
Chapter 04 CPU 스케줄링

02 스케줄링 시 고려 사항

01 선점형 스케줄링과 비선점형 스케줄링

선점형 스케줄링

- 어떤 프로세스가 CPU를 할당받아 실행 중이더라도 운영체제가 CPU를 강제로 빼앗을 수 있는 스케줄링 방식
 - 하나의 프로세스가 CPU를 독점할 수 없기 때문에 빠른 응답 시간을 요구하는 대화형 시스템이나 시분할 시스템에 적합하나, 문맥 교환 같은 부가적인 작업으로 인해 낭비가 생김
 - 대부분의 저수준 스케줄러는 선점형 스케줄링 방식을 사용
- 예시) 인터럽트 처리 (CPU가 인터럽트를 받으면 현재 실행 중인 작업을 중단하고 커널을 깨워서 인터럽트를 처리시키며, 인터럽트 처리가 완료되면 원래의 작업으로 돌아감)

비선점형 스케줄링

- 어떤 프로세스가 CPU를 점유하면 다른 프로세스가 이를 빼앗을 수 없는 스케줄링 방식
- 선점형 스케줄링보다 스케줄러의 작업량이 적고 문맥 교환에 의한 낭비가 적으나, CPU 사용 시간이 긴 프로세스 때문에 CPU 사용 시간이 짧은 여러 프로세스가 오랫동안 기다리게 되어 전체 시스템의 처리율이 떨어짐
- 과거 일괄 작업 시스템에서 사용

02 프로세스 우선순위

- CPU 스케줄러는 각 프로세스에 우선순위를 부여하는데,
우선순위가 높다는 것은 더 빨리 자주 실행된다는 의미
- 시스템에는 다양한 우선순위의 프로세스가 공존하며,
우선순위가 높은 프로세스가 CPU를 먼저, 더 오래 차지하게 됨

커널 프로세스
일반 프로세스

입출력 집중 프로세스
CPU 집중 프로세스

전면 프로세스
후면 프로세스

대화형 프로세스
일괄 처리 프로세스

02 프로세스 우선순위

커널 프로세스

- 우선순위가 높음

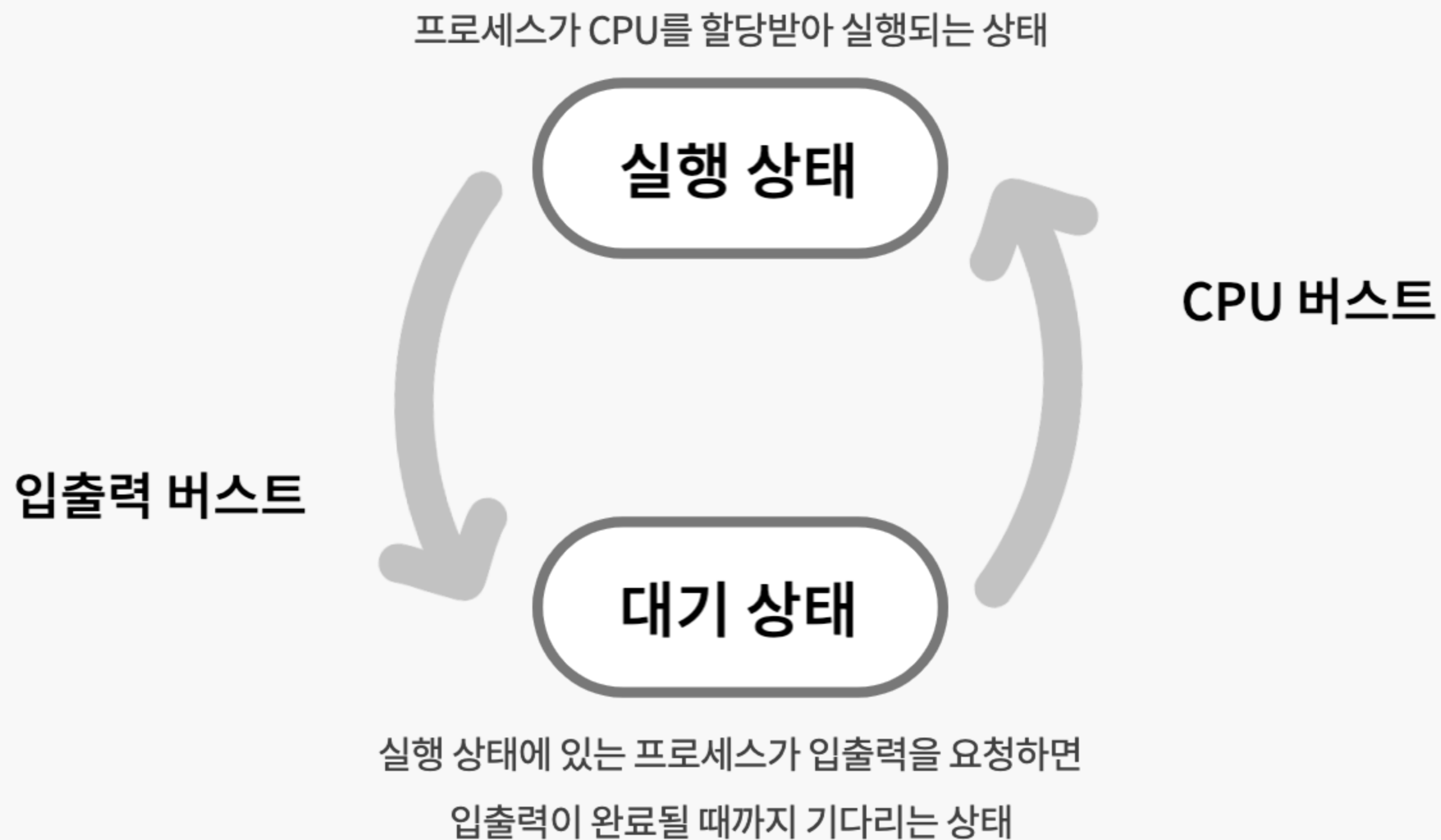
일반 프로세스

- 우선순위가 낮음

- 우선순위를 사용자가 조절할 수 있음

다만, 우선순위를 조절할 때는 해당 프로세스뿐 아니라 다른 프로세스의 실행 속도에도 영향을 미친다는 것을 주의해야 함

03 작업 형태에 따른 프로세스 분류



03 작업 형태에 따른 프로세스 분류

입출력 집중 프로세스

- 입출력을 많이 사용하는 프로세스
- 입출력 버스트(= 입출력 작업)가 많은 프로세스

예시) 저장장치에서 데이터를 복사하는 일

- 스케줄링 시 입출력 집중 프로세스의 우선순위를 CPU 집중 프로세스보다 높이면 시스템 효율이 향상됨
: 입출력 집중 프로세스가 실행 상태로 가면 입출력 요구에 의해 대기 상태로 옮겨지기 때문에 다른 프로세스가 CPU를 사용할 수 있기 때문
(= 사이클 훔치기)

CPU 집중 프로세스

- CPU를 많이 사용하는 프로세스
- CPU 버스트(= CPU를 할당받아 실행하는 작업)가 많은 프로세스

예시) 수학 연산

04 전면 프로세스와 후면 프로세스

전면 프로세스

- GUI를 사용하는 운영체제에서 화면의 맨 앞에 놓인 프로세스
- 현재 입력과 출력을 사용하는 프로세스
- 사용자와 상호작용이 가능하여 상호작용 프로세스라고도 함
- 사용자의 요구에 즉각 반응해야 하기 때문에 우선순위가 높음

후면 프로세스

- 사용자와 상호작용이 없는 프로세스
 - 압축 프로그램처럼 사용자의 입력 없이 작동하기 때문에 일괄 작업 프로세스라고도 함
 - 우선순위가 낮음
-