

Chapter-8

# 운영 체제

정내훈

2023년 가을학기  
게임공학과  
한국공학대학교



# Chapter 8

## 스케줄링 : 멀티레벨 피드백 큐



# 멀티레벨피드백큐(MLFQ)

- 실제 사용 중인 스케줄링 방식
- 목표:
  - 반환 시간 최적화 → Run shorter jobs first → **작업의 길이를 몰라도** 동작 가능하게 하자
  - 반응시간 최소화 → 작업의 성격에 따라서



# MLFQ: 기본 규칙

- MLFQ 여러 개의 서로 다른 큐(**queue**)를 갖는다.
  - 각 큐는 서로 다른 우선순위를 갖는다.
- 실행 준비 상태의 작업은 큐에 위치한다.
  - 높은 우선 순위 큐에 있는 작업이 먼저 실행된다.
  - 같은 큐에 있는 작업들은 라운드로빈으로 스케줄링 된다.

**Rule 1:** If  $\text{Priority}(A) > \text{Priority}(B)$ , A 실행 (B 대기).

**Rule 2:** If  $\text{Priority}(A) = \text{Priority}(B)$ , A와B가 RR로 실행.

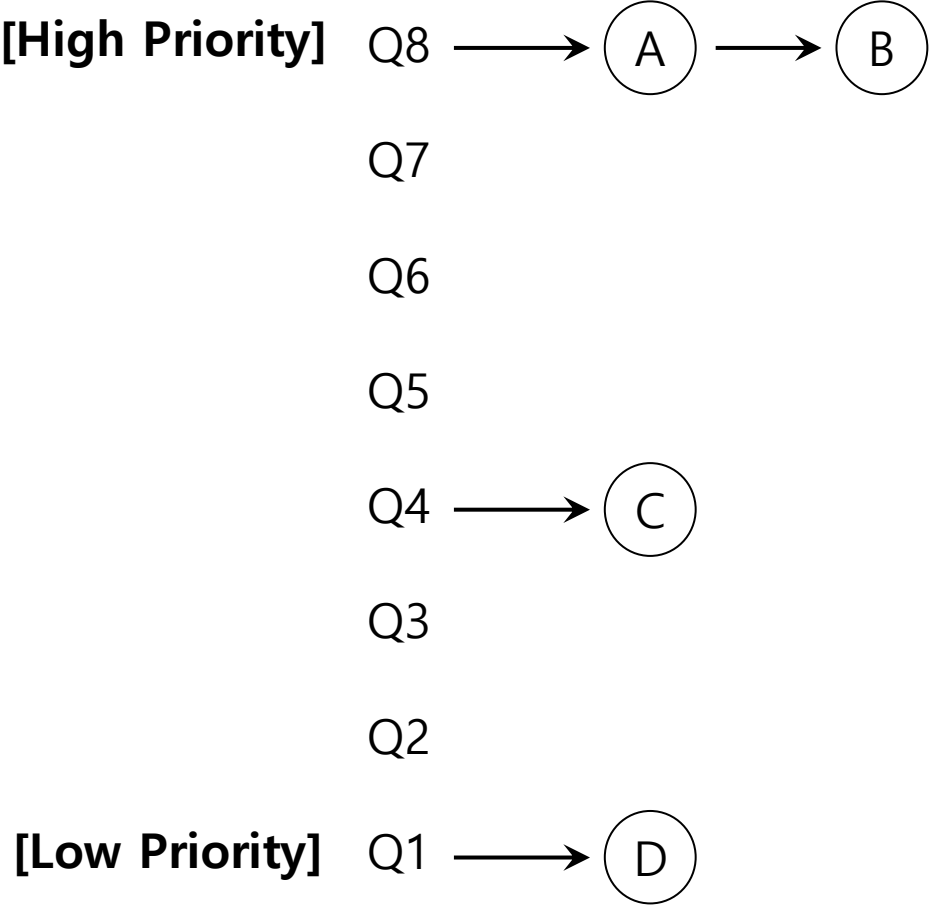


# MLFQ: 기본 규칙 (Cont.)

- MLFQ는 관찰된 행동에 따라 작업이 속할 우선 순위 큐가 정해진다.
- 예:
  - 자주 IO요청을 해서 CPU를 놓아주는 작업 → 높은 우선 순위
  - 긴 시간동안 CPU를 사용하는 작업 → 낮은 우선 순위.



# MLFQ 예





# MLFQ: 우선순위 변경

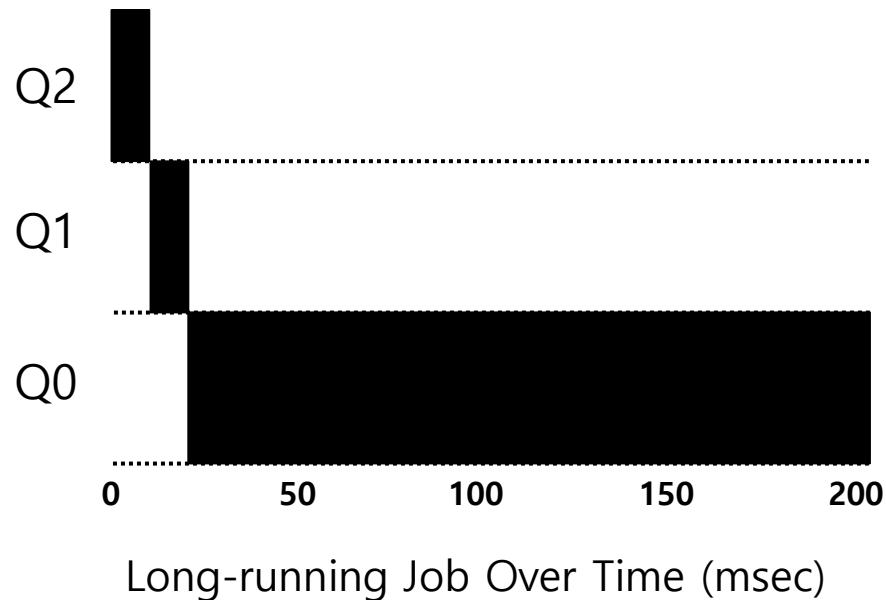
- MLFQ 우선 순위 변경 알고리즘:
  - **Rule 3:** 작업이 처음 시작되면 가장 높은 우선 순위를 갖는다.
  - **Rule 4a:** 작업이 타임 슬라이스를 전부 사용하면 우선 순위가 낮아진다. (낮은 우선순위 큐로 이동)
  - **Rule 4b:** 작업이 타임 슬라이스 사용 도중 CPU를 놓아 주면 우선 순위를 높인다.

이런 방식으로, MLFQ는 SJF를 흉내낸다.



# 예 1: 긴 실행시간 작업

- A three-queue scheduler with time slice 10ms







## 예 2: 작은 작업과 함께

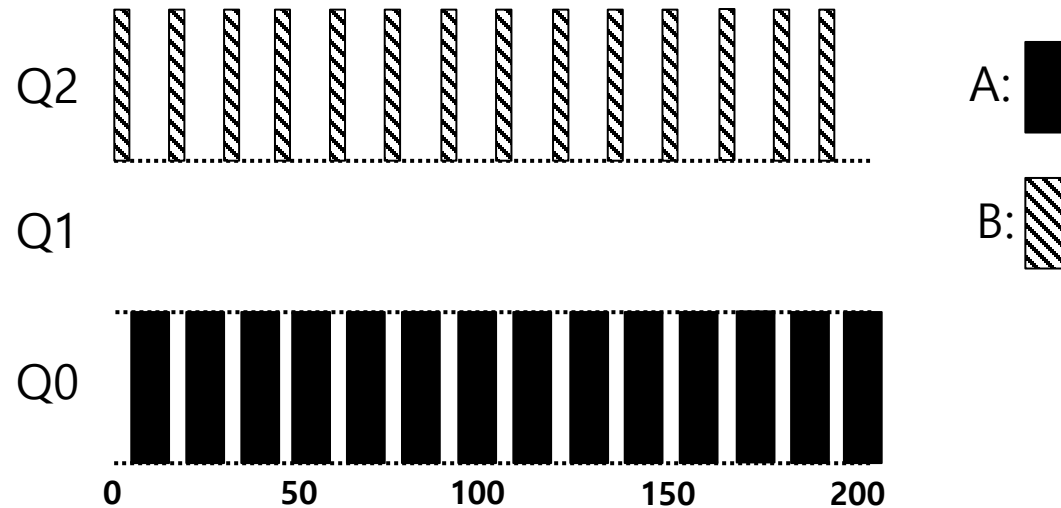
- 가정:
  - Job A:** CPU를 오래 사용하는 작업
  - Job B:** 짧게 짧게 실행되는 작업(20ms runtime)
  - A는 계속 실행 중, 중간에 B가 도착 (100ms).





## 예 3: 입출력 작업은?

- 가정:
  - Job A:** CPU 중심 작업
  - Job B:** I/O요청 전 1ms 정도만 실행되는 대화형 작업(**interactive job**).



A Mixed I/O-intensive and CPU-intensive Workload (msec)

MLFQ방식은 대화형(interactive)작업의 우선 순위를 높게 유지한다.



# 현재 MLFQ의 문제점

- 기아 (Starvation)
  - “너무 많은“ 대화형 작업들이 있다면.
  - 오랜 시간 실행 작업은 CPU를 사용하지 못할 것이다.
- 교활한 사용자
  - 99%의 time slice만 사용하고 I/O를 요청
  - 계속 대부분의 CPU 시간을 사용할 수 있다.
- 실행하면서 작업의 성격이 바뀔 수 있다.
  - CPU 위주(bound) process → I/O 위주(bound) process

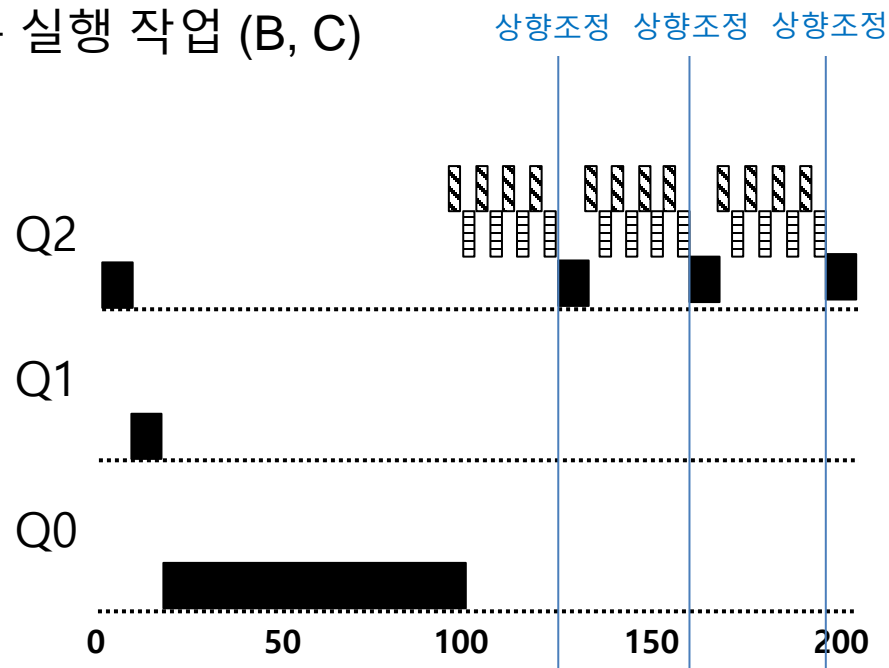
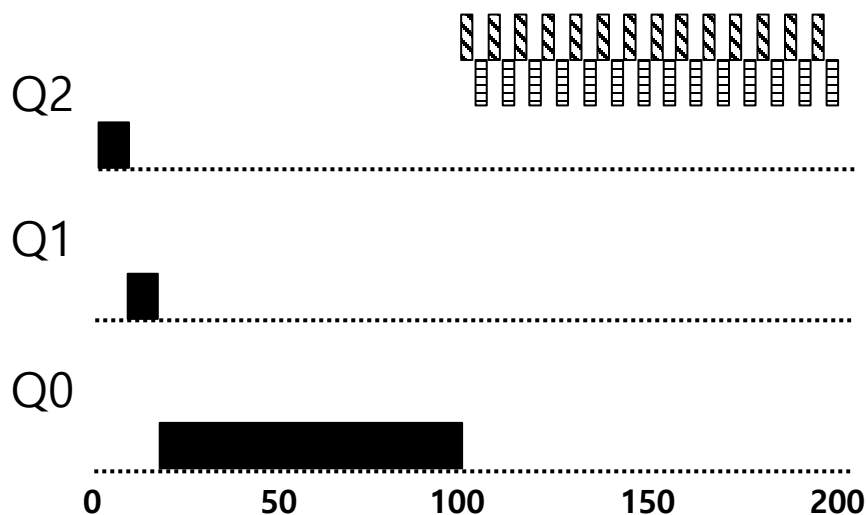


# 우선순위 상향 조정




- **Rule 5:** 일정한 주기  $S$  마다, 모든 작업을 가장 위의 큐로 이동.

– 보기:

- 긴 실행 작업(A)와 대화형 짧은 실행 작업 (B, C)



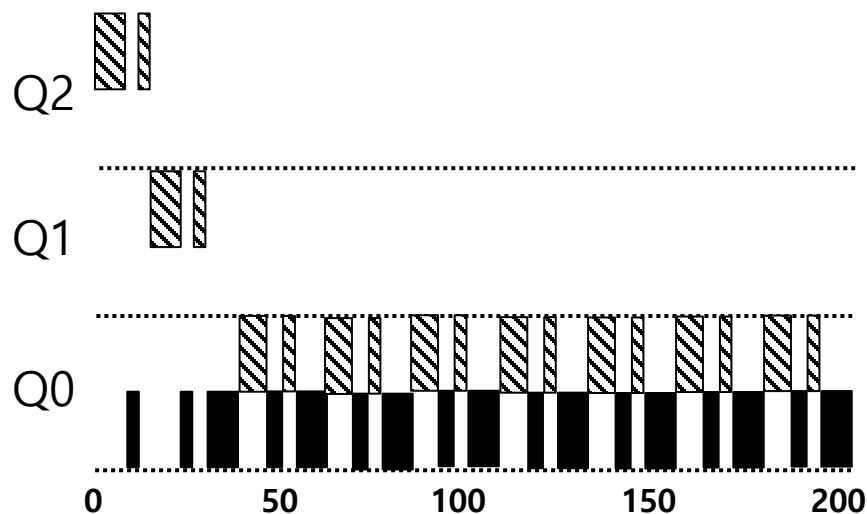
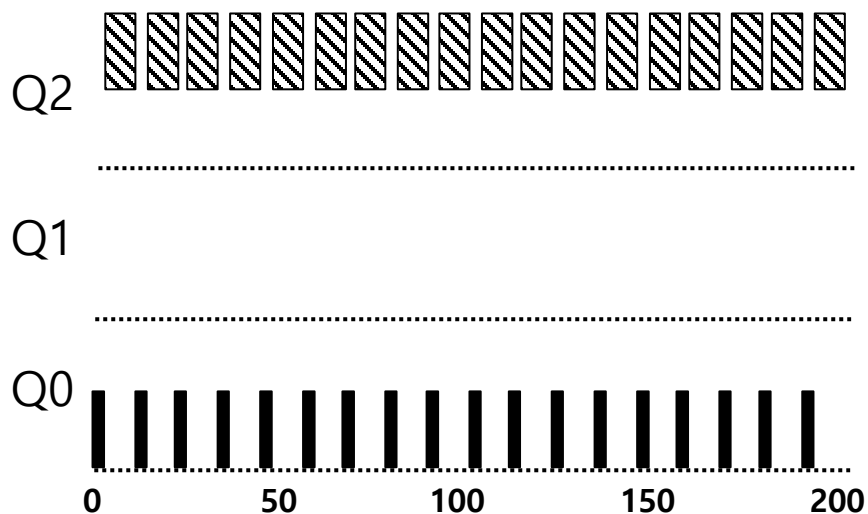
우선순위 상향 조정 없음(왼쪽) 있음(오른쪽)

A:  B:  C: 



# 더 나은 시간 측정

- 스케줄러 속이기에 대한 대처?
- Solution:
  - **Rule 4** (Rewrite Rules 4a and 4b): 총 사용시간 측정. 해당 레벨에서 주어진 시간을 다 사용하면 (I/O 요청 회수에 관계 없이) 우선 순위 강등.



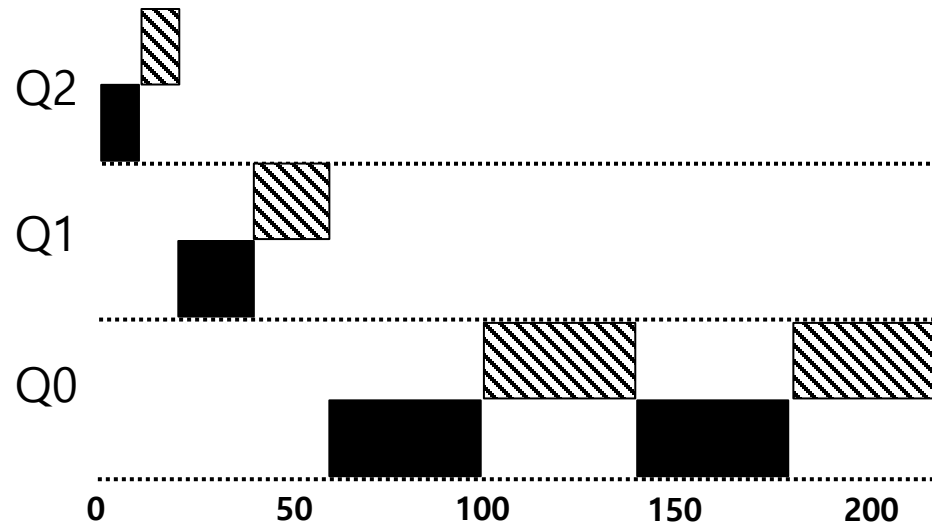
Without(Left) and With(Right) Gaming Tolerance



# MLFQ 조정과 다른 쟁점들

낮은 우선순위에 더 긴 타임 슬라이스

- 높은 우선순위 큐 → 작은 타임 슬라이스
  - E.g., 10 or fewer milliseconds
- 낮은 우선순위 큐 → 긴 타임 슬라이스
  - E.g., 100 milliseconds



Example) 10ms for the highest queue, 20ms for the middle,  
40ms for the lowest



# MLFQ: 요약

- MLFQ 규칙 정리:
  - **규칙 1:** 우선순위(A) > 우선순위(B)이면 A실행.
  - **규칙 2:** 우선순위(A) = 우선순위(B)이면 A,B를 RR로 실행.
  - **규칙 3:** 작업이 도착하면 가장 높은 우선순위에서 시작
  - **규칙 4:** 작업이 지정된 단계에서 배정받은 시간을 소진하면(CPU 포기 횟수 무관) 우선 순위 감소(한단계 아래 큐로 이동)
  - **규칙 5:** 일정 주기 S 마다 모든 작업을 가장 높은 우선 순위 큐로 이동.
- 실제 운영체제들도 비슷한 스케줄링 알고리즘으로 동작한다.

## 9: Scheduling: 비례 배분

---

생략