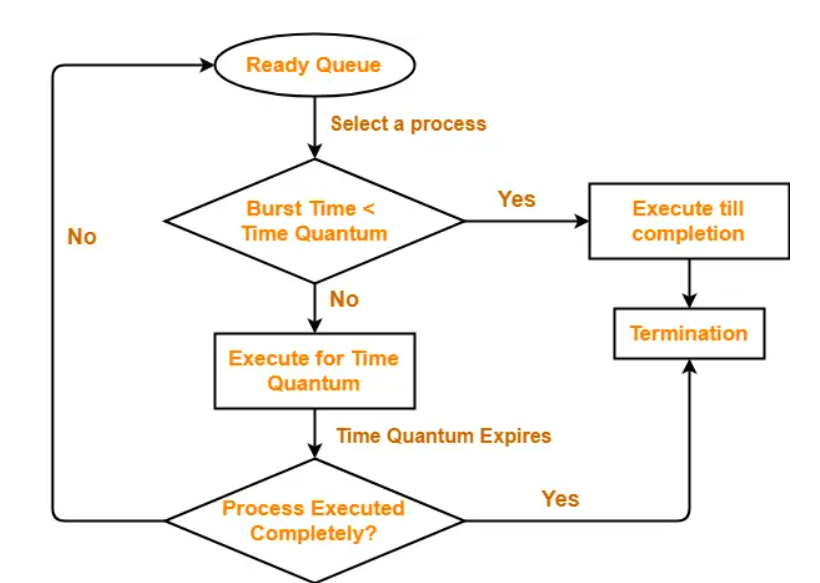
**ROUND-ROBIN SCHEDULING**

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. It is always preemptive in nature.

In **Round Robin**scheduling, in which every process will get an equal amount of time or time quantum of the CPU to execute the process. After the Quantum of time passes, the current running process is preempted and the next process in ready state gets executed for next quantum of time. CPU Scheduler goes around the ready queue and allocates CPU to each process for the interval upto 1 time quantum. Ready Queue is like a First In First Out Structure, where new processes are at last of the ready queue.

Now, while a process is in that quantum executing, one of the two things will happen

1. If the process has a CPU burst of less than quantum time, then as the process gets finished and itself releases the CPU voluntarily. Then the next process in the ready queue occupies the CPU.
2. If the CPU burst is larger than the quantum, the timer will go off and will cause an interrupt to the operating system. A context switch will be executed and the process is preempted and that process will be put at the tail of ready queue. After this, the next process in the ready queue comes in CPU quantum.

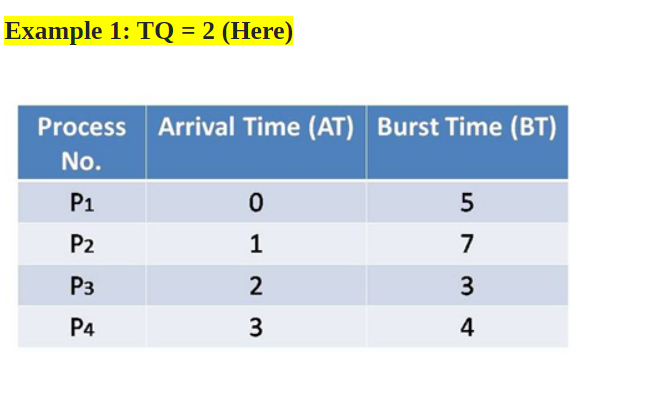
****

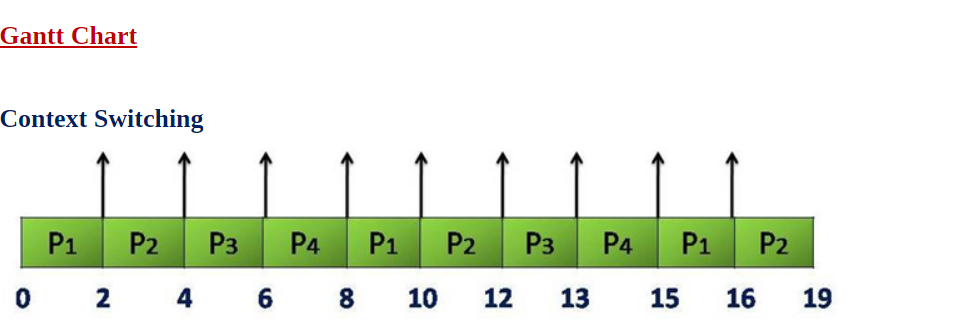
**Example:** Consider the set of 4 processes whose arrival time and burst time are given below-

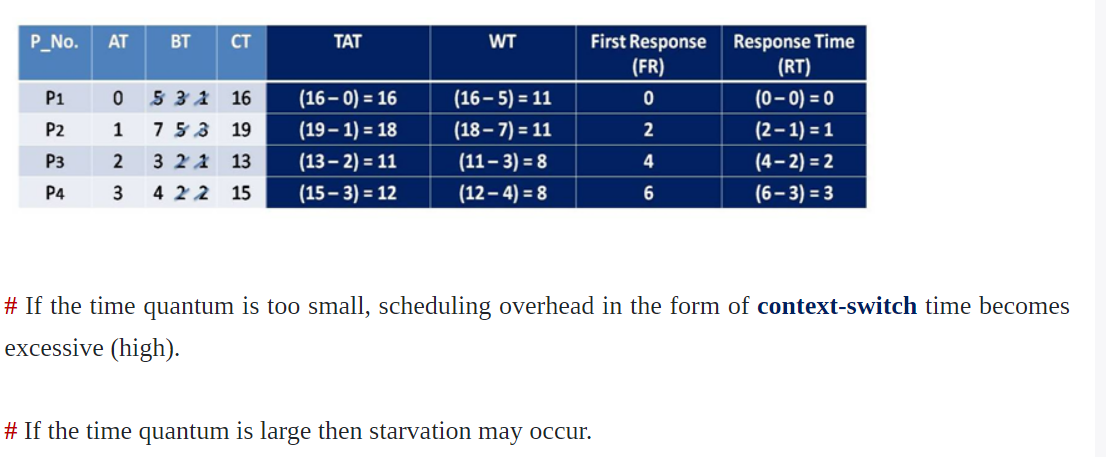
|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 0 | 3 |
| P2 | 0 | 6 |
| P3 | 0 | 4 |
| P4 | 0 | 5 |

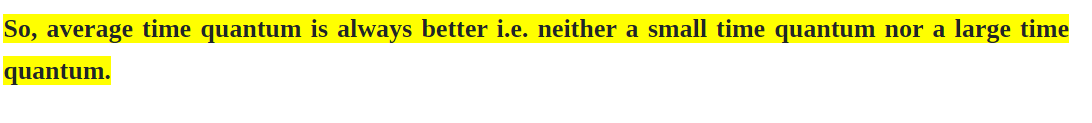
If the CPU scheduling policy is Round Robin with time quantum = 2 unit, calculate the average waiting time and average turnaround time.

**Solution-**

****

****

****

****

**Example:** Consider the set of 6 processes whose arrival time and burst time are given below-

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 0 | 6 |
| P2 | 1 | 5 |
| P3 | 2 | 4 |
| P4 | 3 | 3 |
| P5 | 4 | 2 |
| P6 | 5 | 4 |

If the CPU scheduling policy is Round Robin with time quantum = 2, calculate the average waiting time and average turnaround time.

## ****Solution-****

**Example:** Consider the set of 6 processes whose arrival time and burst time are given below-

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 5 | 6 |
| P2 | 8 | 7 |
| P3 | 3 | 8 |
| P4 | 6 | 3 |
| P5 | 2 | 2 |
| P6 | 4 | 4 |

If the CPU scheduling policy is Round Robin with time quantum = 3, calculate the average waiting time and average turnaround time.

## ****Solution-****

**Example:** Consider the set of 7 processes whose arrival time and burst time are given below-

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 0 | 12 |
| P2 | 0 | 5 |
| P3 | 3 | 9 |
| P4 | 5 | 6 |
| P5 | 2 | 8 |
| P6 | 4 | 2 |
| P7 | 1 | 7 |

If the CPU scheduling policy is Round Robin with time quantum = 3, calculate the average waiting time and average turnaround time.

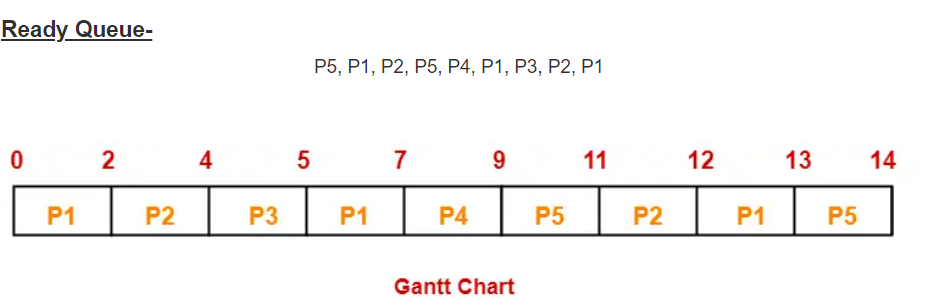
## ****Solution-****

**Example:**Consider the set of 5 processes whose arrival time and burst time are given below-

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 0 | 5 |
| P2 | 1 | 3 |
| P3 | 2 | 1 |
| P4 | 3 | 2 |
| P5 | 4 | 3 |

If the CPU scheduling policy is Round Robin with time quantum = 2 unit, calculate the average waiting time and average turn around time.

## ****Solution-****

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Process Id** | **Exit time** | **Turn Around time** | **Waiting time** |
| P1 | 13 | 13 – 0 = 13 | 13 – 5 = 8 |
| P2 | 12 | 12 – 1 = 11 | 11 – 3 = 8 |
| P3 | 5 | 5 – 2 = 3 | 3 – 1 = 2 |
| P4 | 9 | 9 – 3 = 6 | 6 – 2 = 4 |
| P5 | 14 | 14 – 4 = 10 | 10 – 3 = 7 |

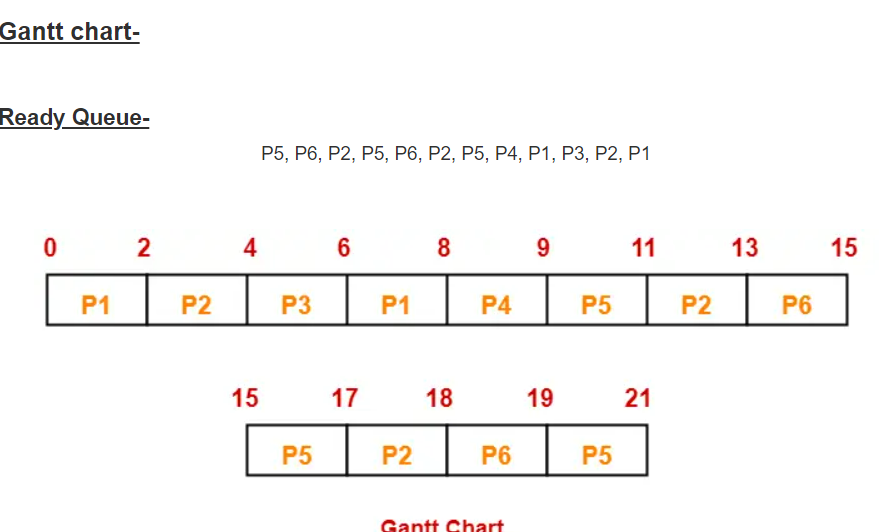
* Average Turn Around time = (13 + 11 + 3 + 6 + 10) / 5 = 43 / 5 = 8.6 unit
* Average waiting time = (8 + 8 + 2 + 4 + 7) / 5 = 29 / 5 = 5.8 unit

**Example:** Consider the set of 6 processes whose arrival time and burst time are given below-

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 0 | 4 |
| P2 | 1 | 5 |
| P3 | 2 | 2 |
| P4 | 3 | 1 |
| P5 | 4 | 6 |
| P6 | 6 | 3 |

If the CPU scheduling policy is Round Robin with time quantum = 2, calculate the average waiting time and average turn around time.

## ****Solution-****

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Process Id** | **Exit time** | **Turn Around time** | **Waiting time** |
| P1 | 8 | 8 – 0 = 8 | 8 – 4 = 4 |
| P2 | 18 | 18 – 1 = 17 | 17 – 5 = 12 |
| P3 | 6 | 6 – 2 = 4 | 4 – 2 = 2 |
| P4 | 9 | 9 – 3 = 6 | 6 – 1 = 5 |
| P5 | 21 | 21 – 4 = 17 | 17 – 6 = 11 |
| P6 | 19 | 19 – 6 = 13 | 13 – 3 = 10 |

Now,

* Average Turn Around time = (8 + 17 + 4 + 6 + 17 + 13) / 6 = 65 / 6 = 10.84 unit
* Average waiting time = (4 + 12 + 2 + 5 + 11 + 10) / 6 = 44 / 6 = 7.33 unit

**Advantages:**

The Advantages of Round Robin CPU Scheduling are:

1) In round-robin algorithm, there is no starvation problem. As Each process gets equal priority and fair allocation of CPU.

2) It can be practically implemented in the system because Burst time is not required to be known in advance.

**Disadvantages:**

The Disadvantages of Round Robin CPU Scheduling are:

1)The average waiting time and turnaround time are more.

2) Its performance heavily depends on time quantum.

3) There is an overhead of context switching as mentioned earlier, too small of quantum time causes overhead and slower execution of process. So, time quantum must be large with respect to Context Switch time.

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.

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.**