Practical 5

AIM: Write a program that implements RR scheduling algorithm.

CODE:

public class GFG

{

static void findWaitingTime(int processes[], int n, int bt[], int wt[], int quantum)

{

int rem\_bt[]=new int[n];

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

rem\_bt[i]=bt[i];

int t=0;

while(true)

{

boolean done=true;

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

{

if (rem\_bt[i]>0)

{

done=false;

if (rem\_bt[i]>quantum)

{

t+=quantum;

rem\_bt[i]-=quantum;

}

else

{

t = t+rem\_bt[i];

wt[i]=t-bt[i];

rem\_bt[i]=0;

}

}

}

if(done==true)

break;

}

}

static void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[])

{

//bt[i]+wt[i]

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

tat[i]=bt[i]+wt[i];

}

static void findavgTime(int processes[], int n, int bt[], int quantum)

{

int wt[]=new int[n], tat[]=new int[n];

int total\_wt=0, total\_tat=0;

findWaitingTime(processes, n, bt, wt, quantum);

findTurnAroundTime(processes, n, bt, wt, tat);

System.out.println("Process No.\t"+"Burst Time\t"+"Waiting Time\t"+"TurnAroundTime");

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

{

total\_wt=total\_wt+wt[i];

total\_tat=total\_tat+tat[i];

System.out.println(""+(i+1)+"\t\t"+bt[i]+"\t\t"+wt[i]+"\t\t"+tat[i]);

}

System.out.println("Average Waiting Time= "+(float)total\_wt/(float)n);

System.out.println("Average Turn Around Time= "+(float)total\_tat/(float)n);

}

public static void main(String[] args)

{

int processes[]={1, 2, 3};

int n=processes.length;

int burst\_time[]={10, 5, 8};

int quantum=2;

findavgTime(processes, n, burst\_time, quantum);

}

}

OUTPUT:

Process No. Burst Time Waiting Time TurnAroundTime

1 10 13 23

2 5 10 15

3 8 13 21

Average Waiting Time= 12.0

Average Turn Around Time= 19.666666