



## 2강. 프로세스 개요

방송대 컴퓨터과학과  
김진욱 교수

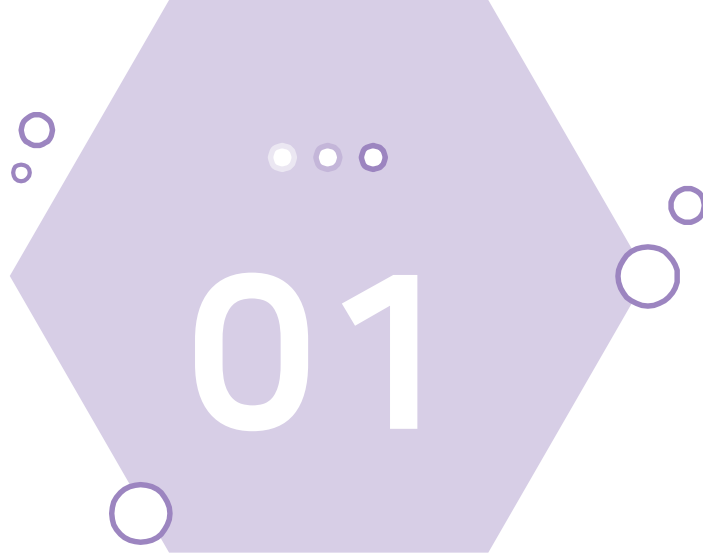


# 목차

01 프로세스

02 쓰레드

03 스케줄링



# 프로세스

# 프로세스

## ■ 프로세스(process): 실행 중인 프로그램

작업 관리자					
파일(F) 옵션(O) 보기(V)					
프로세스 성능 앱 기록 시작프로그램 사용자 세부 정보 서비스					
이름	상태	6% CPU	31% 메모리	0% 디스크	0% 네트워크
앱 (5)					
> Google Chrome(32 bit)		0%	32.7MB	0.1MB/s	0Mbps
> Hancom Office Hanword 2010(...		1.3%	12.9MB	0MB/s	0Mbps
> Microsoft PowerPoint(32 bit)		0.1%	30.4MB	0.1MB/s	0Mbps
> Windows 탐색기		0.3%	29.0MB	0MB/s	0Mbps
> 작업 관리자		0.5%	13.2MB	0.1MB/s	0Mbps
백그라운드 프로세스 (84)					
AcroTray(32 bit)		0%	1.2MB	0MB/s	0Mbps

# 프로세스

## 프로세스(process)

작업 관리자

파일(F) 옵션(O) 보기(V)

프로세스 성능 앱 기록 시작프로그램

이름

앱 (5)

- > Google Chrome(32 bit)
- > Hancom Office Hanword 2010...
- > Microsoft PowerPoint(32 bit)
- > Windows 탐색기
- > 작업 관리자

백그라운드 프로세스 (84)

- AcroTray(32 bit)

응용 프로그램 도구 Hwp80

파일 홈 공유 보기 관리

← → ↕ ↑ < Program Files (x86) > Hnc > Hwp80 Hwp80 검색

이름	수정한 날짜	유형	크기
HncShellExt.dll	2010-02-12 ...	응용 프로그...	
HncShellExt64.dll	2010-02-12 ...	응용 프로그...	
<input checked="" type="checkbox"/> Hwp.exe	2010-02-12 ...	응용 프로그램	
HwpABase.dll	2010-02-12 ...	응용 프로그...	
HwpAGrp1.dll	2010-02-12 ...	응용 프로그...	
HwpAGrp2.dll	2010-02-12 ...	응용 프로그...	
HwpAGrp3.dll	2010-02-12 ...	응용 프로그...	
HwpApp.dll	2010-02-12 ...	응용 프로그...	

파일 설명: Hancom Office Hanword 2010  
회사: Hancom Inc(HNC).  
파일 버전: 8.0.0.466  
만든 날짜: 2010-02-12 오전 8:00  
크기: 4.13MB

0.1%	30.4MB	0.1MB/s	0Mbps
0.3%	29.0MB	0MB/s	0Mbps
0.5%	13.2MB	0.1MB/s	0Mbps
0%	1.2MB	0MB/s	0Mbps

# 프로세스

## 프로세스(process)

작업 관리자

파일(F) 옵션(O) 보기(V)

프로세스 성능 앱 기록 시작프로그램

이름

앱 (5)

- Google Chrome(32 bit)
- Hancom Office Hanword 2010...
- Microsoft PowerPoint(32 bit)
- Windows 탐색기
- 작업 관리자

백그라운드 프로세스 (84)

AcroTray(32 bit)

응용 프로그램 도구 Hwp80

파일 홈 공유 보기 관리

← → ↕ ↑ << Program Files (x86) > Hnc > Hwp80 > Hwp80 검색

로컬 디스크 (C:) 이름 수정한 날짜 유형 크기

- GPKI
- Intel
- PerfLogs
- Program Files
- Program Files (x86)
- Absolute Software
- Adobe

HncShellExt.dll 2010-02-12 ... 응용 프로그램...

HncShellExt64

Hwp.exe

HwpABase.dll

HwpAGrp1.dll

HwpAGrp2.dll

HwpAGrp3.dll

HwpApp.dll

작업 관리자

파일(F) 옵션(O) 보기(V)

프로세스 성능 앱 기록 시작프로그램 사용자 세부 정보

이름	PID	상태	사용자 ...	CPU
googledrivesync.exe	10204	실행 중	jwkim	00
googledrivesync.exe	6588	실행 중	jwkim	00
HimTrayIcon.exe	9880	실행 중	jwkim	00
Hwp.exe	2616	실행 중	jwkim	00
iBTAudioMon.exe	3500	실행 중	jwkim	00
iBTAudioSrv.exe	2260	실행 중	SYSTEM	00
ibtsiva.exe	2212	실행 중	SYSTEM	00
igfxCUIService.exe	1548	실행 중	SYSTEM	00
igfxEM.exe	7104	실행 중	jwkim	00
igfxext.exe	3956	실행 중	SYSTEM	00

이름	CPU	메모리	IO
Hwp.exe	0.1%	30.4MB	0MB/s
Windows 탐색기	0.3%	29.0MB	0MB/s
작업 관리자	0.5%	13.2MB	0MB/s
AcroTray(32 bit)	0%	1.2MB	0MB/s

# 프로세스

- 프로세스(process): 실행 중인 프로그램
  - 프로그램: 동작을 하지 않는 정적·수동적 개체
  - 프로세스: 동작을 하는 능동적 개체
- 운영체제로부터 자원을 할당 받아 동작
  - 자원: CPU, 메모리, 입출력장치, 파일 등
  - 동작: CPU가 프로세스의 명령을 실행
- 사용자 및 시스템 프로세스 존재

# 프로세스와 운영체제

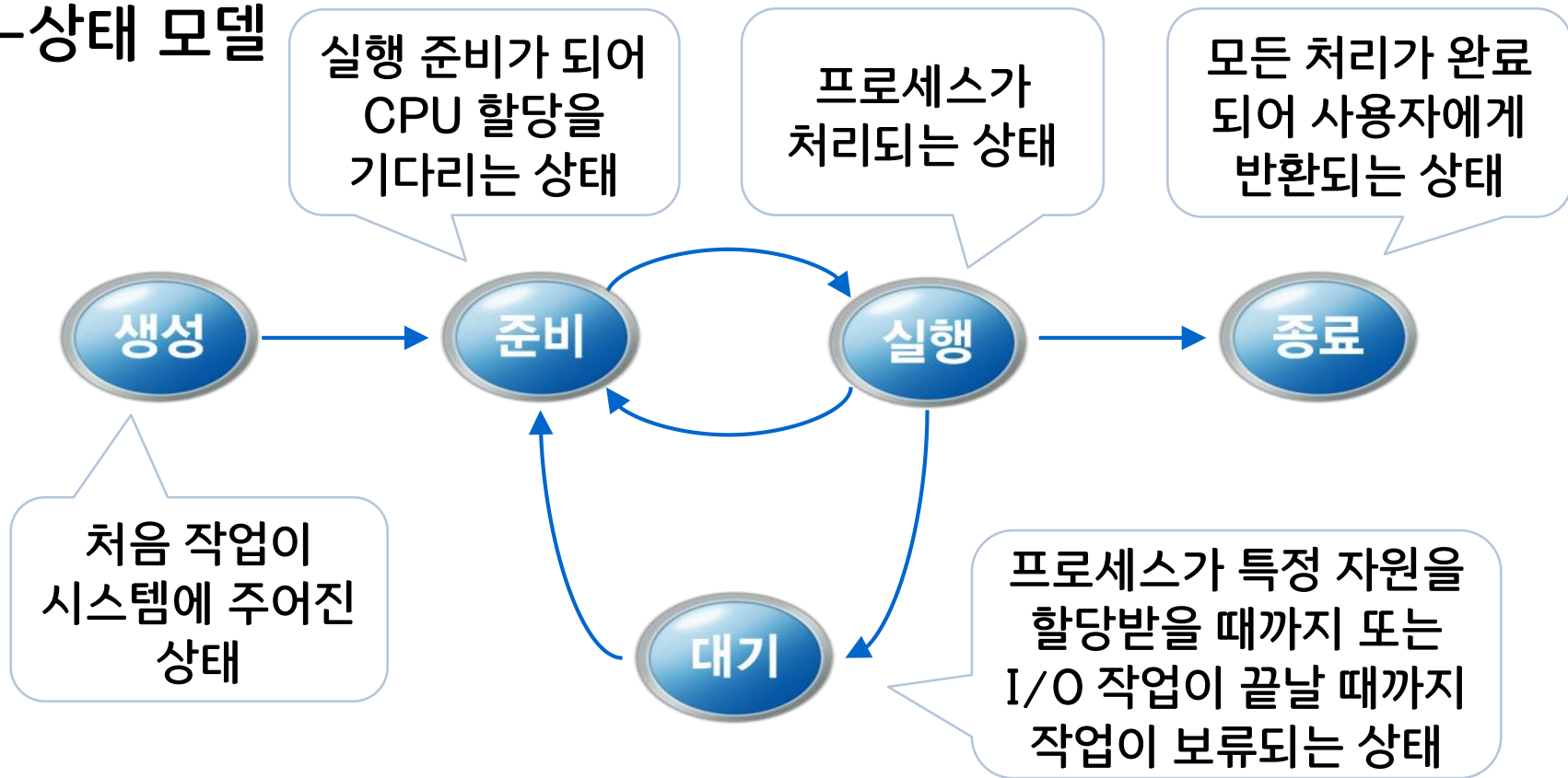
## ■ 프로세스 관리자의 역할

- 프로세스를 생성 및 삭제
- 프로세스 실행(CPU 할당)을 위한 스케줄 결정
- 프로세스의 상태를 관리하며 상태 전이를 처리



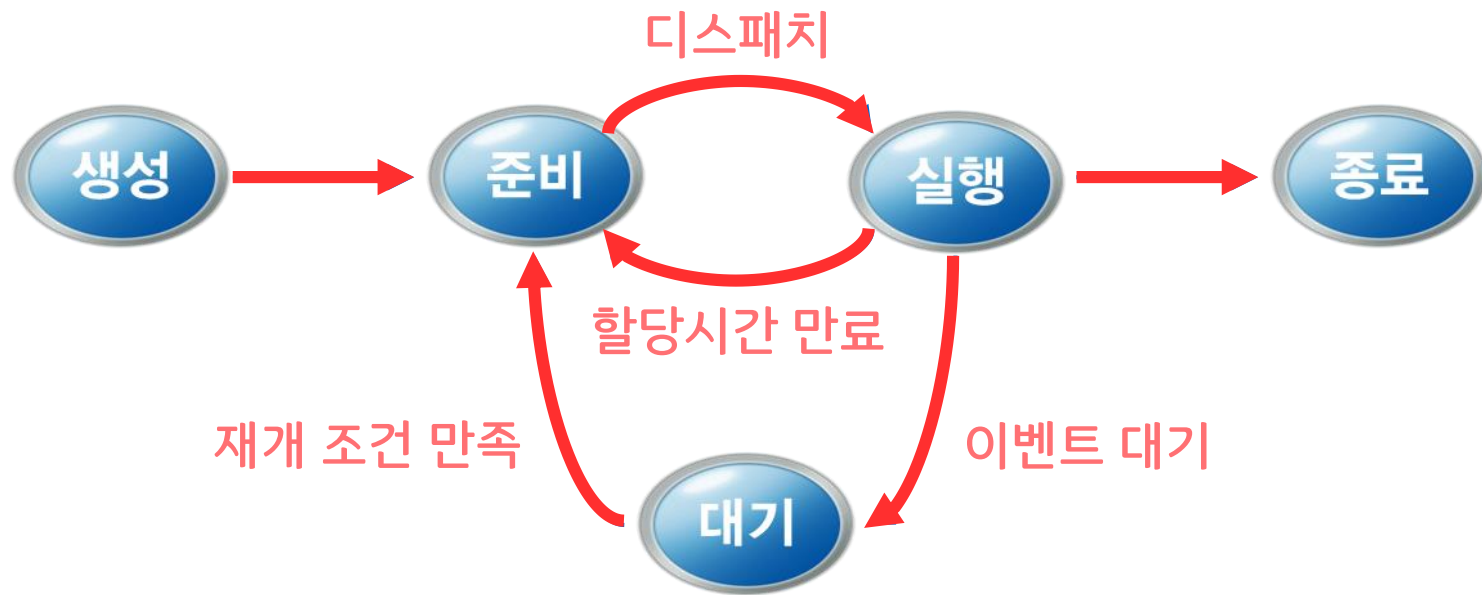
# 프로세스의 상태

## 5-상태 모델



# 프로세스의 상태 변화

## 5-상태 모델



# 프로세스 제어 블록(PCB)

## ■ 프로세스 제어 블록(Process Control Block, PCB)

- 프로세스의 관리를 위한 목적
- 프로세스의 정보를 보관
- 각 프로세스마다 존재
- 프로세스가 진행함에 따라 내용 변경

프로세스 상태
프로세스 번호(PID)
프로그램 카운터(PC)
레지스터들
메모리 관리 정보
프로세스 우선순위
회계 정보
⋮

# 프로세스의 생성과 종료

## ■ 프로세스 생성 작업

- 프로세스의 이름(번호, PID) 결정
- 준비 큐에 삽입
- 초기 우선순위 부여
- 프로세스 제어 블록(PCB) 생성 등

## ■ 프로세스 생성 방법: 시스템 호출

# 프로세스의 생성과 종료

## ■ 프로세스 생성 시스템 호출

- 하나의 프로세스가 프로세스 생성 시스템 호출을 통해 새로운 프로세스를 생성
- 예: `fork( )`
- 호출하는 프로세스: 부모 프로세스
- 생성되는 프로세스: 자식 프로세스

## ■ 시스템 프로세스와 사용자 프로세스 모두 부모 프로세스 가능

## 프로세스의 생성과 종료

### ■ 생성되는 프로세스의 자원

- 운영체제로부터 직접 얻는 경우
- 부모 프로세스 자원의 일부를 얻는 경우

### ■ 자식 프로세스의 자원은 부모 프로세스의 자원으로 제한

- 과도한 자식 프로세스 생성에 따른 시스템 과부하 방지

# 프로세스의 생성과 종료

## ■ 프로세스 종료

- 프로세스의 마지막 명령이 실행을 마치는 경우
- 프로세스 종료 시스템 호출(예: `exit( )`)을 통하는 경우
- 프로세스 종료 후 부모 프로세스에게 실행결과를 되돌려 줌

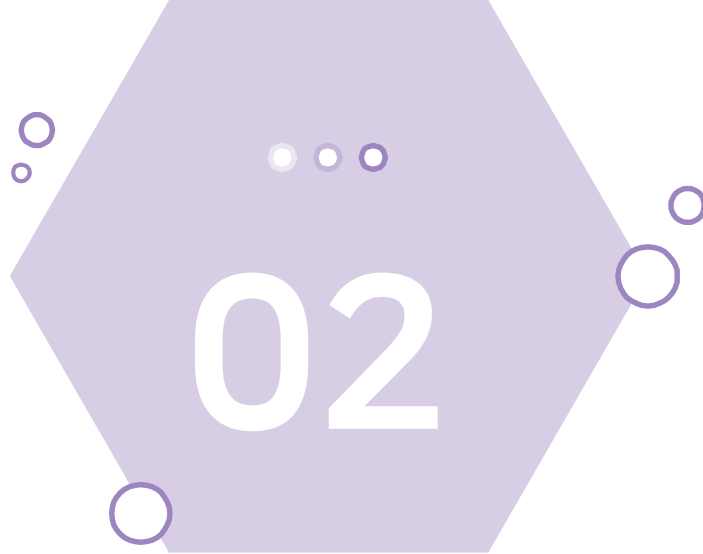
## ■ 프로세스 종료 시스템 호출

- 부모에 의해서만 호출
- 자식 프로세스가 할당된 자원의 사용을 초과할 때 혹은 더 이상 필요치 않을 때

# 프로세스 간의 관계

	독립적 프로세스	유기적 프로세스
의미	다른 프로세스의 영향을 받지 않고 주지도 않음	다른 프로세스와 영향을 주고받음
프로세스 상태	다른 프로세스와 공유하지 않음	다른 프로세스와 공유함
실행	결정적, 재생 가능	비결정적, 재생 불가능
	다른 프로세스와 무관하게 중단 및 재시작 가능	
데이터	다른 프로세스와 공유하지 않음	다른 프로세스와 공유함





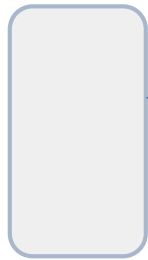
# 쓰레드

# 프로세스와 쓰레드

## ■ 전통적인 프로세스

- 처리의 기본 단위
- 자원 소유의 단위(하나의 주소공간) 및 디스패칭의 단위(하나의 제어흐름)
- 단일 프로세스 내에서 동시처리 불가능 → 쓰레드 등장

프로세스

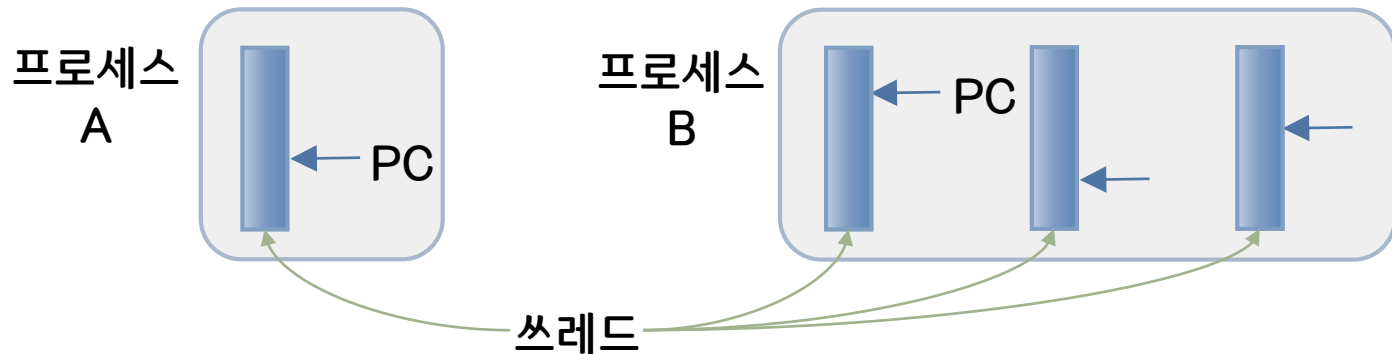


← 하나의 실행점  
(PC)

# 프로세스와 스레드

## ■ 스레드(Thread)

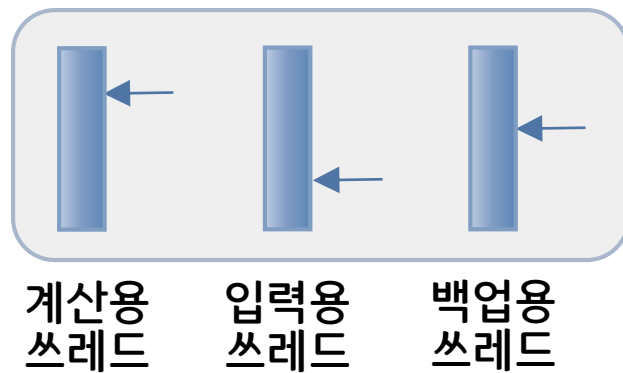
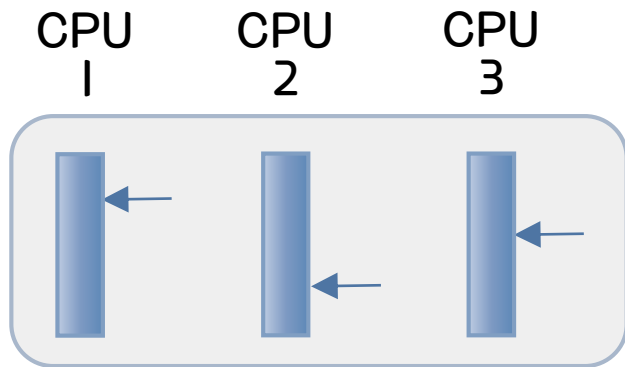
- 프로세스 내에서의 다중처리를 위해 제안된 개념
- 하나의 프로세스 내에는 하나 이상의 스레드가 존재
- 하나의 스레드 내에서는 하나의 실행점만 존재 (**디스패칭의 단위**)
- 실행에 필요한 최소한의 정보만을 가지며,  
자신이 속해 있는 프로세스의 실행환경을 공유

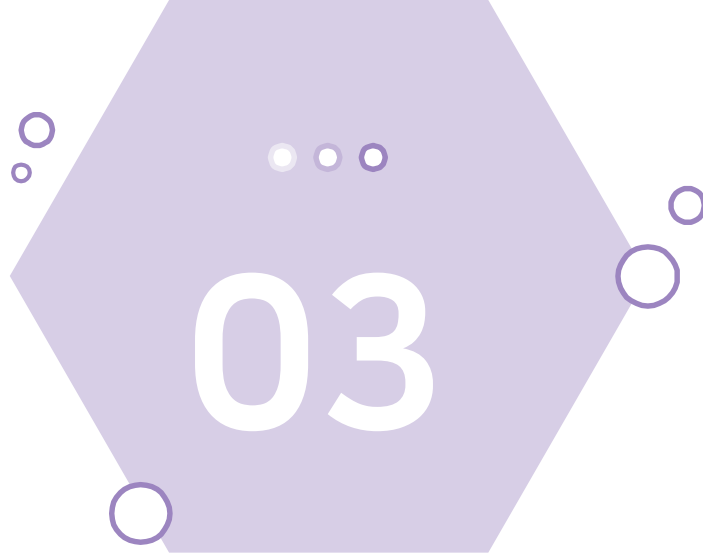


# 쓰레드

## ■ 다중 쓰레드의 장점

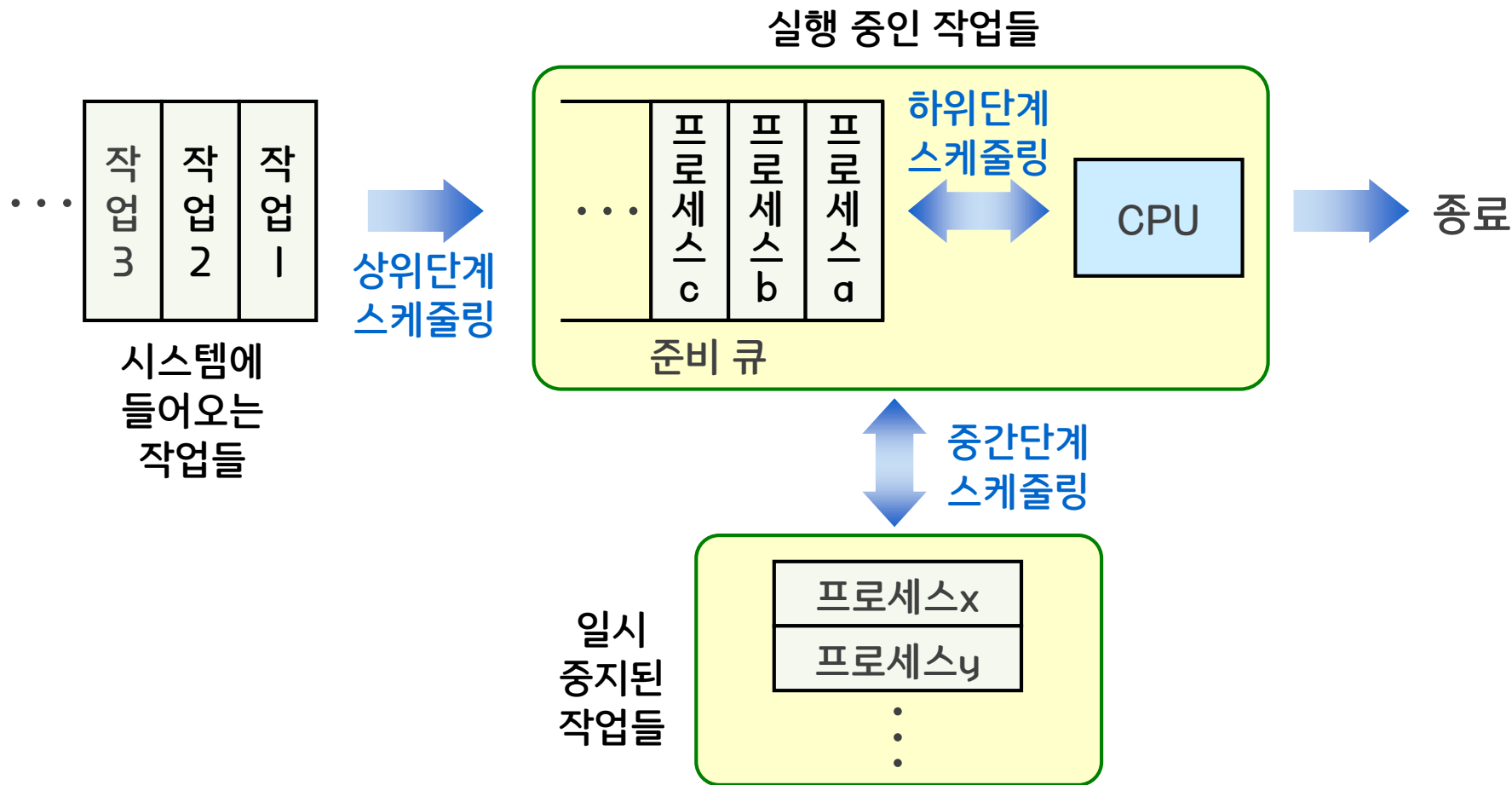
- 멀티CPU 혹은 멀티코어 시스템에서는 병렬처리 가능
- 처리 속도 별로 쓰레드가 나뉘진 경우 효율적인 처리 가능





# 스케줄링

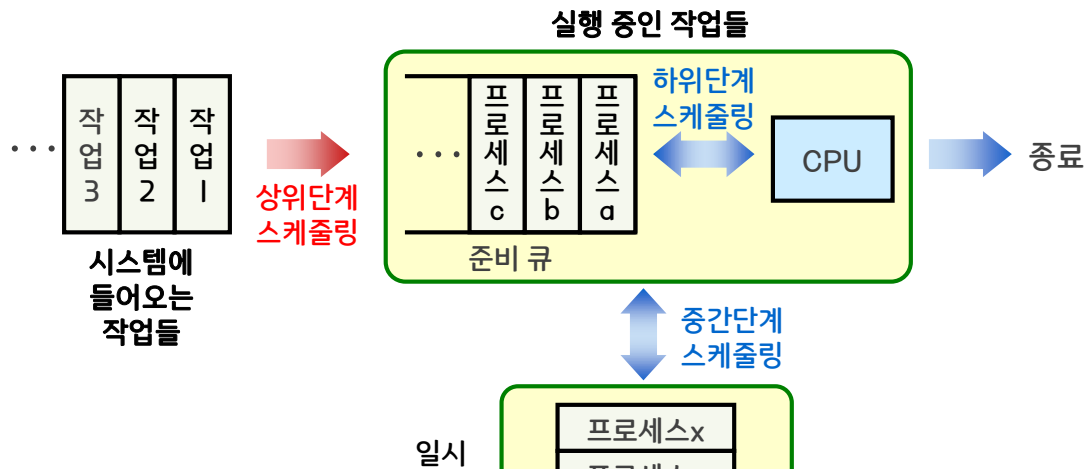
# 스케줄링 단계



# 스케줄링 단계

## ■ 상위단계 스케줄링

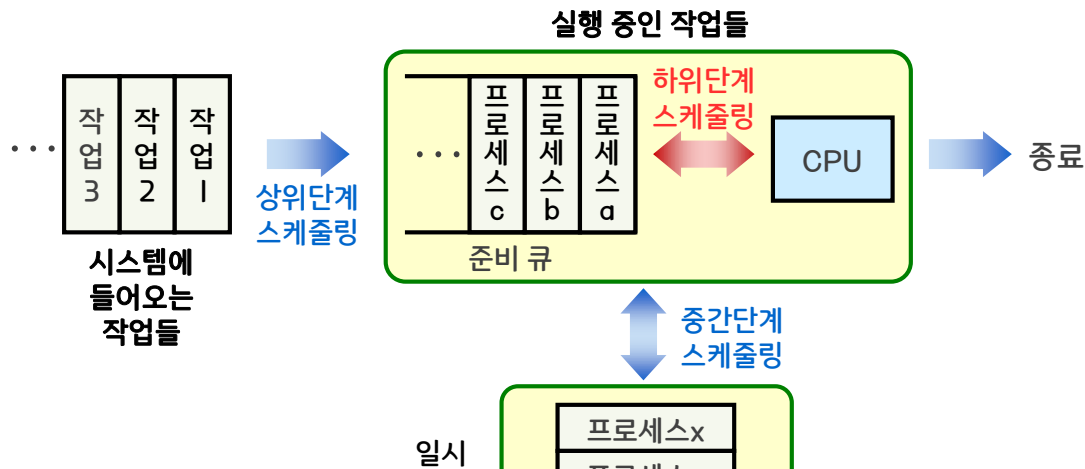
- 시스템에 들어오는 작업들을 선택하여 프로세스를 생성한 후 프로세스 준비 큐에 전달
- 선택 기준: 시스템의 자원을 효율적으로 이용할 수 있도록 하는 것
- 입출력(I/O) 중심 작업과 연산 중심 작업을 균형있게 선택



# 스케줄링 단계

## 하위단계 스케줄링

- 사용 가능한 CPU를 준비상태의 어느 프로세스에게 배당할지를 결정
- CPU를 배당받은 프로세스는 결국 실행상태가 되어 프로세스가 처리됨
- 수행 주체: 디스패처(dispatcher)

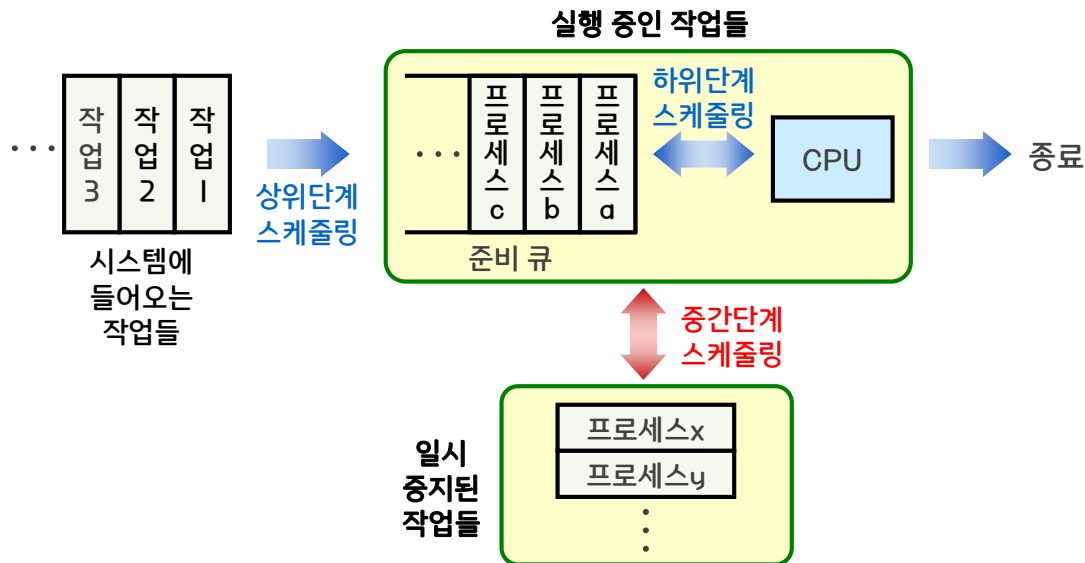




# 스케줄링 단계

## ■ 중간단계 스케줄링

- 프로세스를 일시적으로 메모리에서 제거하여 중지시키거나 다시 활성화시킴
- 시스템에 대한 단기적인 부하를 조절



# 스케줄링 정책

## 스케줄링 기본 목표

- 공정성: 모든 프로세스가 적정 수준에서 CPU 작업을 할 수 있게 함
- 균형: 시스템의 자원들이 충분히 활용될 수 있게 함

### ★ 일괄처리 운영체제

처리량의 극대화  
반환시간의 최소화  
CPU 활용의 극대화

### ★ 대화형 운영체제

빠른 응답시간  
과다 대기시간 방지

### ★ 실시간 운영체제

처리 기한을 맞춤

# 스케줄링 정책

## ■ 선점(Preemptive) 스케줄링 정책

- 진행 중인 프로세스에 인터럽트를 걸고 다른 프로세스에 CPU를 할당하는 스케줄링 전략
- 높은 우선순위의 프로세스를 긴급하게 처리하는 경우에 유용
- 대화식 시분할 시스템에서 빠른 응답시간을 유지하는데 유용
- 문맥 교환에 따른 오버헤드 발생

### ★ 문맥

CPU의 모든 레지스터와 기타 운영체제에 따라 요구되는 프로세스의 상태

# 스케줄링 정책

## ■ 선점(Preemptive) 스케줄링 정책

- 진행 중인 프로세스에 인터럽트를 걸고 다른 프로세스에 CPU를 할당하는 스케줄링 전략
- 높은 우선순위의 프로세스를 긴급하게 처리하는 경우에 유용
- 대화식 시분할 시스템에서 빠른 응답시간을 유지하는데 유용
- 문맥 교환에 따른 오버헤드 발생

### ★ 문맥 교환

CPU의 현재 실행하고 있는 프로세스의 문맥을 PCB에 저장하고 다음 프로세스의 PCB로부터 문맥을 복원하는 작업

# 스케줄링 정책

## ■ 비선점(Nonpreemptive) 스케줄링 정책

- 프로세스가 CPU를 할당받아 실행이 시작되면  
작업 자체가 I/O 인터럽트를 걸거나 작업을 종료할 때까지  
실행상태에 있게 됨
- 모든 프로세스가 공정하게 순서에 따라 실행됨 → 응답시간 예측 가능
- 짧은 프로세스가 긴 프로세스를 기다리게 될 수 있음



강의를 마쳤습니다.

다음시간에는  
**3강. 스케줄링 알고리즘**