**MULTILEVEL QUEUE SCHEDULING**

Another class of scheduling algorithms has been created for situations in which processes are easily classified into different groups.

**For example,** A common division is made between foreground (or interactive) processes and background (or batch) processes. These two types of processes have different response-time requirements, and so might have different scheduling needs. In addition, foreground processes may have priority over background processes.

A multi-level queue scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority, or process type. Each queue has its own scheduling algorithm.

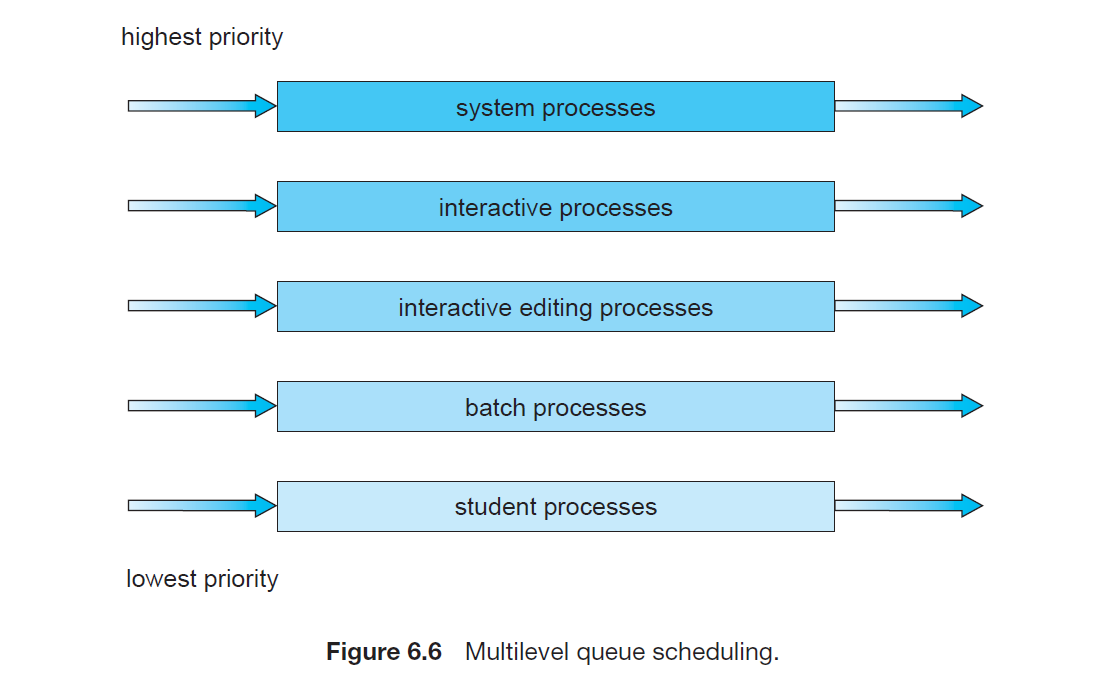
**For example,** separate queues might be used for foreground and background processes. The foreground queue might be scheduled by the Round Robin algorithm, while the background queue is scheduled by an FCFS algorithm.

In addition, there must be scheduling among the queues, which is commonly implemented as fixed-priority preemptive scheduling.

**For example,** The foreground queue may have absolute priority over the background queue.

Let us consider an example of a multilevel queue-scheduling algorithm with five queues listed below in order of priority:

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



Each queue has absolute priority over lower-priority queues. No process in the batch queue, for example, could run unless the queues for system processes, interactive processes, and interactive editing processes were all empty. If an interactive editing process entered the ready queue while a batch process was running, the batch process will be preempted.

Another possibility is to time-slice among the queues. Here, each queue gets a certain portion of the CPU time, which it can then schedule among its various processes. For instance, in the foreground–background queue example, the foreground queue can be given 80 percent of the CPU time for RR scheduling among its processes, while the background queue receives 20 percent of the CPU to give to its processes on an FCFS basis.

**Advantages:**

1)Different type of process has different scheduling algorithm, as per requirement.

2) The scheduling overhead is very low. Since processes are permanently assigned to their respective queues, the overhead of scheduling is low, as the scheduler only needs to select the appropriate queue for execution.

**Disadvantages:**

1) The main disadvantage of Multilevel Queue Scheduling is the problem of starvation for lower-level processes.

2) This scheduling is rigid and difficult to implement.