/” Problem Statement: Write a JAVA program (using oop features) to implement following

1. FCFS
2. SJF(Preemptive)
3. Priority(Non- Preemptive)
4. Round Robin(Preemptive)

1.FCFS

”/

import java.io.\*;

import java.utiI.Scanner; public class FCFS

public static void main(String args[])

int i,no\_p,burst\_time[],TT[],WT[]; float avg\_wait=0,avg\_TT=0; burst\_time=new int[50];

TT=new int[50];

WT=new int[50];

WT[0]=0;

Scanner s=new Scanner(System.in); System.out.println("Enter the number of process: "); no\_p=s.nextInt();

System.out.printIn("\nEnter Burst Time for processes:"); for(i=0;i<no\_p;i++)

System.out.print("\tP"+(i+1)+": "); burst\_time[i]=s.nextlnt();

for(i=1;i<no\_p;i++)

WT[i]=WT[i-1]+burst\_time[i-1]; avg\_wait+=WT[i];

avg\_wait/=no\_p; for(i=0;i<no\_p;i++)

TT[i]=WT[i]+burst\_time[i]; avg\_TT+=TT[i];

avg\_TT/=no\_p;

System.out.println("\n\*\*”\*\*\*\*\*\*”\*\*\*\*\*\*\*”\*\*\*\*\*\*\*\*\*\*\*\*\*”\*\*\*\*””\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); System.out.printIn("\tProcesses:");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*””\*\*\*\*”””\*\*\*\*\*””\*\*\*\*\*””\*\*\*\*\*”\*\*\*\*\*\*”\*”\*\*\*”””\*\*\*\*””\*"); System.out.println(" Process\tBurst Time\tWaiting Time\tTurn Around Time"); for(i=0;i<no\_p;i++)

System.out.printIn("\tP"+(i+1)+"\t "+burst\_time[i]+"\t\t "+WT[i]+"\t\t "+TT[i]);

System.out.println("\n ");

System.out.printIn("\nAverage waiting time : "+avg\_wait); System.out.printIn("\nAverage Turn Around time : "+avg\_TT+"\n");

/\*Output:

Enter the number of process: 3

Enter Burst Time for processes: P1: 24

P2: 3

P3: 3



Processes:



|  |  |  |  |
| --- | --- | --- | --- |
| Process | Burst Time | Waiting Time | Turn Around Time |
| P1 | 24 | 0 | 24 |
| P2 | 3 | 24 | 27 |
| P3 | 3 | 27 | 30 |

Average waiting time : 17.0

Average Turn Around time : 27.0 \*/

/”Round Robin(Preemptive)\*/ import java.utiI.\*;

import java.io.”;

class RoundR

public static void main(String args(])

int Process[]=new int[10]; int a[]=new int[10];

int Arrival\_time[]=new int[10]; int Burst\_time[]=new int[10]; int WT[]=new int[10];

int TAT[]=new int[10]; int Pno,sum=0;;

int TimeQuantum;

System.out.printIn("\nEnter the no. of Process::");

Scanner sc=new Scanner(System.in); Pno=sc.nextlnt(); System.out.println("\nEnter each process::"); for(int i=0;i<Pno;i++)

Process[i]=sc.nextlnt();

System.out.printIn("\nEnter the Burst Time of each process::"); for(int i=0;i<Pno;i++)

Burst\_time[i]=sc.nextlnt();

System.out.printIn("\nEnter the Time Quantum::"); TimeQuantum=sc.nextlnt();

do(

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

if(Burst\_time[i]>TimeQuantum) Burst\_time[i]-=TimeQuantum;

for(int j=0;j<Pno;j++)

if((j!=i)&&(Burst\_time[j]!=0)) WTO]+=TimeQuantum;

else

for(int j=0;j<Pno;j++)

if((j!=i)&&(Burst\_time[j]!=0)) WTO]+=Burst\_time[i];

Burst\_time[i]=0;

sum=0;

for(int k=0;k<Pno;k++) sum=sum+Burst\_time[k];

} while(sum!=0);

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

TAT[i]=WT[i]+a[i];

System.out.printIn("process\t\tBT\tWT\tTAT"); for(int i=0;i<Pno;i++)

System.out.printIn("process"+(i+1)+"\t"+a[i]+"\t"+WT[i]+"\t"+TAT[i]); float avg\_wt=0;

float avg tat=0;

for(int j=0;j<Pno;j++) avg wt+=WT§];

for(int j=0;j<Pno;j++) avg\_tat+=TAT§];

System.out.println("average waiting time "+(avg\_wt/Pno)+"\n Average turn around time"+(avg\_tat/Pno));

/\*OUTPUT::

unix@unix-HP-280-G1- MT:”/TEA33$ java RoundR Enter the no. of Process:: 5

Enter each process::

1

2

3

4

5

Enter the Burst Time of each process::

2

1

8

4

5

Enter the Time Quantum::

2

|  |  |  |  |
| --- | --- | --- | --- |
| process | BT | WT | TAT |
| process1 | 0 | 0 | 0 |
| process2 | 0 | 2 | 2 |
| process3 | 0 | 12 | 12 |
| process4 | 0 | 9 | 9 |
| process5 | 0 | 13 | 13 |

average waiting time 7.2

Average turn around time7.2 \*/

/” 2. SJF(Non-Preemptive) \*/ import java.utiI.Scanner;

class SJF1(

public static void main(String args[])(

int burst\_time[],process[],waiting\_time[],tat[],i,j,n,totaI=0,pos,temp; float wait\_avg,TAT\_avg;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of process: "); n = s.nextInt();

process = new int[n]; burst\_time = new int[n]; waiting time = new int[n]; tat = new int[n];

System.out.printIn("\nEnter Burst time:"); for(i=0;i<n;i++)

System.out.print("\nProcess["+(i+1)+"]: "); burst\_time[i] = s.nextInt();;

process[i]=i+1; //Process Number

//Sorting for(i=0;i<n;i++)

pos=i; for(j=i+1;j<n;j++)

if(burst\_time§]<burst\_time[pos]) pos=j;

temp=burst\_time[i]; burst\_time[i]=burst\_time(pos]; burst\_time[pos]=temp;

temp=process[i]; process[i]=process[pos]; process[pos]=temp;

//First process has 0 waiting time waiting\_time[0]=0;

//calculate waiting time for(i=1;i<n;i++)

waiting\_time[i]=0; for(j=0;j<i;j++) waiting\_time[i]+=burst\_time[j]; total+=waiting\_time[i];

//Calculating Average waiting time wait\_avg=(float)totaI/n;

totaI=0;

System.out.printIn("\nProcess\I Burst Time \tWaiting Time\tTurnaround Time"); for(i=0;i<n;i++)

tat[i]=burst\_time[i]+waiting\_time[i]; //Calculating Turnaround Time total+=tat(i];

System.out.println("\n p"+process[i]+"\t\I "+burst\_time[i]+"\t\I "+waiting\_time[i]+"\t\t "+tat[i]);

//Calculation of Average Turnaround Time TAT\_avg=(float)total/n;

System.out.printIn("\n\nAverage Waiting Time: "+wait avg); System.out.printIn("\nAverage Turnaround Time: "+TAT\_avg);

/” 2. SJF(Preemptive)\*/ import java.utiI.Scanner;

class sjf\_swap1{

public static void main(String args[])

int burst\_time[],process[],waiting\_time[],tat[],arr\_time[],compIetion\_t\me[],i,j,n,totaI=0,totaI\_comp=0,pos,t emp;

float wait avg,TAT avg;

Scanner s = new Scanner(System.in); System.out.print("Enter number of process: "); n = s.nextInt();

process = new int[n]; burst\_time = new int[n]; waiting\_time = new int[n]; arr\_time=new int[n];

tat = new int[n]; compIetion\_time=new int(n];

//burst time

System.out.printIn("\nEnter Burst time:"); for(i=0;i<n;i++)

System.out.print("\nProcess["+(i+1)+"]: "); burst time[i] = s.nextInt();;

process[i]=i+1; //Process Number

//arrival time

System.out.printIn("\nEnter arrival time:"); for(i=0;i<n;i++)

System.out.print("\nProcess["+(i+1)+"]: "); arr\_time[i] = s.nextInt();;

process[i]=i+1; //Process Number

//Sorting for(i=0;i<n;i++)

pos=i; for(j=i+1;j<n;j++)

if(burst\_timeb]<burst\_time[pos])

pos=j;

temp=burst\_time[i]; burst\_time[i]=burst\_time[pos]; burst\_time[pos]=temp;

temp=process[i]; process[i]=process[pos]; process[pos]=temp;

System.out.printIn("process"+process[i]);

//completion time new for(i=1;i<n;i++)

compIetion\_time[i]=0; for(j=0;j<i;j++) compIetion\_time[i]+=burst\_time[j]; totaI\_comp+=compIetion\_t1me[i];

//First process has 0 waiting time

waiting\_time[0]=0;

//calculate waiting time for(i=1;i<n;i++)

waiting\_time[i]=0; for(j=0;j<i;j++) waiting\_time[i]+=burst\_time[j]; total+=waiting\_time[i];

//Calculating Average waiting time wait\_avg=(float)totaI/n;

total=0;

System.out.printIn("\nPro\_number\t Burst Time \tcompIetion\_time\tWaiting Time\tTurnaround Time"); for(i=0;i<n;i++)

tat[i]=burst\_time[i]+waiting\_time[i];

//Calculating Turnaround Time total+=tat[i];

System.out.printIn("\n"+process[i]+"\t\t "+burst\_time[i]+"\t\t "+compIetion\_time[i]+"\t\t"+waiting\_time[i]+"\t\t "+tat[i]);

//Calculation of Average Turnaround Time TAT\_avg=(float)total/n; System.out.printIn("\n\nAWT: "+wait\_avg); System.out.printIn("\nATAT: "+TAT\_avg);