Round Robin Scheduling

Round Robin(RR) scheduling algorithm is mainly designed for time-sharing systems. This algorithm is similar to FCFS scheduling, but in Round Robin(RR) scheduling, preemption is added which enables the system to switch between processes.

- A fixed time is allotted to each process, called a **quantum**, for execution.
- Once a process is executed for the given time period that process is preempted and another process executes for the given time period.
- Context switching is used to save states of preempted processes.
- This algorithm is simple and easy to implement and the most important is thing is this algorithm is starvation-free as all processes get a fair share of CPU.
- It is important to note here that the length of time quantum is generally from 10 to 100 milliseconds in length.

Some important characteristics of the Round Robin(RR) Algorithm are as follows:

1. Round Robin Scheduling algorithm resides under the category of Preemptive Algorithms.
2. This algorithm is one of the oldest, easiest, and fairest algorithm.
3. This Algorithm is a real-time algorithm because it responds to the event within a specific time limit.
4. In this algorithm, the time slice should be the minimum that is assigned to a specific task that needs to be processed. Though it may vary for different operating systems.
5. This is a hybrid model and is clock-driven in nature.
6. This is a widely used scheduling method in the traditional operating system.

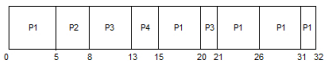
Important terms

1. **Completion Time** It is the time at which any process completes its execution.
2. **Turn Around Time** This mainly indicates the time Difference between completion time and arrival time. The Formula to calculate the same is: **Turn Around Time = Completion Time – Arrival Time**
3. **Waiting Time(W.T):** It Indicates the time Difference between turn around time and burst time. And is calculated as **Waiting Time = Turn Around Time – Burst Time**

Let us now cover an example for the same:

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

The GANTT chart for round robin scheduling will be,



The average waiting time will be, 11 ms.

In the above diagram, arrival time is not mentioned so it is taken as 0 for all processes.

Note: If arrival time is not given for any problem statement then it is taken as 0 for all processes; if it is given then the problem can be solved accordingly.

Explanation

The value of time quantum in the above example is 5.Let us now calculate the Turn around time and waiting time for the above example :

Processes	Burst Time	Turn Around Time **Turn Around Time = Completion Time – Arrival Time**	Waiting Time **Waiting Time = Turn Around Time – Burst Time**
P1	21	32-0=32	32-21=11
P2	3	8-0=8	8-3=5
P3	6	21-0=21	21-6=15

P4	2	15-0=15	15-2=13

Average waiting time is calculated by adding the waiting time of all processes and then dividing them by no.of processes.

average waiting time = waiting time of all processes/ no.of processes

average waiting time=11+5+15+13/4 = 44/4= **11ms**

C++ Implementation For RR Scheduling

```
// Program implementation in C++ for Round Robin scheduling
#include<iostream>
using namespace std;

//The Function to find the waiting time for all processes
void fWaitingTime(int processes[], int n,
            int bt[], int wt[], int quantum)
{
    // Let us Make a copy of burst times bt[] to store remaining burst times

    int rem_bt[n];
    for (int i = 0 ; i < n ; i++)
        rem_bt[i] = bt[i];

    int t = 0; // for Current time

    // Let us keep traverse the processes in the round robin manner until all of them are not c

    while (1)
```

Output

The output of the above code is as follows:

```
Processes  Burst time  Waiting time  Turn around time
 1             21       31             52
 2             13       32             45
 3             6        16             22
 4             12       30             42
Average waiting time = 27.25
Average turn around time = 40.25
```

Advantages of Round Robin Scheduling Algorithm

Some advantages of the Round Robin scheduling algorithm are as follows:

- While performing this scheduling algorithm, a particular time quantum is allocated to different jobs.
- In terms of average response time, this algorithm gives the best performance.
- With the help of this algorithm, all the jobs get a fair allocation of CPU.
- In this algorithm, there are no issues of starvation or convoy effect.
- This algorithm deals with all processes without any priority.
- This algorithm is cyclic in nature.
- In this, the newly created process is added to the end of the ready queue.
- Also, in this, a round-robin scheduler generally employs time-sharing which means providing each job a time slot or quantum.
- In this scheduling algorithm, each process gets a chance to reschedule after a particular quantum time.

Disadvantages of Round Robin Scheduling Algorithm

Some disadvantages of the Round Robin scheduling algorithm are as follows:

- This algorithm spends more time on context switches.
- For small quantum, it is time-consuming scheduling.
- This algorithm offers a larger waiting time and response time.
- In this, there is low throughput.
- If time quantum is less for scheduling then its Gantt chart seems to be too big.

Some Points to Remember

1.Decreasing value of Time quantum

With the decreasing value of time quantum

- The number of context switches increases.
- The Response Time decreases
- Chances of starvation decrease in this case

- Chances of starvation decrease in this case.

For the **smaller value of time quantum,** it becomes better in terms of **response time.**

2.Increasing value of Time quantum

With the increasing value of time quantum

- The number of context switch decreases
- The Response Time increases
- Chances of starvation increases in this case.

For the higher value of time quantum, it becomes better in terms of the **number of the context switches.**

3. If the value of **time quantum is increasing** then Round Robin Scheduling tends to **become FCFS Scheduling.**

4.In this case, when the value of time quantum **tends to infinity** then the Round Robin Scheduling **becomes FCFS Scheduling.**

5. Thus the performance of Round Robin scheduling mainly depends on the **value of the time quantum.**

6.And the value of the **time quantum** should be such that it is neither **too big nor too small.**

← Prev

Next →