\OSpracs>
```