

2. 운영체제_멀티 큐, 멀티 피드백 큐_ 발표용

- 멀티 큐(Multi Queue)
 - 운영체제에서 준비 상태와 대기 상태의 프로세스들은 보통 큐로 관리 됨.
 - 멀티 큐 구조는 아래처럼 상태별로 큐를 분리하여 관리하는 방식
 1. 준비 상태의 큐
 2. 대기 상태의 큐
- 멀티 피드백 큐(MLFQ, Multi-Level Feedback Queue)
 - MLFQ는 프로세스의 행동 패턴(짧은 작업/ 긴 작업/대화형 작업 등)을 기반으로 우선순위를 동적으로 조정하여 짧은 작업을 빠르게 처리하고, 긴 작업에도 CPU 기회를 부여해 공정성을 보장하는 스케줄링 알고리즘임.
 - 목표
 1. 반한 시간(Thrunaround Time) 최소화
 - a. 짧은 작업을 우선 처리하여 반환 시간을 줄임
 2. 응답 시간(Response Time) 최적화
 - a. 대화형 작업에 빠르게 반응
 3. 공정성(Fairness) 보장
 - a. 긴 작업도 일정 주기마다 CPU를 할당받을 수 있도록 함
 - 구조
 - 여러 개의 우선순위 큐로 구성되며, 각 큐는 서로 다른 우선순위를 가짐
 1. 새로 들어온 작업은 가장 높은 우선순위 큐에서 시작
 2. CPU 사용 시간이 길어질수록 점진적으로 낮은 우선순위 큐로 이동
 3. 높은 우선순위 큐는 짧은 작업 / 사용자와 상호작용이 많은 작업을 처리
 4. 낮은 우선순위 큐는 시간이 오래 걸리는 작업(배치 작업 등)을 처리
 - MLFQ의 규칙
 1. 높은 우선순위 작업 우선 실행

- a. 우선순위 높은 큐부터 스케줄링 됨
 - b. 같은 우선순위 큐에서는 라운드 로빈 방식 사용
→ 도착 순서대로 일정 시간만큼 CPU 사용
2. 작업의 우선순위 조정
 - a. 새 작업은 항상 최상위 우선순위 큐에서 시작
 - b. 주어진 시간 할당량을 모두 사용하면 우선순위 하락
 - c. 일정 시간이 지나면 모든 작업의 우선순위가 초기화되어 기아 방지
 3. 동적 피드백
 - a. 작업의 CPU 사용 패턴을 보고 CPU 많이 쓰면 아래 큐로, I/O 많이 하고 짧게 CPU 쓰면 위 큐로 올림 → 동적으로 행동 기반 피드백 수행
- 동작 방식
 1. 프로세스 도착
 - a. 새 프로세스는 항상 가장 높은 우선순위 큐에 삽입되며 시작함
 2. CPU 실행
 - a. 각 단계는 정해진 시간 할당량동안 실행됨
 3. 우선순위 하락
 - a. 시간 할당량을 모두 사용해도 끝나지 않거나 I/O로 인해 중단된 경우 낮은 우선순위 큐로 이동됨
 4. 우선순위 상승
 - a. 일정 시간이 지나면 모든 프로세스의 우선순위를 초기화하여 기아를 방지함
 - MLFQ가 해결하려는 문제
 1. 짧은 작업 우선 처리 → 반환시간 최적화
 2. 대화형 작업 응답시간 최적화 → 사용자 체감 성능 향상
 3. 우선순위 초기화를 통한 공정성 확보