Smt.Chandibai Himathmal Manuskhani College

USCS3P01: USCS303-Operating System(OS) Practical-03

Date:28/7/2021

Round-Robin Scheduling algorithm

Contents

Aim: Round-Robin Scheduling algorithm

Algorithm:

Content:

CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way.

Process:

Input the number of process and time quantum or slice required .

Calculate the finish time, Turn around time and waiting time for each process.

Calculate Average Waiting Time and Average Turn Around Time required by CPU to scheduling given set

Of process using RR.

Prior Knowledge:

Basic of java programming language , Cyclic queue traveling , average.

Round-robin (RR) scheduling algorithm is mainly designed for time -sharing system.

This algorithm is similar to FCFS scheduling , but in round robin scheduling , preemption is added which

Enables the system to switch between processes.

Step l:Input the number of process and time quanta or time slice required to be scheduling using RR , burst time for each process.

Step 2: Choose the first process in the ready queue , set a timer to interrupt it after quantum and dispatches it. Check if any other process has arrived . if a process request arrives during the quantum time in which another process is executed then add the new process to the ready queue.

Step 3: After the quantum time has passed , check for any processes in the ready queue . if the ready queue is empty.

Then continue the current process . if the queue not empty and the current process is not complete , then add Add the current process to the end of the queue.

Step 4: Take the first process from the ready queue and start executing it. Calculate the Turn Around Time and Wating Time for each process using RR.

Step 5: Repeat all step above from Step 2 to Step 4.

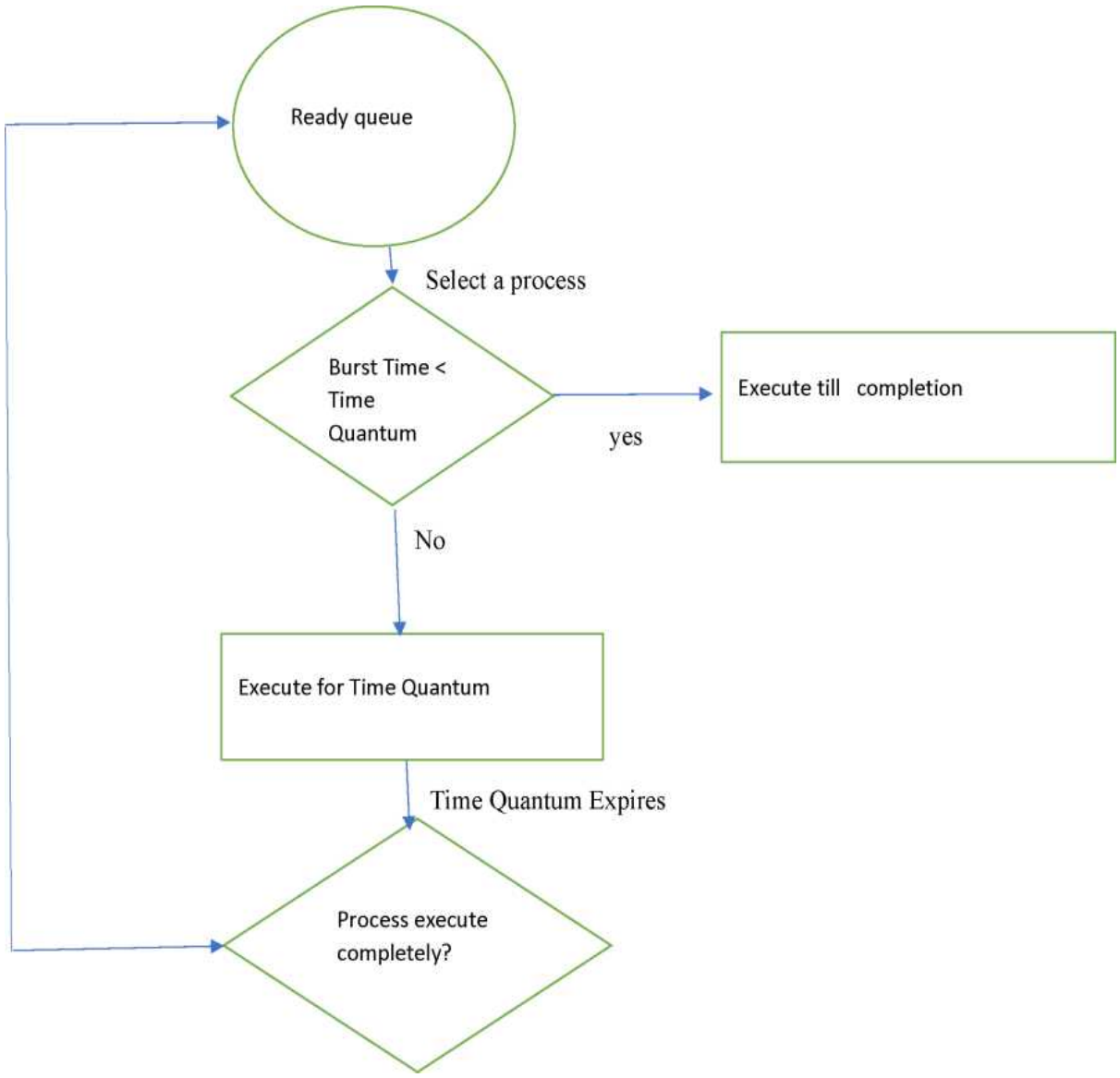
Step 6: If the process is complete and ready queue is empty then the task is complete.

Step 7: Calculate the Average Waiting Time and Average Turn Around Time.

Step 8: Stop.

Flowchart:

RR Flowchart



Example 1: Consider the following example containing three processes arriving at time t=0 ms.

Process ID	Burst Time
PO	24
Pl	3
P2	3

Assume time Quanta : 4 ms.

Step 1: Consider the time Quanta / time slice = 4ms.

Step 2: Following show the scheduling and execution of process.

Step 2.1: PO process arrive at 0 with 24 me as the burst time which is greater than time quanta = 4 ms. So pO execute for 4 ms and goes in waiting queue.

System Time	0
Process scheduling	PO
Remaining Time	24-4=20
Waiting Time	0-0=0
Turn Around Time	0+4=4

Step 2.2: Next Pl process execute for 3 ms which is greater than quanta time. So Pl executes and get terminated.			
	System Time	7	
	Process scheduling	PO,P1	
	Remaining Time	3-4=-l=0	
	Waiting Time	4-0=4	
	Turn Around Time	4+3=7	
	Step 2.3: Next P2 process execute for 3 ms which is greater than quanta time. So P2 execute and gets terminated .		
	System Time	7	
	Process scheduling	PO,P1,P2	
	Remaining Time	3-4=-l=0	
	Waiting Time	7-0=7	
	Turn Around Time	7+3=10	

Step 2.4: Now pO turn comes again and it's the only process for execute so for 4 ms quanta it gets executed.			
7	System Time	10	
	Process scheduling	P0,Pl,P2,P0	
	finish Time	20-4=16	
	Waiting Time	10-4=7	
	Turn Around Time	10+4=14	
Step 2.5: Again pO continue to execute for next 4 ms. Waiting for pO will be zero.			
	System Time	14	
	Process scheduling	P0,Pl,P2,P0,P0	
	Finish Time	16-4=12	
	Waiting Time	0	
	Turn Around Time	14+4=18	

.

Step 2.6: PO continue execute for next 4 ms.

System Time	18
Process scheduling	P0,Pl,P2,P0,P0,P0
Finish Time	12-4=8
Waiting Time	0
Turn Around Time	18+4=22

Step 2.7: PO continue to execute for next 4 ms.

System Time	22
Process scheduling	P0,Pl,P2,P0,P0,P0,P0
Finish Time	8-4=4
Waiting Time	0
Turn Around Time	22+4=26

Step 2.8: PO continue to executed for next 4 ms.

System Time	26
Process scheduling	P0,Pl,P2,P0,P0,P0,P0, P0
Finish Time	4-4=0
Waiting Time	0
Turn Around Time	26+4=30

Step 3: Calculate Average waiting Time and Average Turn Around Time.

Average Waiting time =6+4+7/3
=17/3
= 5.666667

Average turn around time=3 0+7+10/3 = 47/3
= 15.666667

Step 4 : After scheduling of all provided processes.

Process Id	Burst time	Turn Around time	Waiting Time
P0	24	30-0=30	30-24=6
Pl	3	4+3=7	7-3=4
P2	3	7+3=10	10-3=7
Average		15.666667	5.666667

Gantt chart

Process Id	Burst time	Turn Around time	Waiting Time
P0	24	30-0=30	30-24=6
Pl	3	4+3=7	7-3=4
P2	3	7+3=10	10-3=7
Average		15.666667	5.666667

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

P0	Pl	P2	P0	P0	P0	P0	P0

Example 2: Consider the following example containing three process arrive at same time having slice 1ms.

Process ID	**Burst Time**
PO	2
Pl	1
P2	6

Step 4 : After scheduling of all provided processes.

Process Id	Burst time	Turn Around time	Waiting Time
PO	2	4	2
Pl	1	2	1
P2	6	9	3
Average		5.000000	2.000000

Gantt chart

Process Id	Burst time	Turn Around time	Waiting Time
P0	24	30-0=30	30-24=6
P1	3	4+3=7	7-3=4
P2	3	7+3=10	10-3=7
Average		15.666667	5.666667

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

P0	P1	P2	P0	P0	P0	P0	P0

Example 3: Consider the following example containing three process arrive at same time . Time quanta =3.

Process ID	Burst Time
PO	7
Pl	3
P2	2
P3	10
P4	8

Step 4 : After scheduling of all provided processes.

Process ID	Burst Time	Waiting Time	Turn Around Ti
PO	7	17	24
Pl	3	3	6
P2	2	6	8
P3	10	20	30
P4	8	21	29
average		13.400000	19.400000

Gantt chart

Process ID	Burst Time	Waiting Time	Turn Around Ti
PO	7	17	24
Pl	3	3	6
P2	2	6	8
P3	10	20	30
P4	8	21	29
average		13.400000	19.400000

PO	Pl	P2	P3	P4	P0	P0	P0	P0	P0

Implementation :

```
//Name: Subrat Sahu
//Batch: B2
//PRN: 2020016400833692
//Date: 28/7/2021
//Prac-03: Round-Robin Scheduling Algorithm

import java.util. Scanner;

class P3_RR_SS
{
public static void main(String args[]){
Scanner input=new Scanner(System.in);
int i,j,k,q,sum=0;
System.out.print("Enter number of process:");
int n=input.nextInt();
int burstTime[]=new int[n];
int waitingTime[]=new int[n];
int turnAroundTime[]=new int[n];
int a[]=new int[n];
System.out.println("Enter the burst time of each process: "); for(i=0;i<n;i++){
System.out.print("enter the burst time for process-p"+(i+l)+":");
burstTime[i]=input.nextInt();

a[i]=burstTime[i];

}System.out.print("Enter time quantum: ");
```

```
q=input.nextInt();
for(i=0;i<n;i++)
waitingTime[i]=0;
int timer=0;
do{
for(i=0;i<n;i++){
if(burstTime[i]>q){
timer +=q;

burstTime[i] -=q;
for(j=0;j<n;j++){
if ((j!=i) && (burstTime[j]!=0))
waitingTime[j]+=q;

}
}
else{
timer +=burstTime[i];
for(j=0;j<n;j++){
if((j!=i) && (burstTime[j]!=0))
waitingTime[j] +=burstTime[i];
}
burstTime[i]=0;
}
}
sum=0;
for(k=0;k < n;k++)
sum +=burstTime[k];
}while(sum!=0);
for(i=0;i<n;i++)
turnAroundTime[i]=waitingTime[i]+a[i];
```

```
float total=0;
for(intm: waitingTime) {
total += m;
}
float averageWaitingTime=total/n;
total=0;
for(int m:turnAroundTime){
total +=m;
}

float averageTurnAroundTime=total/n;
System.out.println(" RR Algorithm:");
System.out.format("%20s%20s%20s%20s\n","ProcessId","BurstTime"
, "WaitingTime"," Turn AroundTime");
for( i=0;i<n;i++){
System.out.format("%20s%20d%20d%20d\n", "p"+(i+l),

a[i],waitingTime[i],turnAroundTime[i]); }


System.out.format("%40s%20f^D/o20f\n",

"Average",averageWaitingTime,averageTumAroundTime); }

}
```

Input:

```
Enter number of process:3
Enter the burst time of each process:
enter the burst time for process-p1:24
enter the burst time for process-p2:3
enter the burst time for process-p3:3
Enter time quantum: 4
```

Output:

```
RR Algorithm:
          ProcessId          BurstTime          WaitingTime
                 p1                 24                    6
                 p2                  3                    4
                 p3                  3                    7
                                Average             5.666667
```

Sample Output:

```
Enter number of process:3
Enter the burst time of each process:
enter the burst time for process-p1:24
enter the burst time for process-p2:3
enter the burst time for process-p3:3
Enter time quantum: 4
 RR Algorithm:
          ProcessId          BurstTime          WaitingTime        TurnAroundTime
                 p1                 24                    6                    30
                 p2                  3                    4                     7
                 p3                  3                    7                    10
                                Average             5.666667             15.666667
```

Input:

```
Enter number of process:3
Enter the burst time of each process:
enter the burst time for process-p1:2
enter the burst time for process-p2:1
enter the burst time for process-p3:6
Enter time quantum: 1
```

Output:

```
RR Algorithm:
          ProcessId           BurstTime         WaitingTime      TurnAroundTime
                 p1                   2                   2                   4
                 p2                   1                   1                   2
                 p3                   6                   3                   9
                                Average            2.000000            5.000000
```

Sample Output:

```
Enter number of process:3
Enter the burst time of each process:
enter the burst time for process-p1:2
enter the burst time for process-p2:1
enter the burst time for process-p3:6
Enter time quantum: 1
 RR Algorithm:
          ProcessId           BurstTime         WaitingTime      TurnAroundTime
                 p1                   2                   2                   4
                 p2                   1                   1                   2
                 p3                   6                   3                   9
                                Average            2.000000            5.000000
```

Input:

```
Enter number of process:5
Enter the burst time of each process:
enter the burst time for process-p1:7
enter the burst time for process-p2:3
enter the burst time for process-p3:2
enter the burst time for process-p4:10
enter the burst time for process-p5:8
Enter time quantum: 3
```

Output:

```
RR Algorithm:
          ProcessId            BurstTime          WaitingTime       TurnAroundTime
                 p1                    7                   17                   24
                 p2                    3                    3                    6
                 p3                    2                    6                    8
                 p4                   10                   20                   30
                 p5                    8                   21                   29
                                  Average           13.400000            19.400000
```

Sample Output:

```
Enter number of process:5
Enter the burst time of each process:
enter the burst time for process-p1:7
enter the burst time for process-p2:3
enter the burst time for process-p3:2
enter the burst time for process-p4:10
enter the burst time for process-p5:8
Enter time quantum: 3
 RR Algorithm:
          ProcessId            BurstTime          WaitingTime       TurnAroundTime
                 p1                    7                   17                   24
                 p2                    3                    3                    6
                 p3                    2                    6                    8
                 p4                   10                   20                   30
                 p5                    8                   21                   29
                                  Average           13.400000            19.400000
```