Scheduling Algorithms

Round Robin Scheduling

The round-robin (RR) scheduling algorithm is designed especially for time sharing systems. It is similar to FCFS scheduling, but preemption is added to switch between processes. A small unit of time, called a time quantum (or time slice), is defined. The ready queue is treated as a circular queue.

Example

Process	Burst Time
P1	24
P2	3
P3	3

Time quantum = 4 ms

Gantt chart

p1	p2	p3	p1	p1	p1	p1	p1

0 4 7 10 14 18 22 26 30

The average waiting time is 17/3 = 5.66 milliseconds.

Waiting time for P1 = 26 – 20 = 6

P2 = 4

P3 = 7 (6+4+7 / 3 = 5.66 ms)

The performance of the RR algorithm depends heavily on the size of the time–quantum. If the time-quantum is very large(infinite) then RR policy is the same as FCFS policy. If time quantum is very small, RR approach is called processor sharing and appears to the users as though each of n processes has its own processor running at 1/n the speed of a real processor.

Example Problem:

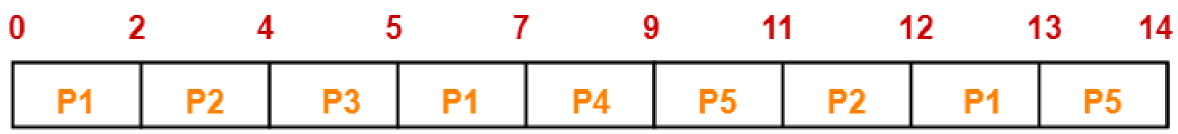
Consider the set of 5 processes whose arrival time and burst time are given below-

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 turnaround time.

Solution:

Ready Queue-
P5, P1, P2, P5, P4, P1, P3, P2, P1



Gantt Chart

Now, we know-

- Turn Around time = Exit time – Arrival time
- Waiting time = Turnaround time – Burst time

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

Now,

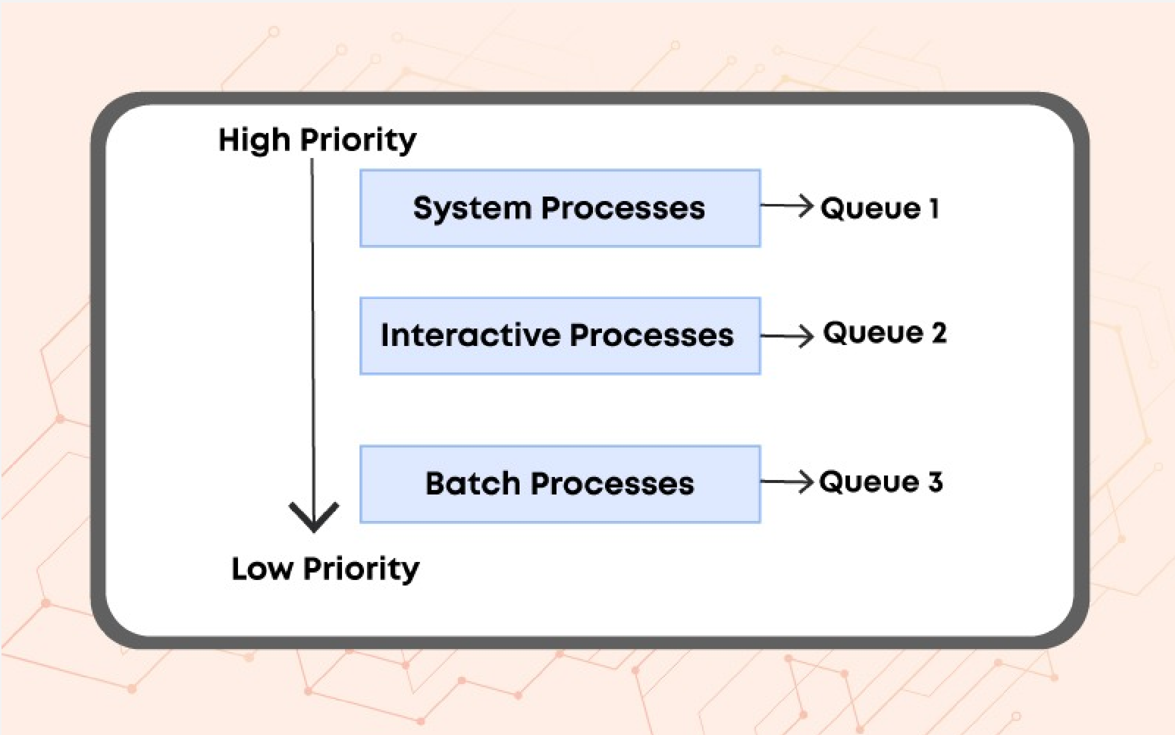
- Average Turnaround time = (13 + 11 + 3 + 6 + 10) / 5 = 43 / 5 = 8.6 unit
- Average waiting time = (8 + 8 + 2 + 4 + 7) / 5 = 29 / 5 = 5.8 unit

Multilevel Queue Scheduling

A multilevel queue-scheduling algorithm partitions the ready queue into several separate queues . The processes are completely assigned to one queue, generally based on some unique property of the process, such as memory size, process priority, or process type.
There must be scheduling between the queues, which is commonly implemented as a fixed-priority preemptive scheduling.
For example the foreground queue may have absolute priority over the background queue.

Example : Three queues:

1. System processes
2. Interactive processes
3. Batch processes



Example Problem:

Consider a system which has a CPU bound process, which requires the burst time of 40 seconds.The multilevel Feedback Queue scheduling algorithm is used and the queue time quantum ‘2’ seconds and in each level it is incremented by ‘5’ seconds. Then how many times the process will be interrupted and on which queue the process will terminate the execution?

Solution :

Process P needs 40 Seconds for total execution.

At Queue 1 it is executed for 2 seconds and then interrupted and shifted to queue 2.

At Queue 2 it is executed for 7 seconds and then interrupted and shifted to queue 3.

At Queue 3 it is executed for 12 seconds and then interrupted and shifted to queue 4.

At Queue 4 it is executed for 17 seconds and then interrupted and shifted to queue 5.

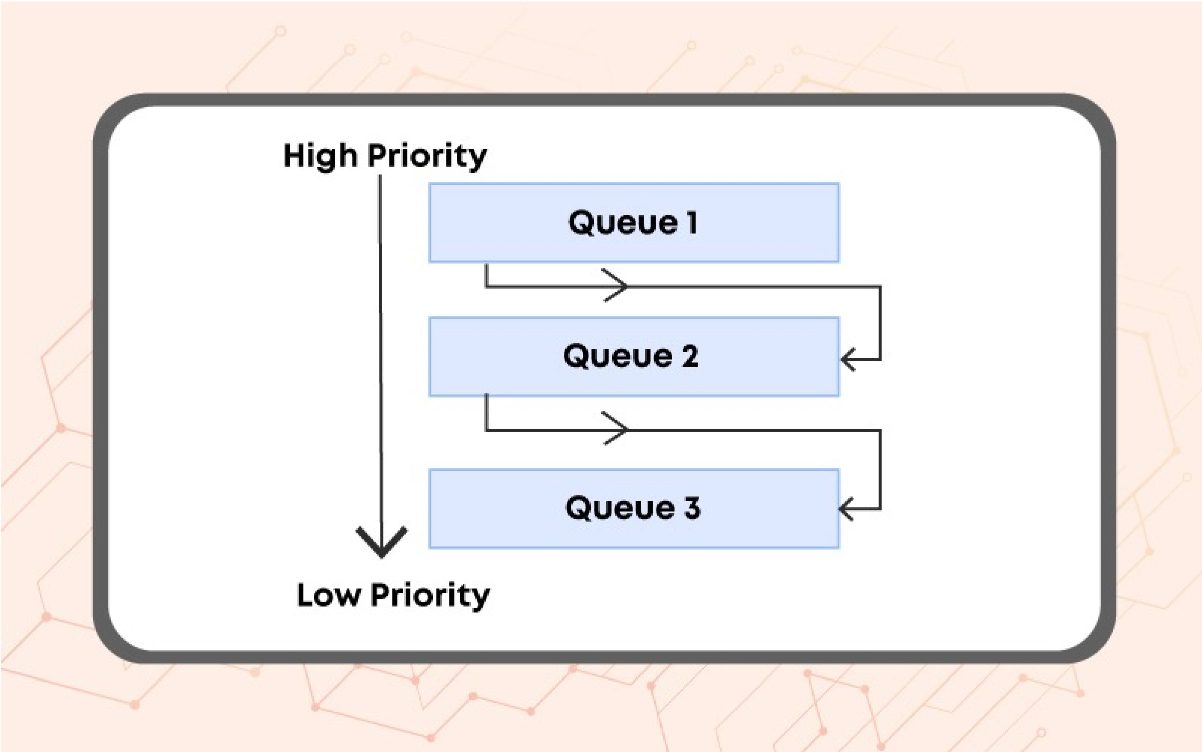
At Queue 5 it executes for 2 seconds and then it completes.

Hence the process is interrupted 4 times and completes on queue 5.

Multilevel Feedback Queue Scheduling

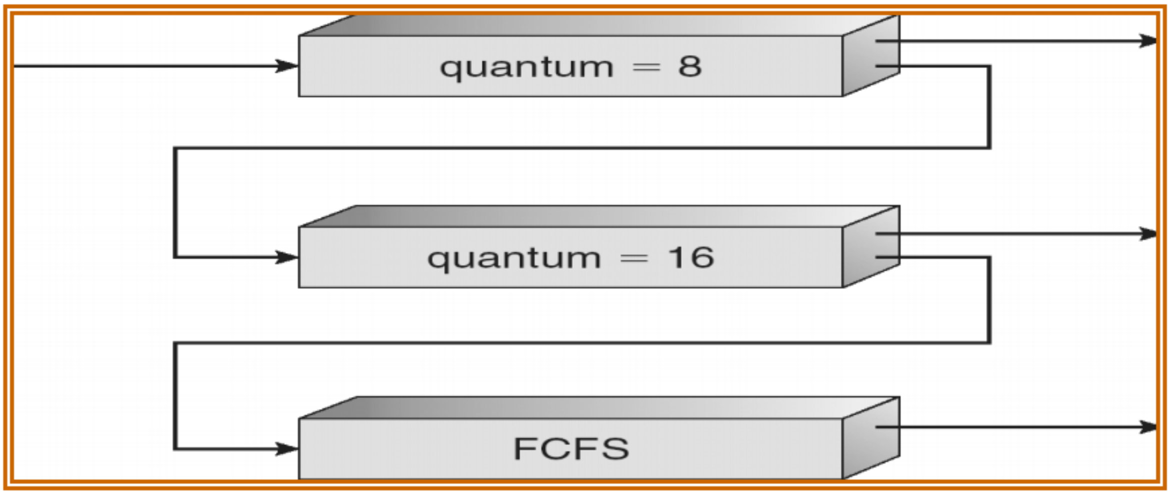
In the multilevel queue-scheduling algorithm, the processes are permanently assigned to the system entry queue. Processes do not move between queues. This setup has the advantage of lower scheduling overhead, but the lack of consistency.
Multilevel Feedback queue scheduling, however, allows the process to move between queues. The idea is to split processes with different CPU-burst features. If the process consumes too much CPU time, it will be moved to a much lower line. Similarly, the longest waiting process in the lowest line can be delivered to the most important line. This form of aging prevents starvation.

- Allows processes to move between queues
- Inter-queue scheduling: preemptive priority scheduling
- A process waiting too long in a low-priority queue may be moved to a high-priority queue.



Example:

Consider a multilevel feedback queue scheduler with three queues, numbered from 0 to 2 The scheduler first executes all processes in queue 0. Only when queue 0 is empty will it execute processes in queue 1. Similarly, processes in queue 2 will be executed only if queues 0 and 1 are empty. A process that arrives for queue 1 will preempt a process in queue 2. A process that arrives for queue 0 will, in turn, preempt a process in queue 1.



Example Problem:

Consider a system which has a CPU bound process,which require the burst time of 40 seconds.The Multilevel Feedback Queue scheduling algorithm is used and the queue time quantum '2' seconds and in each level it is incremented by '5' seconds.Then how many times the process will be interrupted and on which queue the process will terminate the execution?

(a) 5,4 (b) 4,5 (c) 3,4 (d) 4,3

Solution:

Process P needs 40 Seconds for total execution.
Multilevel Feedback queue:
Queue 1 = 2
Queue2 = 7 (2+5)
Queue3 = 12 (7+5)
Queue4 = 17 (12+5)
Queue5 = 2 (left time)
hence you can see process is interrupted 4 times, and on 5th queue process completed its execution.
Hence, (b) is correct option