Contents

**USCSP301- USCS303:Operating System (OS) Practical-01**

**Practial-01:**First come serve (FCFS) Algorithm……………………………..1

**Practical Date:** 17th July.2021………………………………………………. ......2

**Practical Aim:** Implement FCFS scheduling algorithm using Java...3

**Algorithm**………………………………………………………………………..............4

Solved Example…………………………………………………………………...........5

Gnatt Chart………………………………………………………………………............6

Implementation…………………………………………………………………...........7

Input………………………………………………………………………….....................8

Output...............................................................................................9

Sample Output -01............................................................................10

Sample Output -02............................................................................11

1) Algorithm

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

\* Scheduling algorithm are broadly classified into two main type namely preemptive and non-preemptive .

\* FIRST COME FIRST OUT(FCFS) is also know as FIRST IN FIRST OUT (FIFO) SCHEDUAL algorithm is theand simplest CPU.

\* A process scheduling different process to be assigned to the CPU based on particular scheduling algorithm thereare six popular process scheduling algorithm which we are going to discuss in this chapter FIRST COME FIRST OUT SHEDULING . 1) Algorithm

**EXAMPLE 1:** Consider the following example containing five process arrive at same time.

|  |  |
| --- | --- |
| **Process ID** | **Times new** |
| P0 | 6 |
| P1 | 3 |
| P2 | 8 |
| P3 | 3 |
| P4 | 4 |

**SOLVE:**

**Step 1:** *Process get execute according to their arrival time*.

**Step 2:** *Following show the scheduling and execute of process .*

**Step 2.2:** *At start p0 arrive and get execute for 6 seconds.*

|  |  |
| --- | --- |
| **System time** | 0 |
| **Process Schedualed** | P0 |
| **Turn around time** | 6-0=6 |
| **Wating Time** | 6-6=0 |

**Step 2.2:** p1 arrive after completion of p0 , p1 is execute for 3

|  |  |
| --- | --- |
| **System time** | 6 |
| **Process Schedualed** | P0,p1 |
| **Turn around time** | 9-0=9 |
| **Wating Time** | 9-3=6 |

**Step2.3*:*** *p2 arrive after complete execution of process p1 for 8.*

|  |  |
| --- | --- |
| **System time** | 9 |
| **Process Schedualed** | P0,p1,p2 |
| **Turn around time** | 17-0=17 |
| **Wating Time** | 17-8=9 |

**Step 2.4:***p3 arrive and gets execute for 3.*

|  |  |
| --- | --- |
| **System time** | 17 |
| **Process Schedualed** | P0,p1,p2,p3 |
| **Turn around time** | 20-0=20 |
| **Wating time** | 20-3=17 |
|  |  |

**Step 2.5:** *similary p4 arrives gets execute for 4.*

|  |  |
| --- | --- |
| **System time** | 20 |
| **Process Schedualed** | P0,p1,p2,p3,p4 |
| **Turn around time** | 24-0=24 |
| **Wating time** | 24-4=20 |

**Step 3:** *calculate average wating time and average turn around time*.

Average wating 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

**Gnatt Chart.**

**Step 4:** *after scheduling of all provided processes*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** | **Finish time** | **Turn Around time** | **Wating time** |
| P0 | 6 | 0 | 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 |  |  |  | 15.200000 | 10.400000 |

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

**EXAMPLE 2:**

Consider the following example contain five with varied arrive time.

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Burst Time** | **Arrival Time** |
| P0 | 6 | 2 |
| P1 | 3 | 5 |
| P2 | 8 | 1 |
| P3 | 3 | 0 |
| P4 | 4 | 4 |

**Step 1:** *Process get execute according to their arrival time.*

**Step 2:** *Following show the scheduling and execute of process .*

**Step 2.2:** *At start p3 arrive and get execute for 0-3 second.*

|  |  |
| --- | --- |
| **System time** | 0 |
| **Process Schedualed** | P3 |
| **Turn around time** | 3-0=3 |
| **Wating Time** | 3-3=0 |

**Step 2.3:** *p0 arrives at time 4 sec but gets resource of cpu at 17 second for execution its execution period is 17-21 second*.

|  |  |
| --- | --- |
| **System time** | 11 |
| **Process Schedualed** | P3.p2,p0 |
| **Turn around time** | 17-2=15 |
| **Wating Time** | 15-6=13 |

**Step 2.4*:*** *p4 arrives at time 4 sec but gets resource of cpu at 17 second for execution period is 17-21 second.*

|  |  |
| --- | --- |
| **System time** | 17 |
| **Process Schedualed** | P0,p1,p2,p3 |
| **Turn around time** | 20-0=20 |
| **Turn around time** | 20-3=17 |
|  |  |

**Step 2.5:** *similarly p1 arrives at time 5 sec but its execution gets started turn around time 21 second and last for a period*

*21-24 second.*

|  |  |
| --- | --- |
| **System time** | 21 |
| **Process Schedualed** | P3,p2,p0,p4,p1 |
| **Turn around time** | 24-5=19 |
| **Turn around time** | 19-5=19 |

**Step 3:** *calculate average wating time and average turn around time.*

Average wating time =(0+2+9+13+16)/5

=40/5

=8

Average turn around time :(3+10+15+17+19)/5

=64/5

=12.8

**Gnatt Chart.**

**Step 4:** *after scheduling of all provided processes.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** | **Finish time** | **Turn Around time** | **Wating time** |
| P3 | 3 | 0 | 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 |  |  |  | 12.8000000 | 8.000000 |

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

0 3 11 17 21 24

**EXAMPLE 3:** *Consider the following example containing five processes arrive at the Same time .*

|  |  |
| --- | --- |
| **Process ID** | **Times new** |
| P0 | 2 |
| P1 | 1 |
| P2 | 6 |

**SOLVE:**

**Step 1:** *Process get execute according to their arrival time.*

**Step 2:** *Following show the scheduling and execute of process .*

**Step 2.1:** *At start p0 arrive and get execute for 2 second.*

|  |  |
| --- | --- |
| **System time** | 0 |
| **Process Schedualed** | P0 |
| **Turn around time** | 2-0=2 |
| **Wating Time** | 2-2=0 |

**Step 2.2:** p1 arrive after completion of p0 , p1 is execute for 1.

|  |  |
| --- | --- |
| **System time** | 2 |
| **Process Schedualed** | P0,p1 |
| **Turn around time** | 3-0=3 |
| **Wating Time** | 3-1=2 |

**Step2.3:** *p2 arrive after complete execution of process p1 for 6.*

|  |  |
| --- | --- |
| **System time** | 3 |
| **Process Schedualed** | P0,p1,p2 |
| **Turn around time** | 9-0=17 |
| **Wating Time** | 9-6=3 |

**Step 3:** *calculate average wating time and average turn around time.*

Average wating time =(0+2+3)/3

=5/3

=1.6666

Average turn around time :(2+3+9)/

=14/3

=4.6666

|  |
| --- |
| **Gnatt Chart.** |

**Step 4:** *after scheduling of all provided processes.*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Process id** | | **Burst time** | | **Arrival time** | | **Finish time** | | **Turn Around time** | | | **Wating time** |
| P0 | | 2 | | 0 | | 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 | | 9-6=3 | 9-6=3 | | |
|  | |  | |  | |  | |  |  | | |
|  | |  | |  | |  | |  |  | | |
| AVERAGE | |  | |  | |  | | 4.666 | 1.666 | | |

|  |  |  |
| --- | --- | --- |
| P0 | P1 | P2 |

0 2 3 9

**EXAMPLE 4:** *Consider the following example containing five process with varied arrival time.*

|  |  |  |
| --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** |
| P0 | 4 | 3 |
| P1 | 3 | 5 |
| P2 | 2 | 0 |
| P3 | 1 | 5 |
| P4 | 3 | 4 |

**Step 3:** *calculate average wating time and average turn around time*.

Average wating time =(3+1+7+4+6)/5

=21/5

=4.2

Average turn around time :(1+2+9+5+9)/5

=26/5

=5.2

**Gnatt Chart.**

**Step 4:** *after scheduling of all provided processes*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** | **Finish time** | **Turn Around time** | **Wating time** |
| P0 | 4 | 3 | 6 | 3 | 1 |
| P1 | 3 | 5 | 12 | 7 | 4 |
| P2 | 2 | 0 | 2 | 2 | 0 |
| P3 | 1 | 5 | 13 | 8 | 7 |
| P4 | 3 | 4 | 9 | 5 | 92 |
| AVERAGE |  |  |  | 5.0000 | 2.40000 |

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

2 6 9 12 13 0

**IMPLEMENATION:**

import java.util.Scanner;

public class P1\_FCFS\_SS

{

int burstTime[]; int arrivalTime[]; String[] processId;

int numberOfProcess;

void getProcessData(Scanner input){

System.out.println("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.println("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 Scheduling algorithm :"); System.out.format("%20s%20s%20s%20s%20s%20s\n","ProcessId","BurstTime"

,"ArrivalTime","FinishTime","TurnAroundTime","WatingTime"); 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);

}

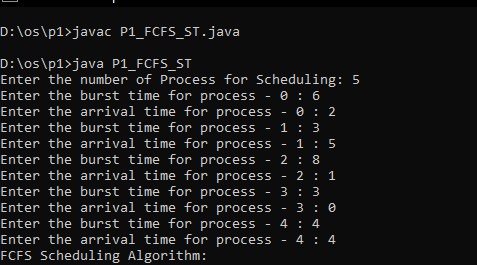
public static void main(String[] args){ Scanner input=new Scanner(System.in); P1\_FCFS\_SS obj=new P1\_FCFS\_SS(); obj.getProcessData(input);

obj.firstComeFirstServeAlgorithm();

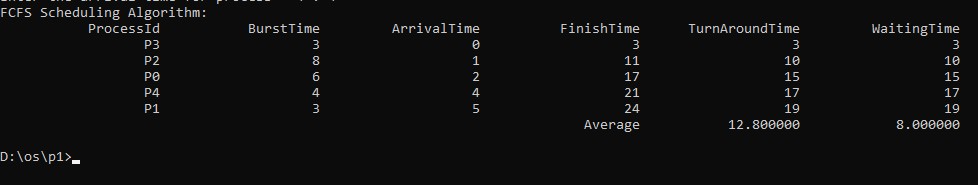
}

}

**INPUT:**



**OUTPUT:**



**SAMPLE OUTPUT:-**

