LAB : FCFS & SRTF

Name of the Student: **Rahebar Shaikh** Div. **AI & DS - B**  Roll No.**72** PRN:**12120069**

**Aim**: Implementation of FCFC and SRTF

FCFS

Code-

import java.util.\*;

public class FCFS {

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("enter no of process: ");

int n = sc.nextInt();

int pid[] = new int[n]; // process ids

int ar[] = new int[n]; // arrival times

int bt[] = new int[n]; // burst or execution times

int ct[] = new int[n]; // completion times

int ta[] = new int[n]; // turn around times

int wt[] = new int[n]; // waiting times

int temp;

float avgwt=0,avgta=0;

for(int i = 0; i < n; i++)

{

System.out.println("enter process " + (i+1) + " arrival time: ");

ar[i] = sc.nextInt();

System.out.println("enter process " + (i+1) + " brust time: ");

bt[i] = sc.nextInt();

pid[i] = i+1;

}

//sorting according to arrival times

for(int i = 0 ; i <n; i++)

{

for(int j=0; j < n-(i+1) ; j++)

{

if( ar[j] > ar[j+1] )

{

temp = ar[j];

ar[j] = ar[j+1];

ar[j+1] = temp;

temp = bt[j];

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

bt[j+1] = temp;

temp = pid[j];

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

pid[j+1] = temp;

}

}

}

// finding completion times

for(int i = 0 ; i < n; i++)

{

if( i == 0)

{

ct[i] = ar[i] + bt[i];

}

else

{

if( ar[i] > ct[i-1])

{

ct[i] = ar[i] + bt[i];

}

else

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

}

ta[i] = ct[i] - ar[i] ; // turnaround time= completion time- arrival time

wt[i] = ta[i] - bt[i] ; // waiting time= turnaround time- burst time

avgwt += wt[i] ; // total waiting time

avgta += ta[i] ; // total turnaround time

}

System.out.println("\npid arrival brust complete turn waiting");

for(int i = 0 ; i< n; i++)

{

System.out.println(pid[i] + " \t " + ar[i] + "\t" + bt[i] + "\t" + ct[i] + "\t" + ta[i] + "\t"

+ wt[i] ) ;

}

sc.close();

System.out.println("\naverage waiting time: "+ (avgwt/n)); // printing average waiting time.

System.out.println("average turnaround time:"+(avgta/n)); // printing average turnaround time.

}

}

Output:

Text

Description automatically generated

**SRTF**

Code:

import java.io.\*;

public class SRTF {

public static void main(String args[]) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int n;

System.out.println("Please enter the number of Processes: ");

n = Integer.parseInt(br.readLine());

int proc[][] = new int[n + 1][4];//proc[][0] is the AT array,[][1] - RT,[][2] - WT,[][3] - TT

for(int i = 1; i <= n; i++)

{

System.out.println("Please enter the Arrival Time for Process " + i + ": ");

proc[i][0] = Integer.parseInt(br.readLine());

System.out.println("Please enter the Burst Time for Process " + i + ": ");

proc[i][1] = Integer.parseInt(br.readLine());

}

System.out.println();

//Calculation of Total Time and Initialization of Time Chart array

int total\_time = 0;

for(int i = 1; i <= n; i++)

{

total\_time += proc[i][1];

}

int time\_chart[] = new int[total\_time];

for(int i = 0; i < total\_time; i++)

{

//Selection of shortest process which has arrived

int sel\_proc = 0;

int min = 99999;

for(int j = 1; j <= n; j++)

{

if(proc[j][0] <= i)//Condition to check if Process has arrived

{

if(proc[j][1] < min && proc[j][1] != 0)

{

min = proc[j][1];

sel\_proc = j;

}

}

}

//Assign selected process to current time in the Chart

time\_chart[i] = sel\_proc;

//Decrement Remaining Time of selected process by 1 since it has been assigned the CPU for 1 unit of time

proc[sel\_proc][1]--;

//WT and TT Calculation

for(int j = 1; j <= n; j++)

{

if(proc[j][0] <= i)

{

if(proc[j][1] != 0)

{

proc[j][3]++;//If process has arrived and it has not already completed execution its TT is incremented by 1

if(j != sel\_proc)//If the process has not been currently assigned the CPU and has arrived its WT is incremented by 1

proc[j][2]++;

}

else if(j == sel\_proc)//This is a special case in which the process has been assigned CPU and has completed its execution

proc[j][3]++;

}

}

//Printing the Time Chart

if(i != 0)

{

if(sel\_proc != time\_chart[i - 1])

//If the CPU has been assigned to a different Process we need to print the current value of time and the name of

//the new Process

{

System.out.print("--" + i + "--P" + sel\_proc);

}

}

else//If the current time is 0 i.e the printing has just started we need to print the name of the First selected Process

System.out.print(i + "--P" + sel\_proc);

if(i == total\_time - 1)//All the process names have been printed now we have to print the time at which execution ends

System.out.print("--" + (i + 1));

}

System.out.println();

System.out.println();

//Printing the WT and TT for each Process

System.out.println("P\t WT \t TT ");

for(int i = 1; i <= n; i++)

{

System.out.printf("%d\t%2dms\t%2dms",i,proc[i][2],proc[i][3]);

System.out.println(); }

System.out.println();

//Printing the average WT & TT

float WT = 0,TT = 0;

for(int i = 1; i <= n; i++)

{ WT += proc[i][2];

TT += proc[i][3]; }

WT /= n;

TT /= n;

System.out.println("The Average WT is: " + WT + "ms");

System.out.println("The Average TT is: " + TT + "ms");

} }

Output:

Text

Description automatically generated