**Difference between Multilevel Queue (MLQ) and Multi Level Feedback Queue (MLFQ) CPU scheduling algorithms**

In multi programming environment, it often happens that more than one processes compete for CPU resources at the same time. If only one CPU is available choice has to be made between processes to run next. Part of the Operating System responsible for making choice of process is called Scheduler and the algorithm it used is called Scheduling Algorithm.

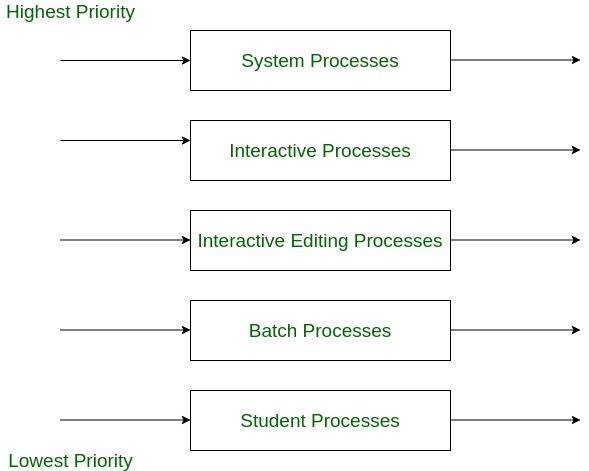
The objective of multi programming is to maximize CPU utilization. Criteria like Turnaround time, Response time, Waiting time, Throughput are suggested on basis of which scheduling algorithms are judged. There are many CPU scheduling algorithms two of which are-

1. [Multilevel Queue Scheduling](https://www.geeksforgeeks.org/multilevel-queue-mlq-cpu-scheduling/)
2. [Multilevel Feedback Queue Scheduling](https://www.geeksforgeeks.org/multilevel-feedback-queue-scheduling-mlfq-cpu-scheduling/)

**Difference between Multilevel Queue (MLQ) and Multi Level Feedback Queue (MLFQ) CPU scheduling algorithms :**

| **Multilevel queue scheduling (MLQ)** | **Multilevel feedback queue scheduling (MLFQ)** |
| --- | --- |
| It is queue scheduling algorithm in which ready queue is partitioned into several smaller queues and processes are assigned permanently into these queues. The processes are divided on basis of their intrinsic characteristics such as memory size, priority etc. | In this algorithm, ready queue is partitioned into smaller queues on basis of CPU burst characteristics. The processes are not permanently allocated to one queue and are allowed to move between queues. |
| In this algorithm queue are classified into two groups, first containing background processes and second containing foreground processes.  80% CPU time is given to foreground queue using Round Robin Algorithm and 20% time is given to background processes using First Come First Serve Algorithm. | Here, queues are classified as higher priority queue and lower priority queues. If process takes longer time in execution it is moved to lower priority queue.  Thus, this algorithm leaves I/O bound and interactive processes in higher priority queue. |
| The priority is fixed in this algorithm. When all processes in one queue get executed completely then only processes in other queue are executed.  Thus, starvation can occur. | The priority for process is dynamic as process is allowed to move between queue. A process taking longer time in lower priority queue can be shifted to higher priority queue and vice versa.  Thus, it prevents starvation. |
| Since, processes do not move between queues, it has low scheduling overhead and is inflexible. | Since, processes are allowed to move between queues, it has high scheduling overhead and is flexible. |

**1. Example for Multilevel Queue Scheduling (MLQ) :**   
A multilevel Queue with five queue is listed below according to order of priority. 

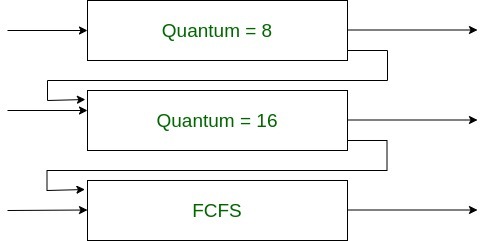


1. System Process Queue
2. Interactive Processes Queue
3. Interactive Editing Processes Queue
4. Batch Processes Queue
5. Student Processes Queue

Here, all queues have their own scheduling algorithm, and process is chosen with highest priority. Then it is executed preemptive or non-preemptively. No process in lower priority queue can get executed until higher process queue are all empty.

For example, if batch process queue is running and interactive process comes in ready state batch process is preempted and interactive process is allowed to execute.

**2. Example for Multilevel Feedback Queue Scheduling (MLFQ) :**   
Now, let us consider multilevel feedback queue with three queues. 



1. A Round Robin queue with time quantum of 8 milliseconds, say Q1.
2. A Round Robin queue with time quantum of 16 milliseconds, say Q2.
3. A First Come First Serve queue, say Q3.

Now, when the process enters Q1 it is allowed to execute and if it does not complete in 8 milliseconds it is shifted to Q2 and receives 16 milliseconds. Again it is preempted to Q3 if it does not complete in 16 seconds. In this manner, scheduling is carried on in this scheme.