**프로세스 스케줄링**

다중프로그래밍

-여러 개 프로세스가 시스템 내 존재

-자원을 할당 할 프로세스를 선택 – 스케줄링

-자원 관리: 1. 시간 분할 관리(processor) 2. 공간분할 관리(memory)

**목적**: 성능 향상, 시스템 성능 지표(응답시간, 작업 처리량, 자원 활용도), 목적에 맞는 지표

대기시간, 응답시간, 실행시간, 반환 시간

**스케줄링 기준**

1. 고려하는 항목
2. 프로세스 특성
3. 시스템 특성
4. 프로세스의 긴급성
5. 프로세스 우선순위
6. 총 실행 시간

CPU burst vs I/O burst

프로세스 수행 – CPU 사용 + I/O 대기 CPU burst – CPU 사용시간I/O burst – I/O 사용시간

**스케줄링의 단계**

빈도 자원에 따른 구분

-long term scheduling: 1. Job scheduling

2. 다중 프로그래밍 정도 조절

3. CPU burst I/O burst 적절히 선택

4. 시분할- long term scheduling 덜 중요

- Mid-term scheduling: 1. 메모리 할당 결정

-short-term scheduling: 1. Process scheduling- 프로세서를 할당할 프로세서를 결정

2. 가장 빈번하게 발생- 매우 빠름

**스케줄링 정책**

선점 vs 비선점

비선점-자원을 반납할 때까지 사용 평균응답시간 증가 context switch overhead 적음

선점- 타인에 의해 자원을 빼앗길 수 있음 응답 성이 높음 overhead가 많음

우선순위

-Static priority: 변하지 않는 우선순위

-Dynamic priority: 상태 변화에 따라 우선순위가 변경

**기본 스케줄링 알고리즘**

**FCFS(first-come-first-service)-** 선착순 알고리즘

-non-preemptive scheduling

-스케줄링 기준: 도착시간 기준

-자원을 효율적으로 사용 가능- overhead가 적음

-batch system에 적합 interactive system에 부적합

단점: convoy effect-대기시간이 김,

**RR (Round-Robin)**

-preemptive scheduling

-스케줄링 기준: 도착시간 시준

-자원 사용 제한 시간이 있음-시간 지나면 자원 반납, 독점 방지, overhead가 큼

- 대화형 시분할 시스템에 적합

-time quantum이 시스템 성능 결정하는 핵심 요소