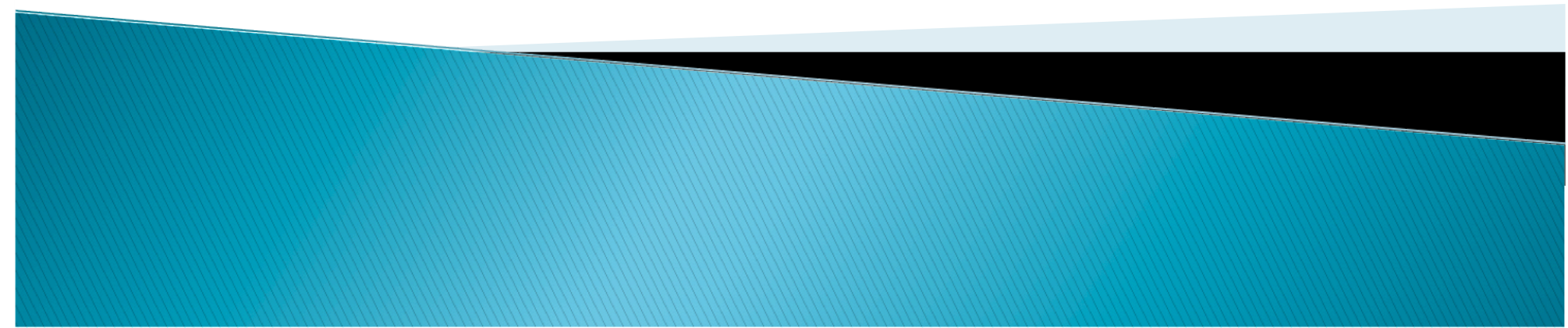


Chapter 5: CPU Scheduling



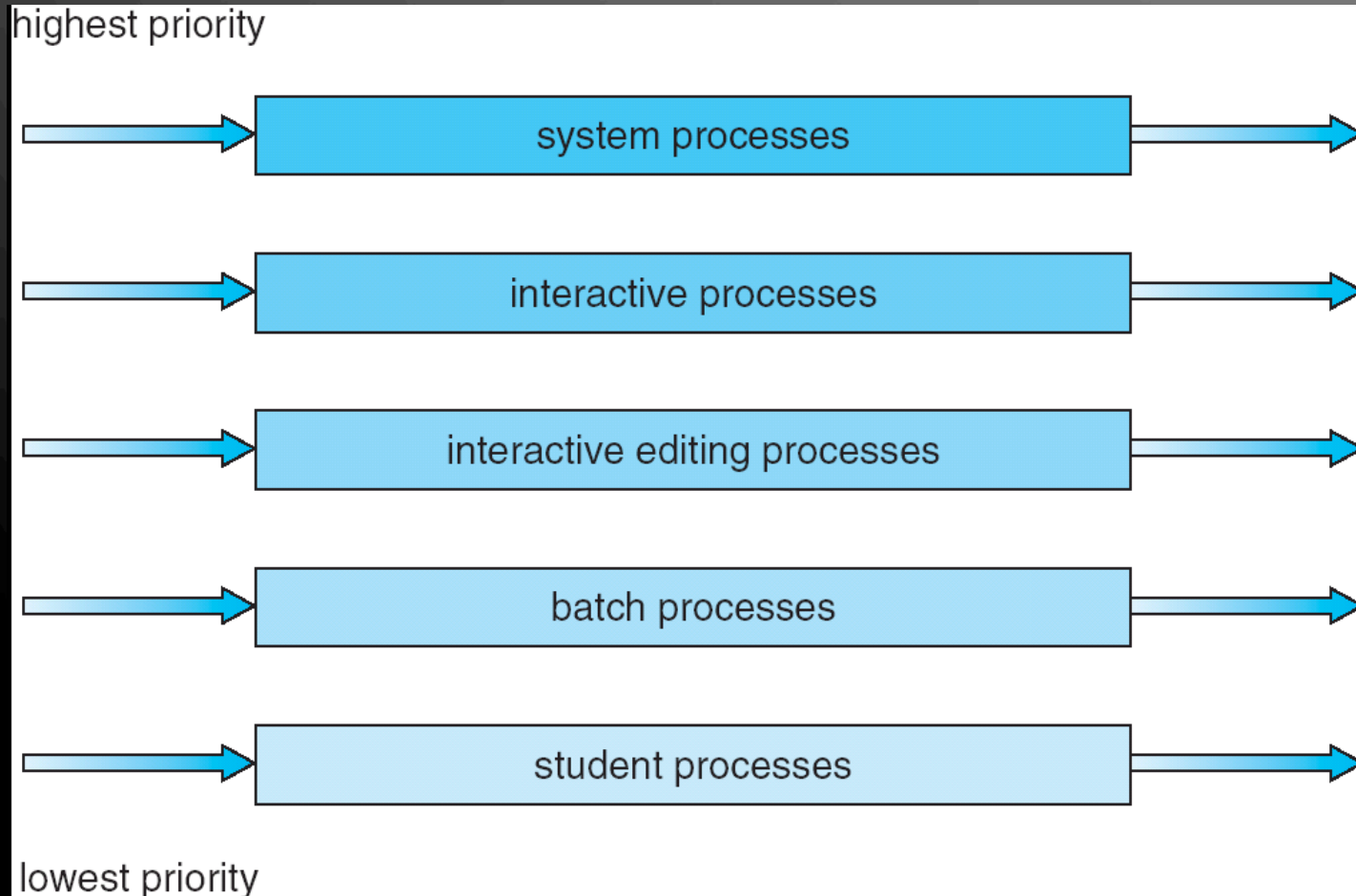
Multilevel Queue Scheduling

- Multilevel queue is used when processes can be **classified into groups** based on some characteristic like response time, priority etc like
 - foreground (interactive)
 - background (batch)
- Ready queue is partitioned into separate queues and each queue has its own scheduling algorithm
 - foreground – RR
 - background – FCFS
- Ø Processes are permanently assigned to one queue, generally based on some property of the process, such as process priority, or process type.

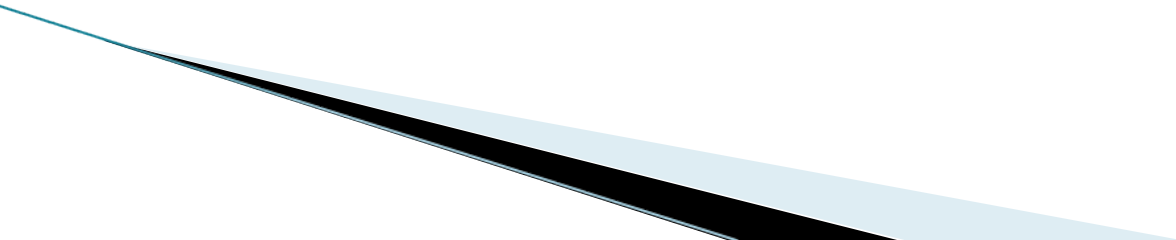
Multilevel Queue (cntd)

- Scheduling must be done between the queues. Two options
 - **Fixed priority preemptive scheduling**; Each queue has absolute priority over lower-priority queues. Possibility of *starvation*.
 - **Time slice** – each queue gets a certain amount of CPU time which it can schedule amongst its processes; i.e., 80% to foreground in RR
20% to background in FCFS
 - Since processes cannot change their foreground or background nature (move from one queue to another), this scheme is inflexible.

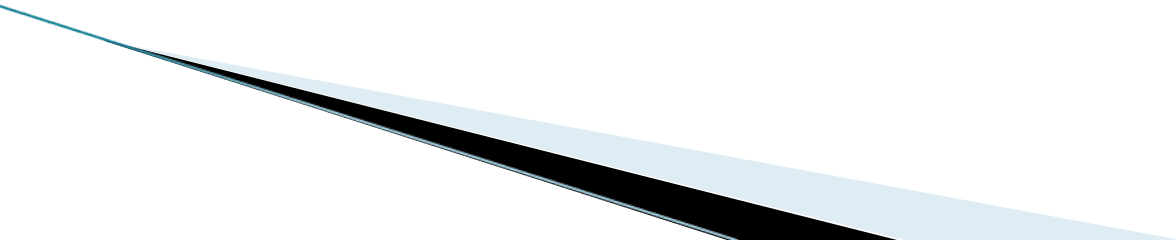
Multilevel Queue Scheduling



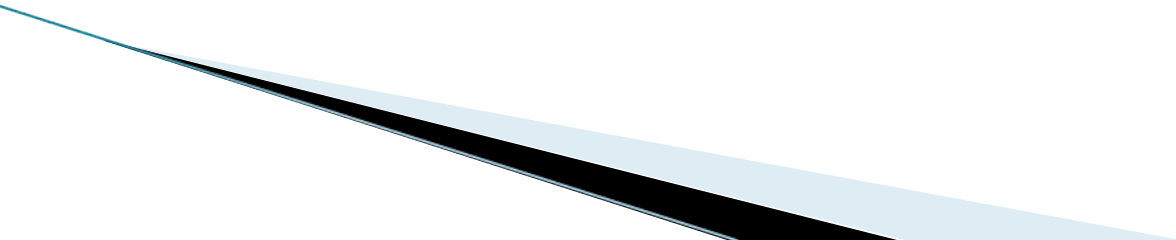
Multilevel Feedback Queue Scheduling

- ❑ Multilevel feedback-queue scheduling algorithm allows a process to **move between queues**.
 - ❑ Processes are scheduled according to the characteristics of their CPU bursts.
 - ❑ If a process uses too much CPU time, it will be moved to a lower-priority queue. This scheme leaves I/O-bound and interactive processes in the higher-priority queues.
 - ❑ In addition, a process that waits too long in a lower-priority queue may be moved to a higher-priority queue. This form of **aging** prevents starvation.
- 

Example of Multilevel Feedback Queue

- Three queues:
 - Q0 – RR with time quantum 8 milliseconds
 - Q1 – RR time quantum 16 milliseconds
 - Q2 – FCFS
 - **Scheduling:** 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 only be executed if queues 0 and 1 are empty.
 - A process entering the ready queue is put in queue 0 and given a time quantum of 8 milliseconds.
 - If it does not finish within this time, it is moved to the tail of queue 1. If queue 0 is empty, the process at the head of queue 1 is given a quantum of 16 milliseconds.
 - If it does not complete, it is preempted and is put into queue 2. Processes in queue 2 are run on an FCFS basis but are run only when queues 0 and 1 are empty.
- 

Multilevel Feedback Queue (cntd)

- Hence highest priority is given to any process with a CPU burst of 8 milliseconds or less as such a process will quickly get the CPU, finish its CPU burst, and go off to its next I/O burst.
 - Processes that need more than 8 but less than 24 milliseconds are also served quickly, although with lower priority than shorter processes.
 - Long processes automatically sink to queue 2 and are served in FCFS order with any CPU cycles left over from queues 0 and 1.
- 

Multilevel Feedback Queues

