

# Operating Systems

## | Multilevel Queue Scheduling

Prof. Dr. Mostafa Elhosseini  
Professor of Smart Systems Engineering

<https://elhosseini.com/>

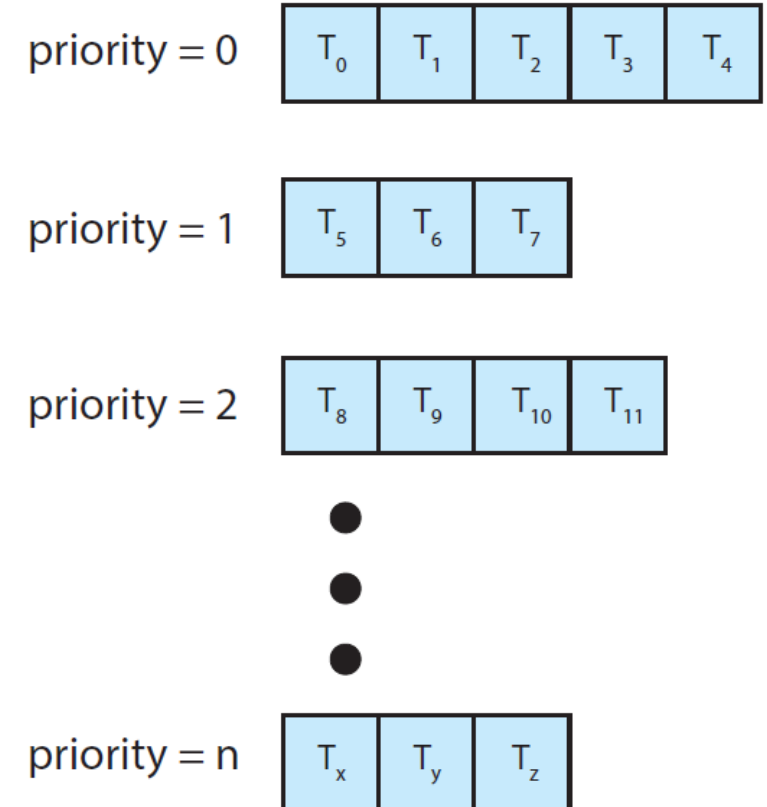
# | Agenda

- Multilevel Queue Scheduling (MLQ)
- Multilevel Feedback Queue (MLFQ)



# Multilevel Queue (MLQ) Scheduling

- In Multilevel Queue Scheduling, processes are divided into groups (queues) based on their type or priority.
- Each queue has its own scheduling algorithm 🧠.
- Queues are assigned different priorities, so one queue may always be served before another.



Separate queues for each priority.



# Multilevel Queue (MLQ) Scheduling

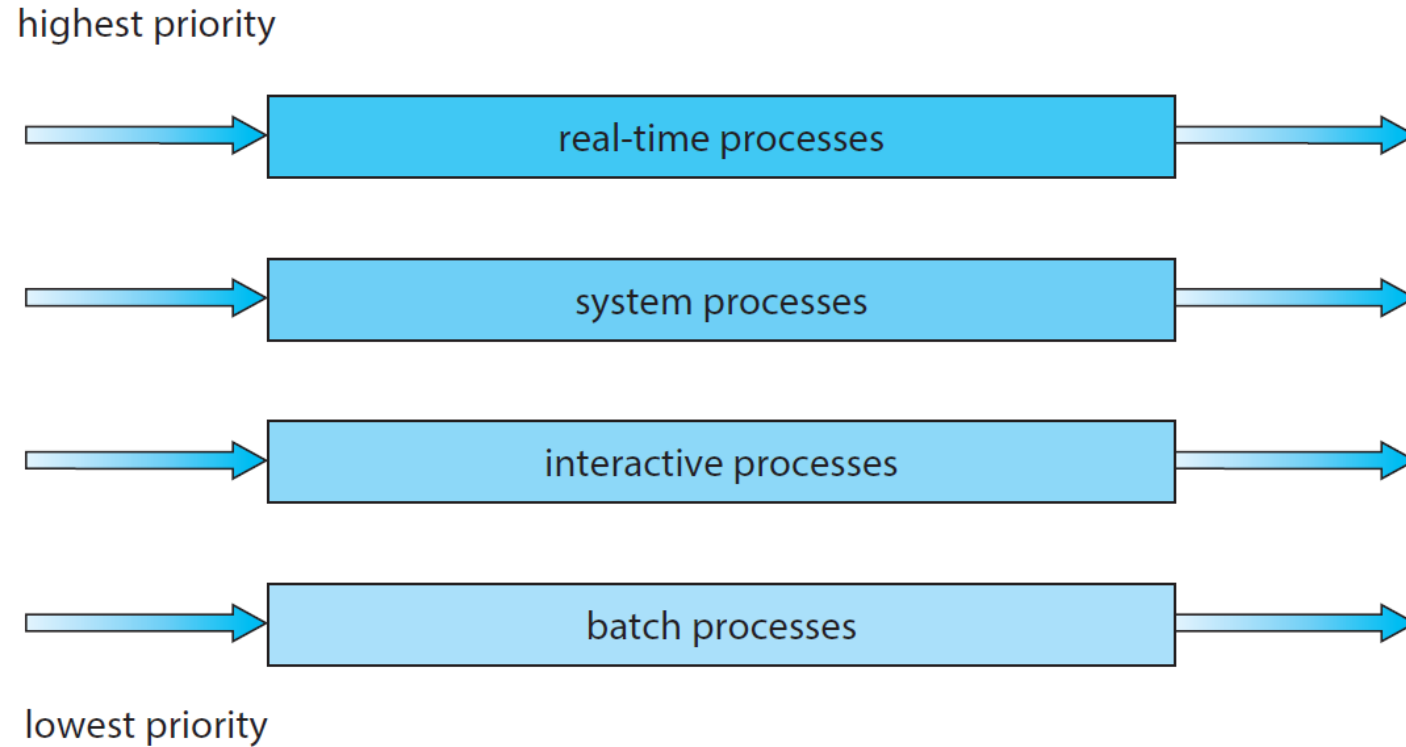


## How It Works

- 1 The ready queue is split into multiple separate queues:
  -  Foreground (interactive)
  -  Background (batch)
  -  System processes, etc.
- 2 Each queue uses its own policy:
  - Foreground → Round Robin 
  - Background → FCFS 
- 3 The CPU scheduler decides which queue to serve next, often giving higher priority to interactive queues.



# Multilevel Queue (MLQ) Scheduling



Multilevel queue scheduling.



# Multilevel Queue (MLQ) Scheduling



## Key Features



Simple and organized — clear separation of job types.



Good control over system behavior.



Inflexible — a process cannot move between queues.



Starvation possible for lower-priority queues.



# Multilevel Queue (MLQ) Scheduling

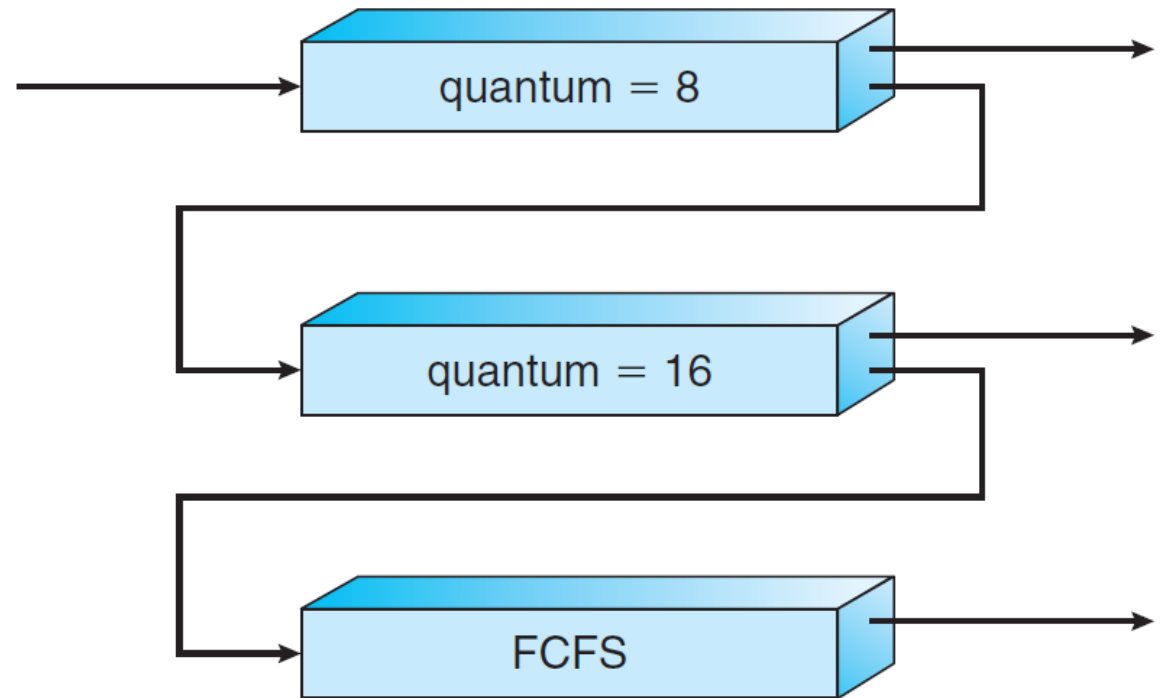
## Analogy

- Imagine an airport  with multiple check-in lines:
-  First Class, Business, and Economy.
- Each line has its own rules and service speed —
- but First Class always boards first! .



# Multilevel Feedback Queue (MLFQ)

- The Multilevel Feedback Queue (MLFQ) is an enhanced and flexible version of the Multilevel Queue (MLQ) algorithm.
- Unlike MLQ, processes can move between queues depending on their behavior and CPU usage.
- Its goal is to dynamically adapt to the needs of each process.



Multilevel feedback queues.



# Multilevel Feedback Queue (MLFQ)



## How It Works

- ❶ The system maintains multiple queues with different priorities.
- ❷ Each queue uses its own scheduling algorithm (e.g., RR, FCFS).
- ❸ New processes start in the highest-priority queue.
- ❹ If a process uses too much CPU time 🕒 → it's moved down to a lower-priority queue.
- ❺ If a process waits too long 🕒 → it's moved up (aging).

# Multilevel Feedback Queue (MLFQ)



## Key Features

-  Adaptive: Adjusts process priority dynamically.
-  Efficient: Gives quick response to short, interactive jobs.
-  Fair: Long processes still get CPU time eventually.
-  Complex: Requires fine-tuning of queue parameters (quantum, number of queues, aging rules).



# Multilevel Feedback Queue (MLFQ)

## Analogy

- Think of a theme park  with several lines:
-  Fast-pass riders (short jobs) start at the top queue.
-  Longer rides move down to slower lanes.
- If someone waits too long, they get bumped back up —
- so everyone gets a fair turn eventually! 



# MLQ vs. MLFQ Scheduling

Feature

**Multilevel Queue (MLQ)**

**Multilevel Feedback Queue (MLFQ)**

**Process Movement**

✗ Fixed — a process stays in its assigned queue.

✓ Flexible — processes can move up or down between queues.

**Priority Handling**

Static priorities between queues.

Dynamic priorities adjusted by feedback (promotion/demotion).

**Flexibility**

Rigid structure.

Adaptive and self-tuning.

**Starvation**

Possible for low-priority queues.

Reduced through aging and feedback.

**Use Case**

Clear separation of job types (e.g., batch vs. interactive).

Mixed workloads in modern OS (Windows, Linux).

- MLQ → Static, fixed hierarchy of queues.
- MLFQ → Dynamic, self-adjusting hierarchy that learns from process behavior.