

# Lecture#09

## Scheduling Algorithms

---

# Multi-Level Queue and Multi-Level Feedback Queue Algorithm

---



# Multi-Level Queue (MLQ) Basics

---

- Processes are permanently assigned to a queue
- Multiple queues with different priorities
- Each queue has its own scheduling algorithm
- Higher priority queues are served first
- Common in systems with distinct process categories

# MLQ Queue Structure

---

- System Processes (Highest Priority)
- Interactive Processes
- Batch Processes
- Student Processes (Lowest Priority)
- Different scheduling per queue: Round Robin for interactive, FCFS for batch



# MLQ Priority Handling

---

- Fixed priority between queues
- No movement between queues
- Starvation risk for lower-priority processes
- Solution: Time slicing between queues (e.g., 80% foreground, 20% background)

# MLQ: Advantages & Disadvantages

---

- Advantages:
  - Simple to implement
  - Low scheduling overhead
  - Good for fixed priority systems
- Disadvantages:
  - Inflexible, possible starvation
  - No adaptation to process behavior



# Introduction to Multi-Level Feedback Queue (MLFQ)

---

- Evolution of MLQ, allows queue movement
- Adapts to process behavior dynamically
- Favors shorter, I/O-bound processes
- Multiple queues with different priorities

# MLFQ Queue Structure

---

- Multiple queues with different time quantum
- Higher priority queues have shorter time quantum
- Lower priority queues have longer time quantum
- New processes enter highest priority queue
- Processes move between queues based on usage



# MLFQ Operation Rules

---

- Higher priority queue processes run first
- Same priority processes run Round Robin
- Process uses full time quantum: demoted
- Process yields before quantum ends: stays

# MLFQ Priority Adjustment

---

- CPU-bound processes move to lower queues
- I/O-bound processes stay in higher queues
- Starvation prevention via priority boost
- Periodic boost returns all processes to top queue



# MLFQ Time Quantum Example

---

- Queue 1: 8 milliseconds
- Queue 2: 16 milliseconds
- Queue 3: 32 milliseconds
- Queue 4: 64 milliseconds
- Exponential time quantum increases efficiency

# MLFQ: Advantages

---

- • Adaptive to process behavior
- • Favors interactive processes
- • Prevents starvation
- • Balances response time and throughput
- • Fair CPU time distribution



# MLFQ: Disadvantages

---

- More complex implementation
- Higher overhead than MLQ
- Possible gaming of the system
- Requires careful parameter tuning

# Comparing MLQ and MLFQ

---

- MLQ:
  - Fixed priorities, No queue movement
  - Simple but inflexible
- MLFQ:
  - Dynamic priorities, Adaptive behavior
  - More complex but efficient



# Best Practices & Implementation

---

- • Choose based on system needs
- • Monitor workload characteristics
- • Implement aging mechanisms
- • Balance complexity vs. performance