

스마트시스템 운영체제 (LD01600)

김준철

정보시스템공학과

greensday@sungshin.ac.kr

4주차 강의 -2

		주차	강의 목차
휴강(9.30) (추석)	9.2	1	과목소개 / 운영체제 개요
	9.9	2	컴퓨터 시스템 구조
	9.16	3	process와 스레드1
	9.22	4	process와 스레드2, CPU스케줄링1
	10.7	5	CPU스케줄링2
	10.14	6	process 동기화
	10.21	7	교착 상태
	10.28	8	중간고사
	11.4	9	물리 메모리 관리
	11.11	10	가상메모리 기초
	11.18	11	가상메모리 관리
	11.25	12	입출력시스템1
	12.2	13	입출력시스템2, 파일시스템1
	12.9	14	파일시스템2
	12.16	15	기말고사

Operating Systems

ch.04 CPU scheduling1

01 scheduling의 개요

02 scheduling 시 고려 사항

03 Multilevel Queue

04 scheduling algorithm

CPU Scheduling의 개념

■ CPU Scheduling 스케줄링 의 개념

■ CPU scheduling:

- 여러 process가 번갈아 사용 하는 **CPU자원을 어떤 시점에 어떤 process에 할당할지 결정**
- PCB의 우선순위 고려하여 ready queue의 process 중 **실행할 것을 선택**

■ CPU scheduling의 필요성:

CPU를 많이 사용하는 작업 I/O를 많이 사용하는 작업	} 적절히 Scheduling할 필요가 있음
------------------------------------	--------------------------

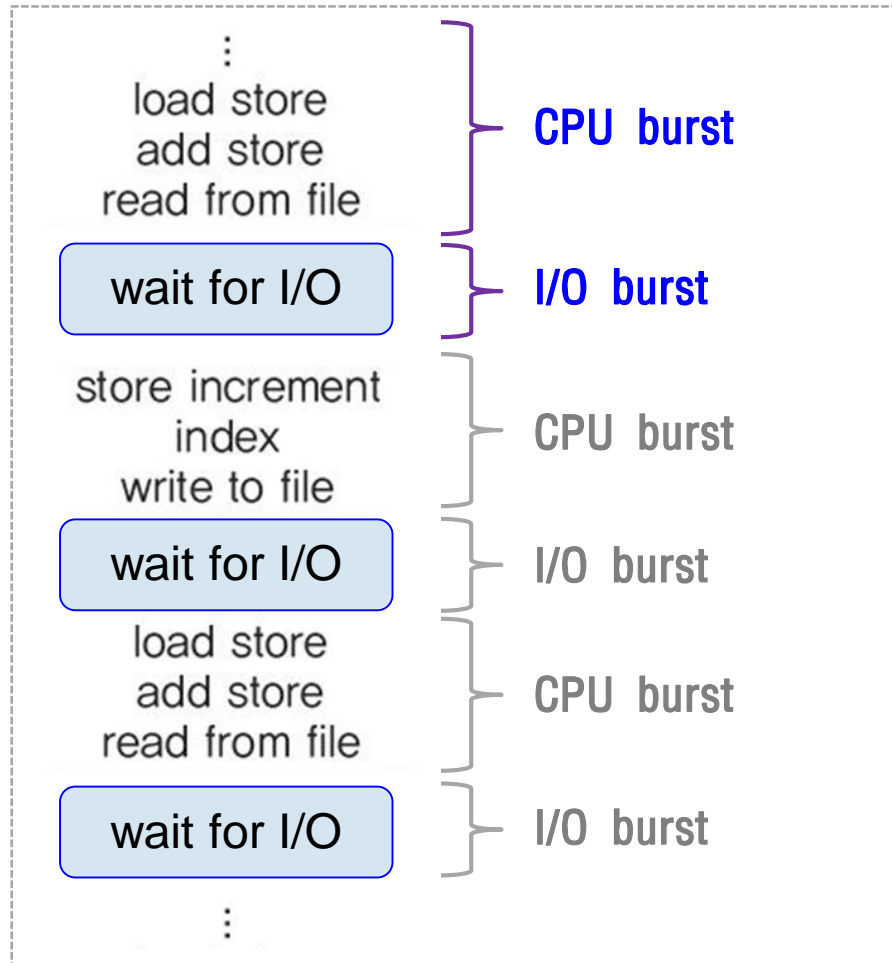
→ scheduling이 **시스템의 성능에 영향** 미침

■ CPU burst, I/O burst

- CPU burst : CPU를 할당 받아 실행하는 작업
- I/O burst : I/O를 하는 작업

CPU burst, I/O burst

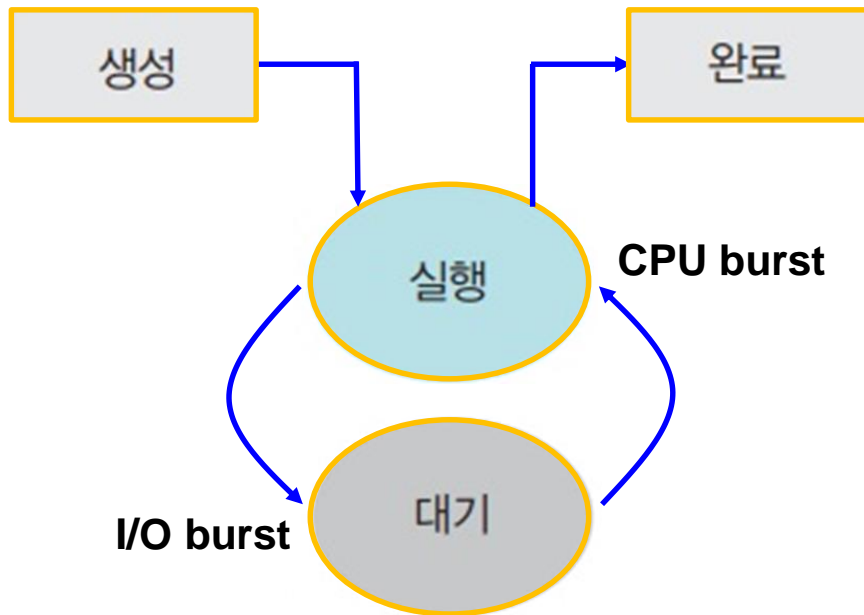
■ Process의 실행



[그림] 실행(process burst, CPU burst)와 입출력 대기(I/O burst)

CPU bound job, I/O bound job

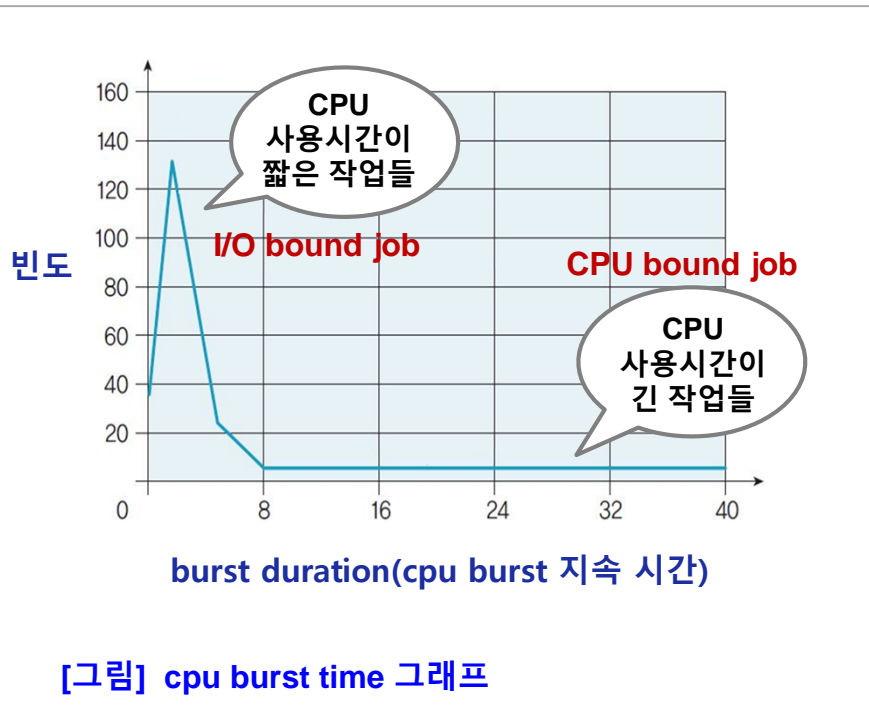
- CPU bound job (CPU bound process, CPU 집중 작업)
 - CPU를 많이 사용하는 process로 CPU burst(processor burst)가 많은 process (e.g.. 복잡한 연산을 하는 simulation program)
- I/O bound job (I/O bound process, 입출력 집중 작업)
 - 입출력을 많이 사용하는 process로 입출력(I/O) burst가 많은 process (e.g.. word 작업 program)



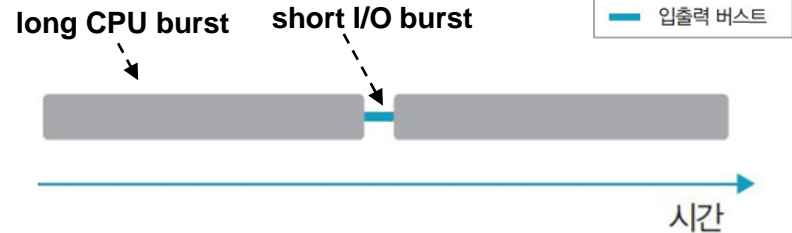
[그림] CPU burst와 I/O burst

CPU bound job, I/O bound job

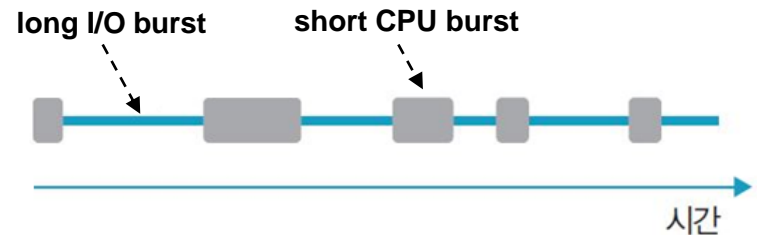
■ CPU burst(Process burst) time 그래프와 예



* CPU bound job



* I/O bound job

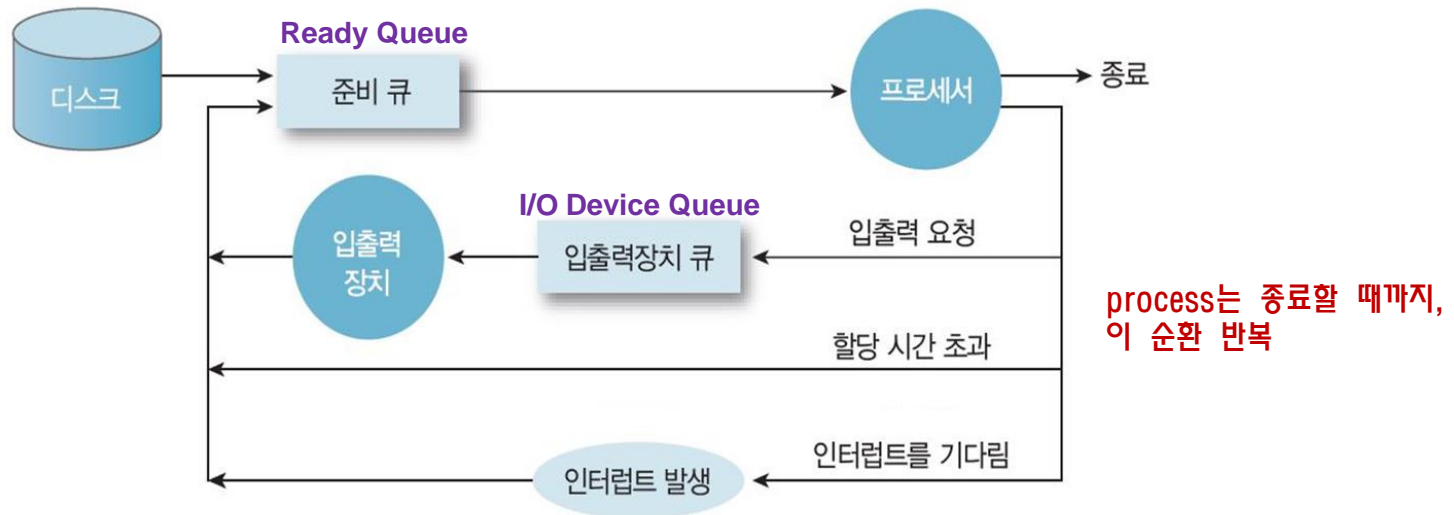


[그림] cpu burst가 긴 경우, 짧은 경우

- scheduling을 할 때 어떤 process의 우선순위를 높이는 것이 시스템 안의 전체 process의 종료시간 감소에 도움이 되겠는가? **I/O bound job**
- CPU사용의 균형 유지 :
 - I/O집중 process** - CPU를 짧게 사용하되 자주 사용하도록 함
 - CPU집중 process** - CPU를 길게 사용하되 사용 횟수를 줄임

Scheduling과 scheduler

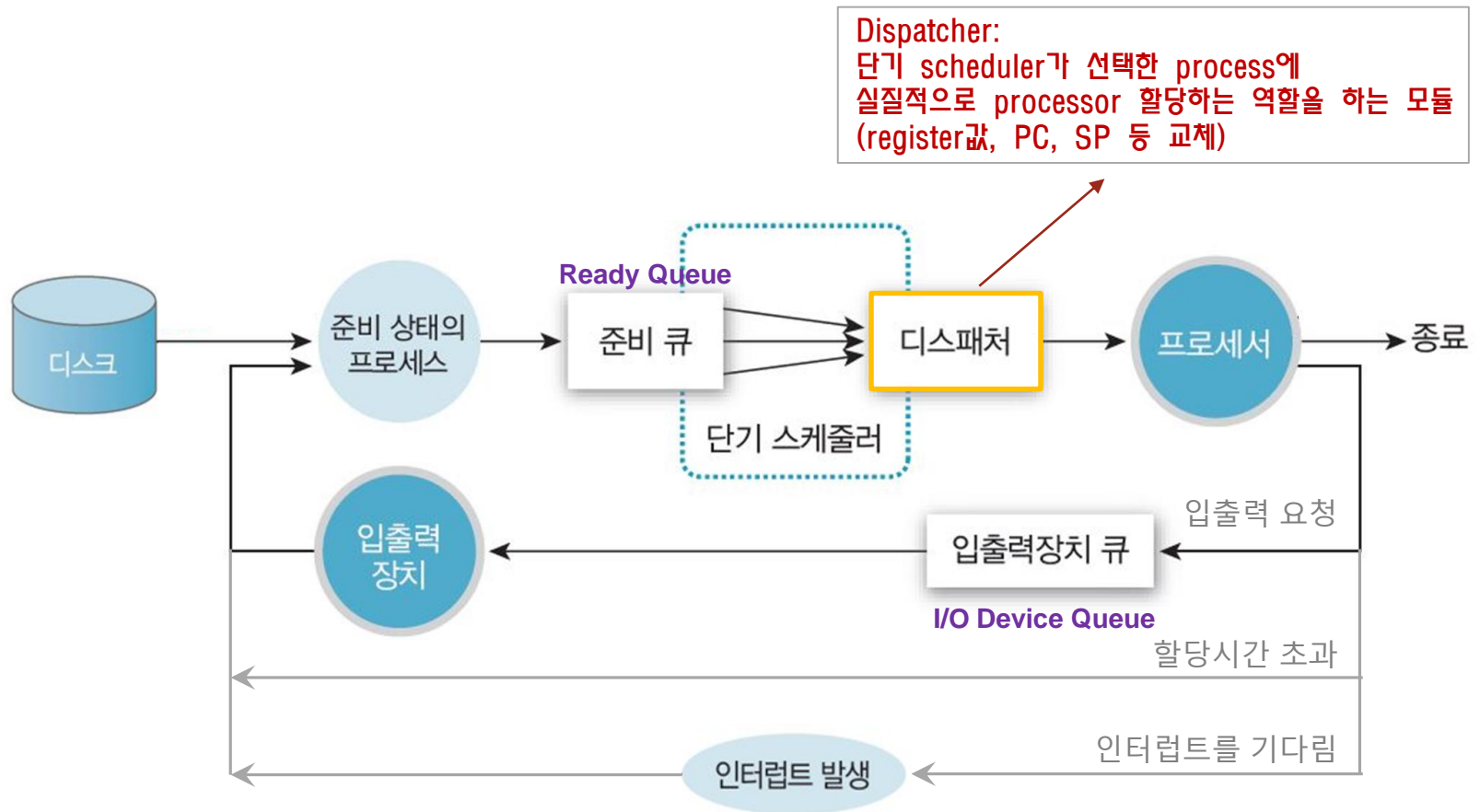
- Queueing diagram (큐잉 도표)
 - process scheduling 표현하는 방법



[그림] Queueing diagram 예

Scheduling과 scheduler

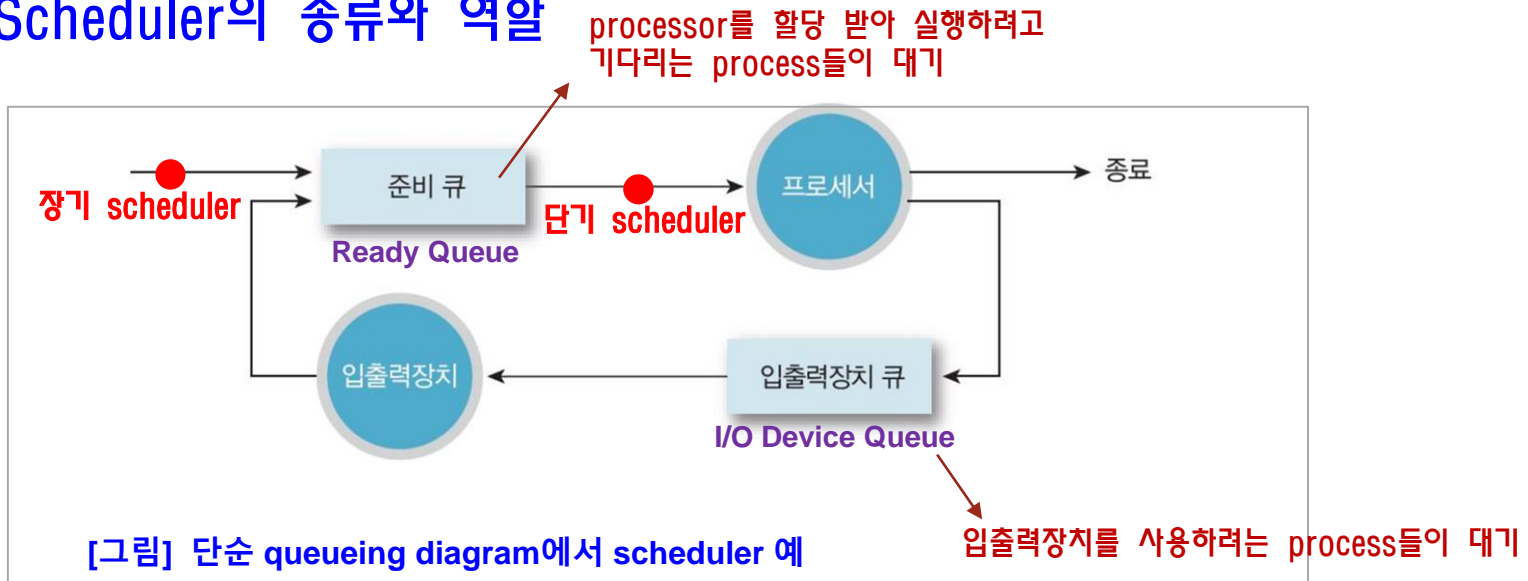
- Queueing diagram에서 dispatcher의 위치와 역할



[그림] Queueing diagram에서 dispatcher의 위치와 역할

Scheduling과 scheduler

■ Scheduler의 종류와 역할



1. Long term scheduler (장기 스케줄러, job scheduler)

- scheduling에 따라 디스크에서 메모리로 작업을 가져와서 처리할 순서 결정 (제출 시간, 작업 이름, 작업 길이(용량) 등의 정보 필요)

2. Short term scheduler (단기 스케줄러, CPU scheduler)

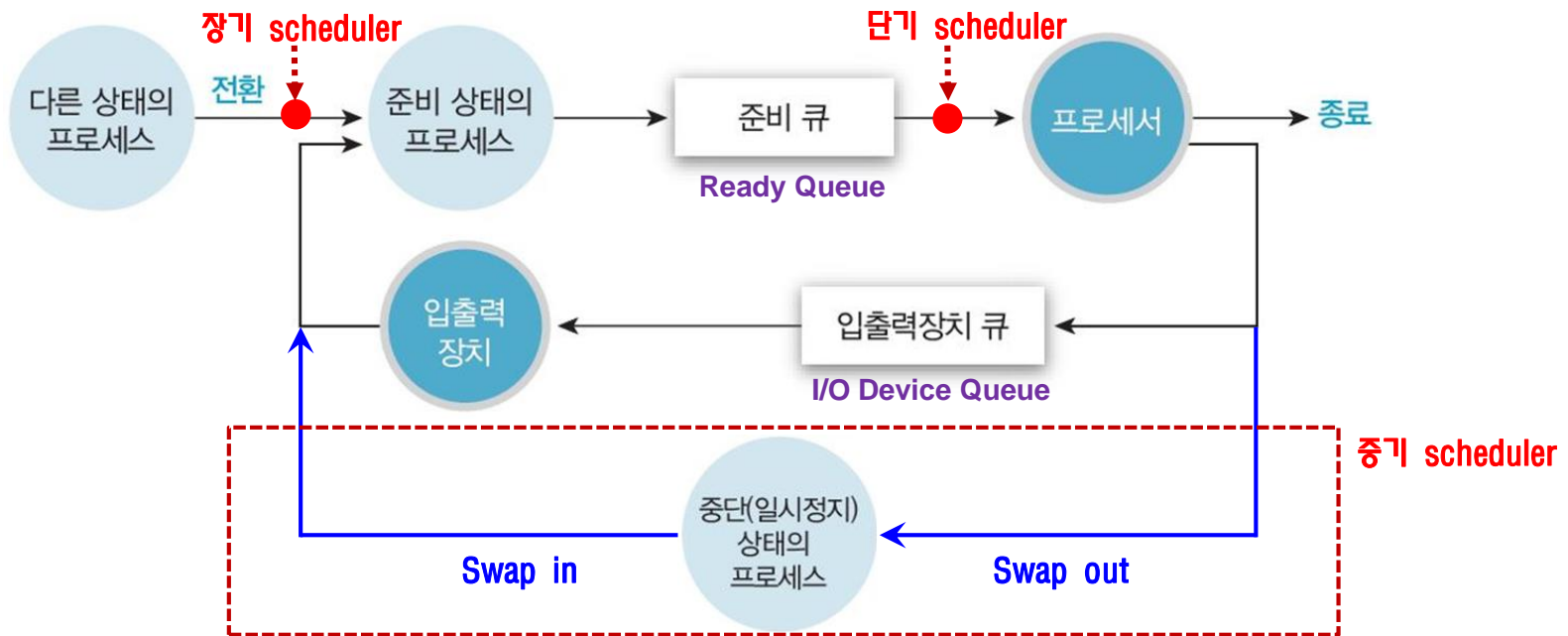
- 메모리에 적재된 process 중 실행 상태가 될 processor를 고르는 process scheduling
→ process가 실행하는 데는 필요한 자원의 요청 만족해야 함

Scheduling과 scheduler

■ Scheduler의 종류와 역할

3. Mid term scheduler (중기 스케줄러)

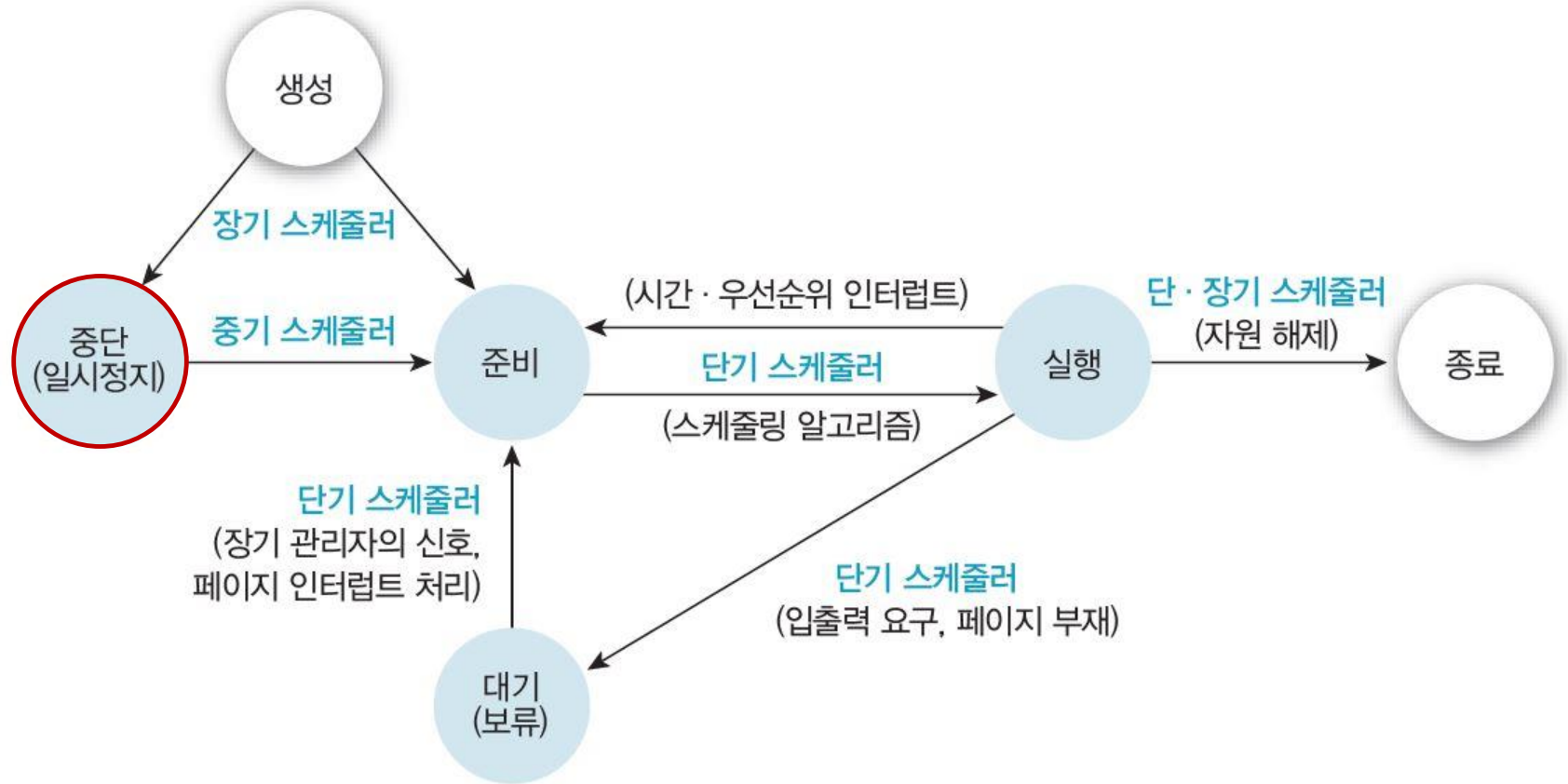
- Swap in(스왑 인)과 swap out(스왑 아웃)을 결정
- degree of multiprogramming을 줄일 수 있음
 - process들의 CPU 차치를 위한 경합이 심할 때 특정 process를 swap area로 내보낼 수 있음



[그림] 중기 scheduler가 추가된 queueing diagram

Scheduling과 scheduler

- Process 상태 변화와 scheduler의 역할



[그림] process 상태 변화와 scheduler의 역할

CPU Scheduling 방법

- **Preemptive scheduling(선점 스케줄링)**
 - OS가 **실행 상태에 있는 process의 작업을 중단**시키고 새로운 작업을 시작할 수 있는 방식
 - 빠른 응답 시간을 요구하는 **대화형 시스템이나 시분할 시스템에 적합** (하나의 process가 CPU를 독점 불가능)
- **Nonpreemptive scheduling(비선점 스케줄링)**
 - process가 종료되거나 자발적으로 대기 상태에 들어가기 전까지는 계속 실행되는 방식
(OS가 실행 상태에 있는 **process의 작업을 강제로 중단시키지 못함**)
 - preemptive보다 scheduler의 작업량이 적고 context switching에 의한 낭비도 적음
 - 전체 시스템의 **처리율이 떨어짐** (CPU 사용 시간이 긴 process가 있으면 CPU 사용 시간이 짧은 여러 process가 오랫동안 기다리게 됨)
 - **일괄 작업 시스템**에서 사용하던 방식

Scheduling algorithm의 성능 비교 기준

■ Scheduling algorithm(스케줄링 알고리즘)의 성능 비교 기준

■ CPU Utilization(CPU 사용률):

전체 시스템의 동작시간 중 CPU가 사용된 시간

■ throughput (처리량):

단위시간당 작업을 마친 process의 수

■ waiting time (대기시간):

process가 생성된 후 종료될 때까지 CPU에서 실행을 위해서 대기한 시간들의 합

■ response time (응답(반응)시간):

process가 ready상태에서 running상태로 되어, 첫 번째 반응(출력)이 나올 때 까지 걸리는 시간

■ turn-around time (반환(소요)시간):

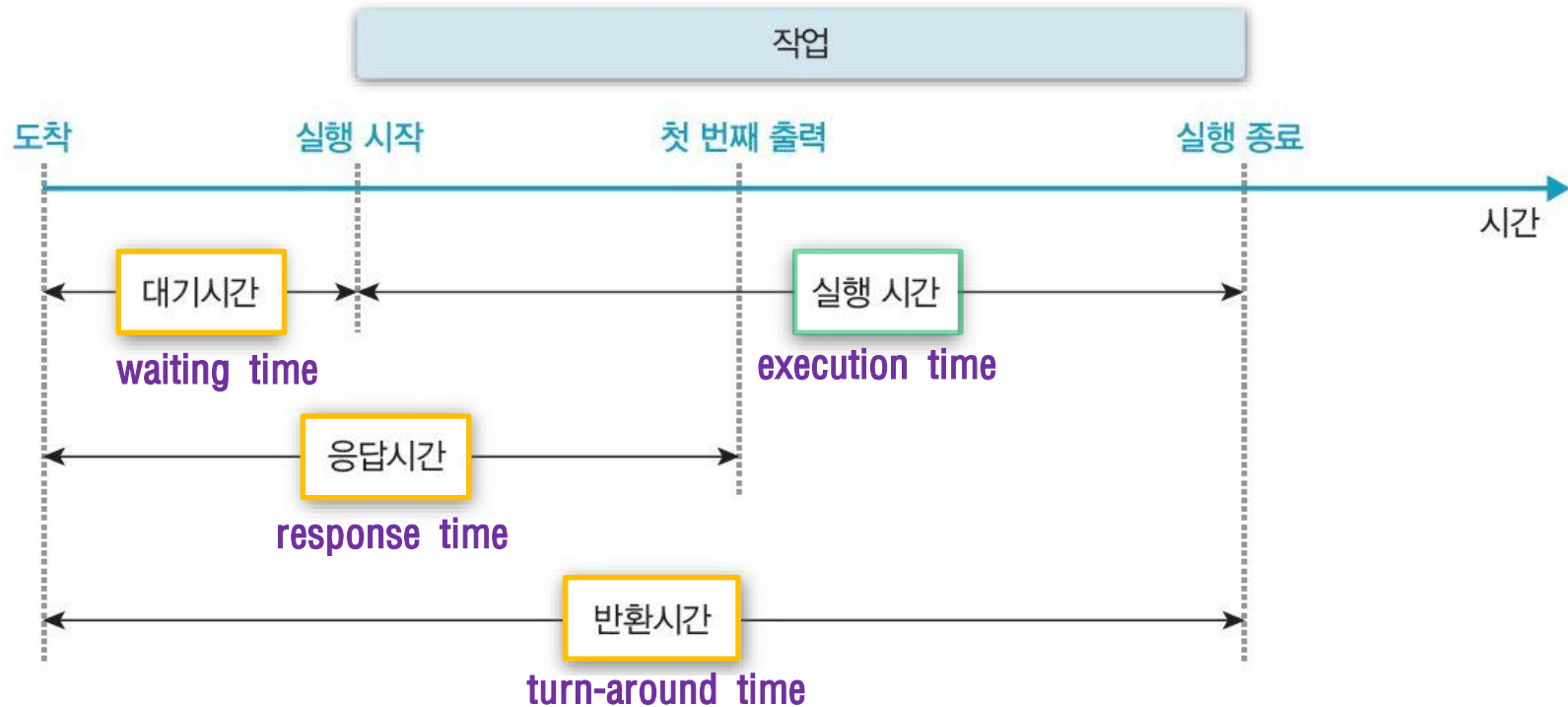
process 생성 후 종료되어 사용하던 자원을 모두 반환하는데 까지 걸리는 시간



시간에 관련된
성능 척도

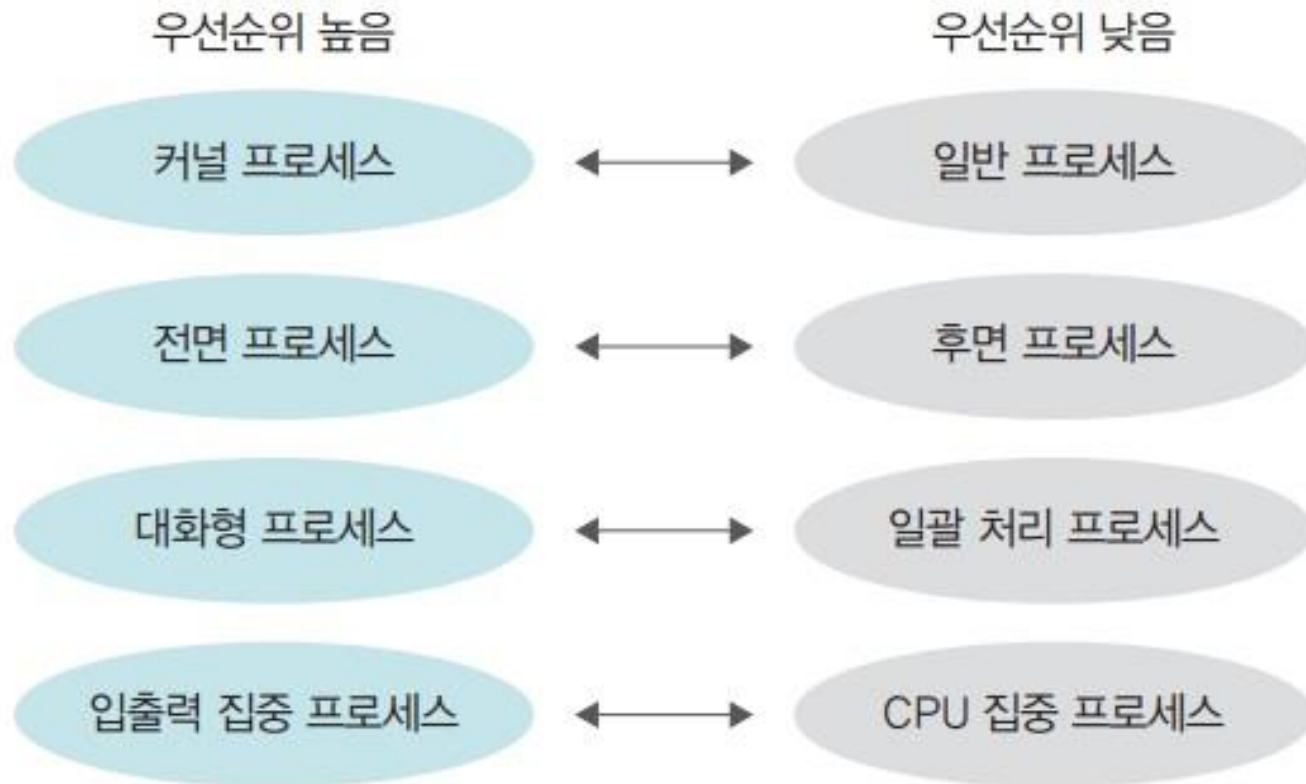
Scheduling algorithm의 선점 기준

- 시간에 관련된 Scheduling algorithm의 성능 척도



[그림] 대기시간, 응답시간, 반환시간의 관계

CPU scheduling의 고려사항



[그림] CPU scheduling 시 고려 사항



감사합니다.

