PRACTICAL 5

**Aim**: Program to demonstrate the Round Robin algorithm.

**Prerequisite:**

1. Understanding of fundamental programming constructs in Java/C/C++/python

**Outcome:** After successful completion of this experiment students will be able to

1. Understand the basic concepts of Process scheduling
2. Understand the algorithm of Round Robin Algorithm.
3. Construction of Gantt Chart for finding out waiting time and turnaround time of a process
4. Understanding how RR outperforms the other scheduling algorithms provided that the time quantum is appropriately chosen.

**Theory:** One of the oldest, simplest, fairest and most widely used algorithm is Round Robin (RR).

In the round robin scheduling, processes are dispatched in a FIFO manner but are given a limited amount of CPU time called a time-slice or a quantum. If a process does not complete before its CPU-time expires, the CPU is preempted and given to the next process waiting in a queue. The preempted process is then placed at the back of the ready list.

Round Robin Scheduling is preemptive (at the end of time-slice) therefore it is effective in time-sharing environments in which the system needs to guarantee reasonable response times for interactive users.

The only interesting issue with round robin scheme is the length of the quantum. Setting the quantum too short causes too many context switches and lower the CPU efficiency. On the other hand, setting the quantum too long may cause poor response time and appoximates FCFS. In any event, the average waiting time under round robin scheduling is often quite long.

1. **Completion Time:** Time at which process completes its execution.
2. **Turn Around Time:** Time Difference between completion time and arrival time.

Turn Around Time = Completion Time – Arrival Time

1. **Waiting Time(W.T):** Time Difference between turn around time and burst time.   
    Waiting Time = Turn Around Time – Burst Time

**Example: FIFO**

Time Quantum = 4

                        Process    Burst Time/Execution time

*P1*24

*P2                     3*

*P3                     3*

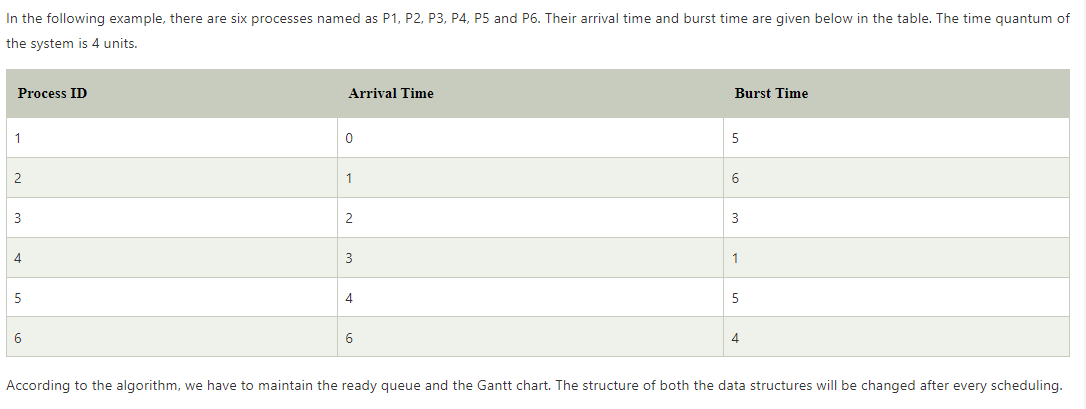
The Gantt chart is:

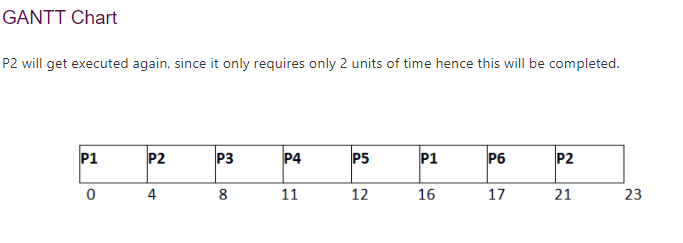
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P1 | P1 | P1 | P1 | P1 |

0          4               7              10            14            18             22          26            30

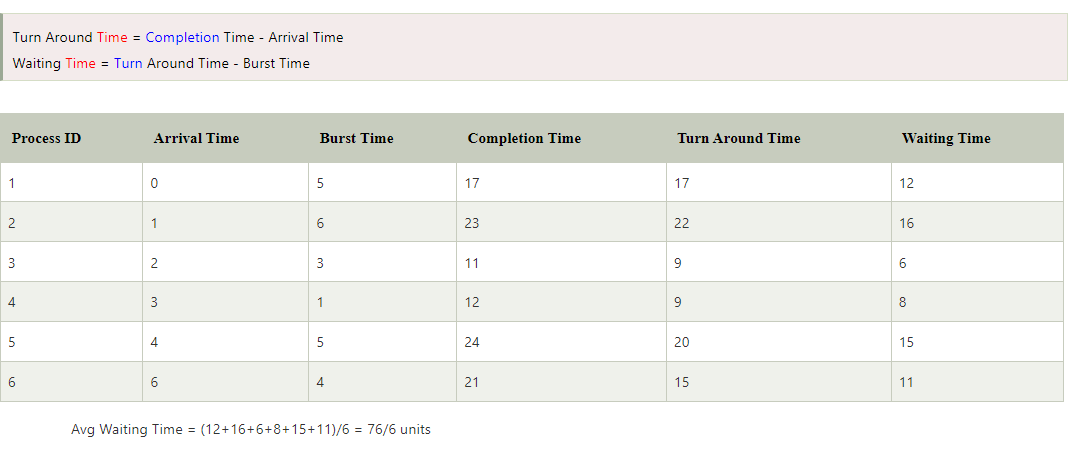
Average waiting time =    [(30-24)+4+7]/3  = 17/3 =5.66

Example 2:

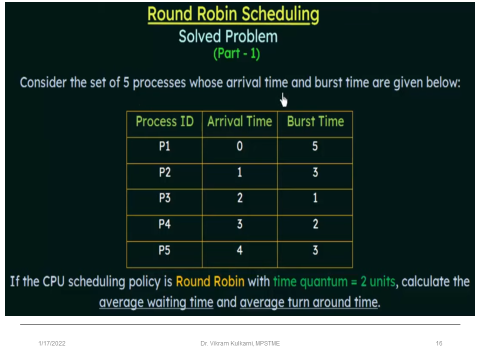




Ref: <https://www.javatpoint.com/os-round-robin-scheduling-example>



Example-3



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Process ID** | **Completion Time** | **Turn around time**  **TAT= CT-AT** | **Waiting time**  **WT= TAT-BT** | **AVG TAT= (13+11+3+6+10)/5 = 8.6** |
| P1 | 13 | 13-0=13 | 13-5=8 |  |
| P2 | 12 | 12-1=11 | 11-3=8 | AVG WT= (8+8+2+4+7)/5=5.8 |
| P3 | 5 | 5-2=3 | 3-1=2 |
| P4 | 9 | 9-3=6 | 6-2=4 |  |
| P5 | 14 | 14-4=10 | 10-3=7 |  |
|  |  |  |  |  |
|  |  |  |  |  |

Conclusion: