

본 강의에서 수업자료로 이용되는 저작물은  
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,  
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.  
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로  
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

2023. 3 . 02 .

부천대학교·한국복제전송저작권협회

# 운 영 체 제

## 2장 프로세스 관리(1)

# 학습 내용

## \* 2장 프로세스 관리

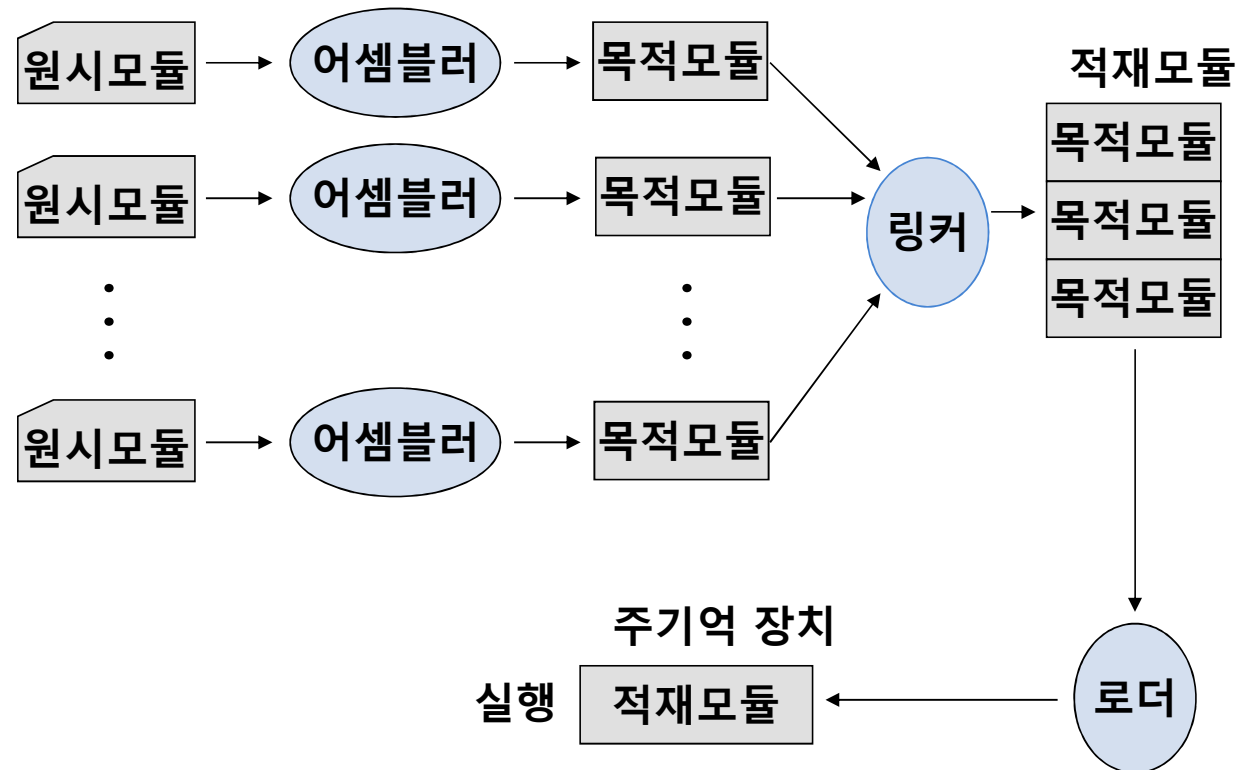
- \* 시스템 소프트웨어의 종류와 기능 <= 2장 프로세스 관리와 5장 메모리 관리 관련
  - \* 컴파일러와 인터프리터
  - \* 어셈블리어와 어셈블러
  - \* 링커와 로더
- \* 프로세스 개요
  - \* 명령어 실행 과정
  - \* 프로세스 개념
  - \* 프로세스 정의
  - \* 프로세스 제어 블록(PCB)
  - \* 프로세스의 상태
  - \* 프로세스 상태의 전이
  - \* 프로세스 상태의 전이 정리
- \* 프로세스 스케줄링 개요
  - \* 프로세스 스케줄링 개요
  - \* 비선점 스케줄링
  - \* 선점 스케줄

## 2장 프로세스 관리(1)

시스템 소프트웨어의 종류와 기능  
프로세스 개요  
프로세스 스케줄링 개요

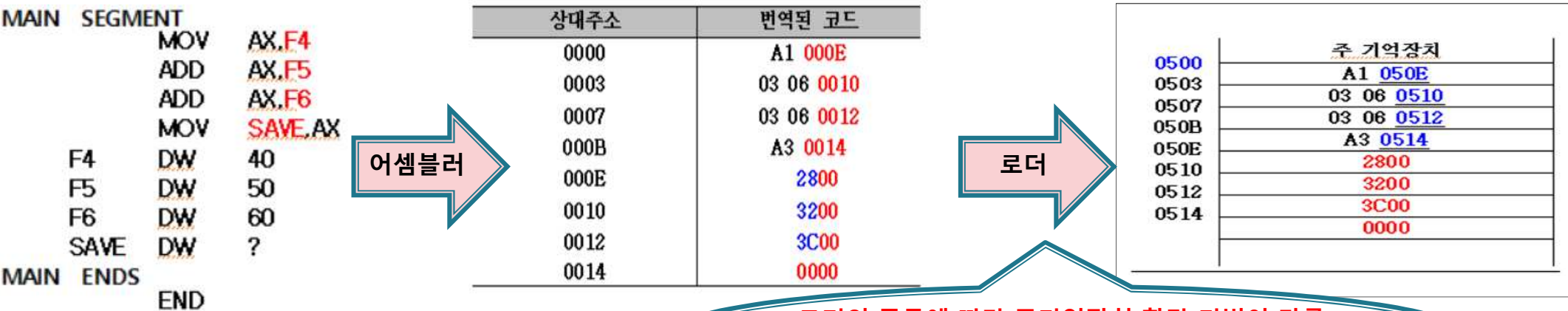
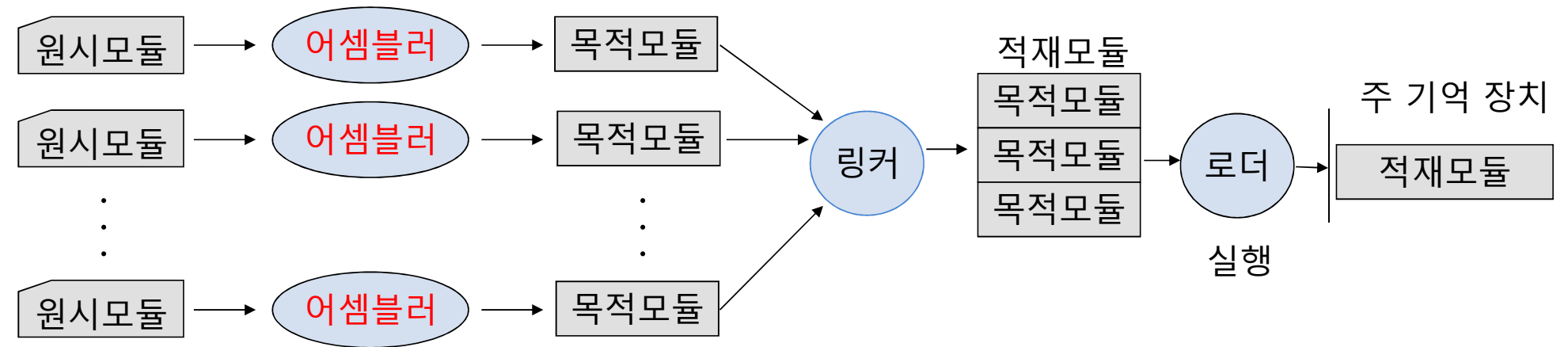
# 시스템 프로그램의 종류

- \* 컴파일러와 인터프리터
- \* 어셈블리어와 어셈블러
- \* 매크로와 매크로프로세서
- \* 링커와 로더
- \* 운영체제



# 어셈블리어와 어셈블러

▶ 어셈블리 언어를 이용한 프로그램 개발 과정



로더의 종류에 따라 주기억장치 할당 기법이 다름  
=> 운영체제의 4장 메모리 관리

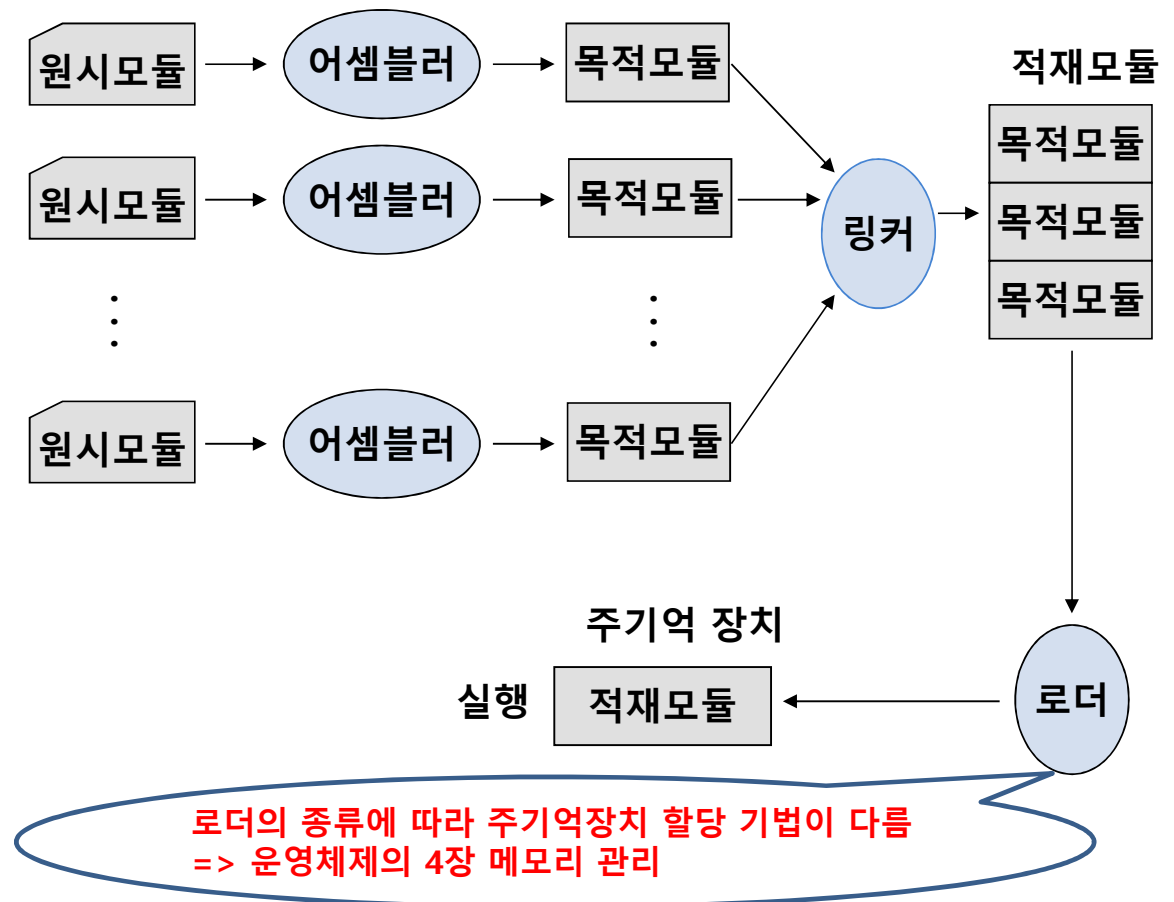
# 링커와 로더 (1/3)

## ❖ 링커

- 여러 개로 나누어진 원시 모듈들은 어셈블러에 의해 별도로 어셈블되어 각각 다른 목적 모듈들로 만들어져, 보조 기억 장치에 저장된다. 이 모듈들을 실행할 때는 연결해 주는 링커로 연결

## ❖ 로더

- 목적프로그램을 주기억장치에 적재
- 프로그램의 실행 준비
- 기억장소를 적게 점유
- 절대적 적재와 재배치 적재



# 링커와 로더 (2/3)

## 링커

- 언어 번역 프로그램이 생성한 목적 프로그램들과 라이브러리, 또 다른 실행 프로그램(로드 모듈) 등을 연결하여 실행 가능한 로드 모듈을 만드는 시스템 소프트웨어로 Linkage Editor라고도 한다.
- 연결 기능만 수행하는 로더의 한 형태로, 링커에 의해 수행되는 작업을 링킹(Linking)이라 한다.

## 로더

정의	컴퓨터 내부로 정보를 들어오거나 로드 모듈을 디스크 등의 보조기억장치로부터 주기억장치에 적재하는 시스템 소프트웨어
기능	<ul style="list-style-type: none"><li>할당(Allocation) : 실행 프로그램을 실행시키기 위해 기억장치 내에 옮겨놓을 공간을 확보하는 기능</li><li>연결(Linking) : 부 프로그램 호출 시 그 부 프로그램이 할당된 기억장소의 시작주소를 호출한 부분에 등록하여 연결하는 기능</li><li>재배치(Relocation) : 디스크 등의 보조기억장치에 저장된 프로그램이 사용하는 각 주소들을 할당된 기억장소의 실제 주소로 배치시키는 기능</li><li>적재&gt;Loading) : 실행 프로그램을 할당된 기억공간에 실제로 옮기는 기능</li></ul>
종류	<ul style="list-style-type: none"><li>Compile And Go 로더 : 별도의 로더 없이 언어 번역 프로그램이 로더의 기능까지 수행하는 방식(할당, 재배치, 적재 작업을 모두 언어 번역 프로그램이 담당)</li><li>절대 로더(Absolute Loader) : 목적 프로그램을 기억 장소에 적재시키는 기능만 수행하는 로더(할당 및 연결은 프로그래머가, 재배치는 언어 번역 프로그램이 담당)</li><li>직접 연결 로더(Direct Linking Loader) : 일반적인 기능의 로더로, 로더의 기본 기능 4가지를 모두 수행하는 로더</li><li>동적 적재 로더(Dynamic Loading Loader) : 프로그램을 한꺼번에 적재하는 것이 아니라 실행 시 필요한 일부분만을 적재하는 로더</li></ul>



## 링커와 로더 (3/3)

### ❖ 로더의 종류

(1) 컴파일 즉시 로더(Compile-and-go loader)

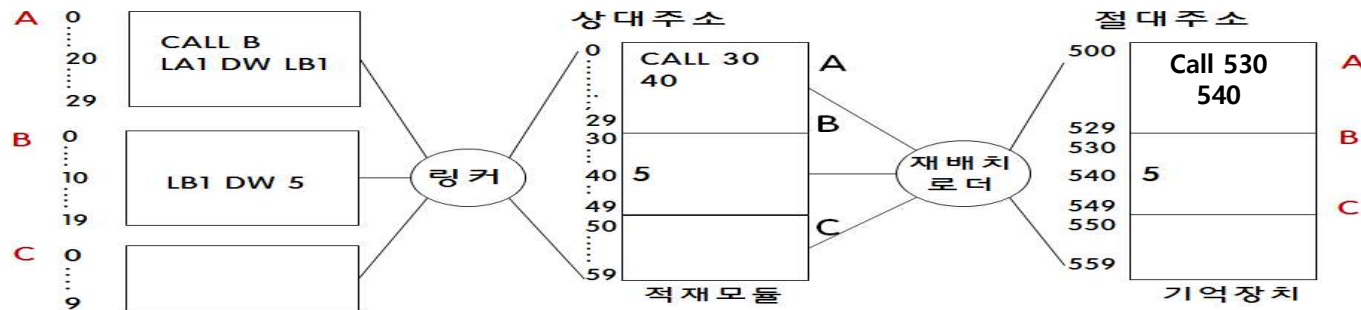
① 링크 : 프로그래머 ② 할당 : 프로그래머 ③ 재배치 : 어셈블러 ④ 적재 : 어셈블러

(2) 절대 로더(absolute loader)

① 링크 : 프로그래머 ② 할당 : 프로그래머 ③ 재배치 : 어셈블러 ④ 적재 : 로더

(3) 링커 및 재배치 로더(linker and relocate loader)

① 링크 : 링커 ② 할당 : 운영체제 ③ 재배치 : 로더 ④ 적재 : 로더



(4) 링킹 로더 (direct linking loader) : 링크 및 적재 과정을 한꺼번에 수행

① 링크 : 링킹로더 ② 할당 : 운영체제 ③ 재배치 : 링킹로더 ④ 적재 : 링킹로더

(5) 동적 로더 (dynamic loader)

① 링크 : 동적로더 ② 할당 : 운영체제 ③ 재배치 : 동적로더 ④ 적재 : 동적로더

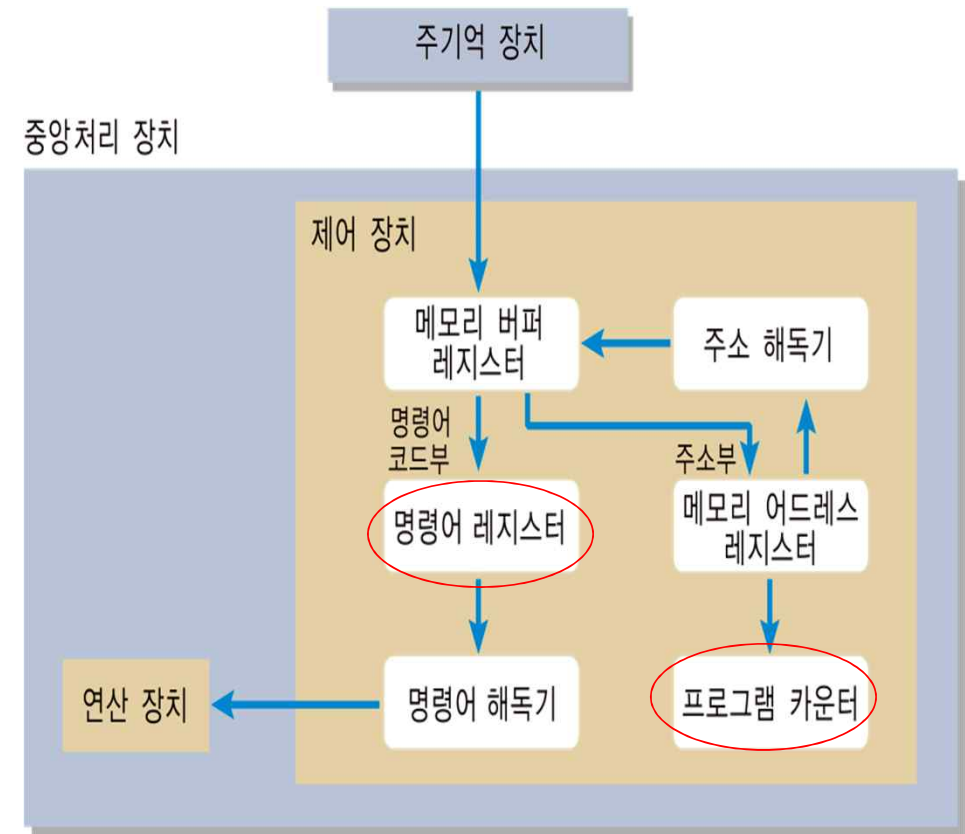
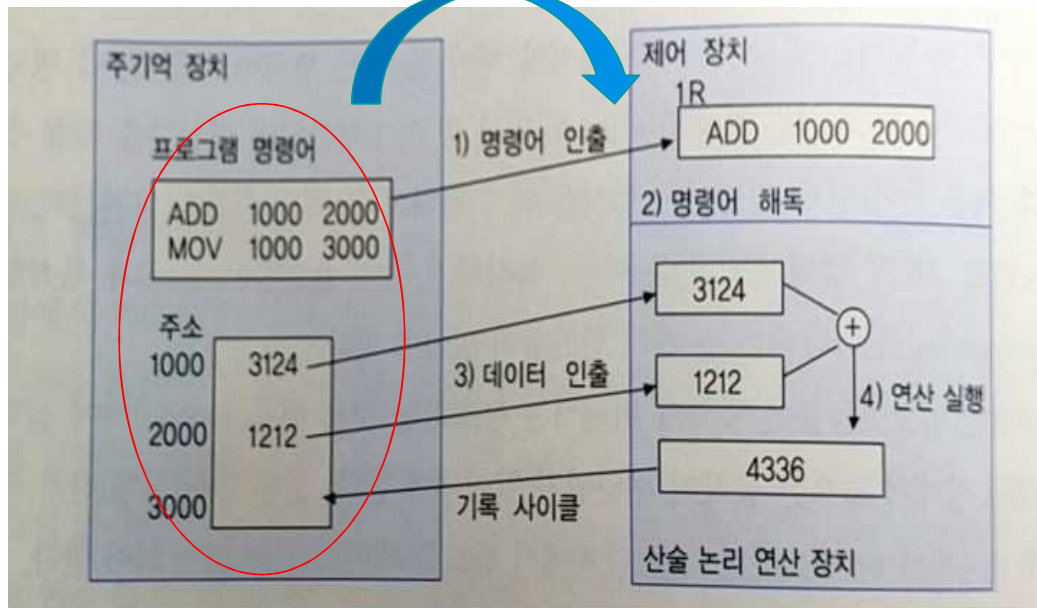
## 2장 프로세스 관리(1)

시스템 소프트웨어의 종류와 기능  
프로세스 개요  
프로세스 스케줄링 개요

# 명령어 실행 과정

- 컴퓨터 동작 ( 예: 명령어 실행 과정 )
  1. 명령어 인출(메모리로부터 명령어 가져오기)
  2. 명령어 해독
  3. 데이터 인출
  4. 실행

OS  
-프로세서 스케줄러  
-디스패처



IR(명령어 레지스터), PC(프로그램 카운터), MBR(메모리 버퍼 레지스터), MAR(메모리 어드레스 레지스터)

# 명령어 실행 과정

[컴퓨터 동작 (예: 명령어 실행 과정 )]

## 1. 명령어 인출

- PC의 내용을 MAR로 내보낸다.
- PC가 1 증가되어 다음 명령어를 지시한다.
- MAR에 해당되는 주소의 메모리 내용이 MBR로 입력된다.
- MBR의 내용을 IR로 전달한다.

## 2. 명령어 해독

## 3. 데이터 인출

- 해석된 명령어 종류에 따라 실행하기 위해 오퍼랜드 준비

## 4. 명령어 실행

=> 명령어 [실행] 사이클(Instruction Cycle)

=> 마이크로 연산(Micro Operation)

[레지스터]

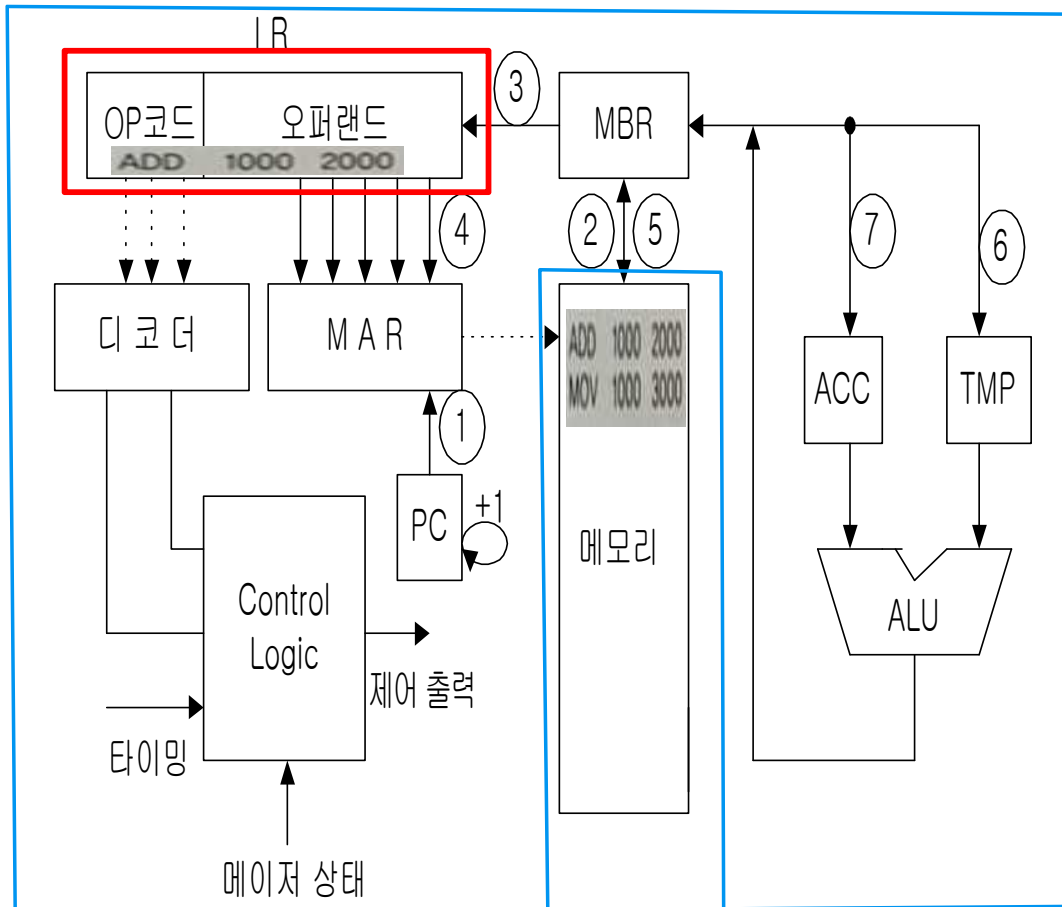
PC : Program Counter

IR : Instruction Register

MAR : Memory Address Register

MBR : Memory Buffer Register

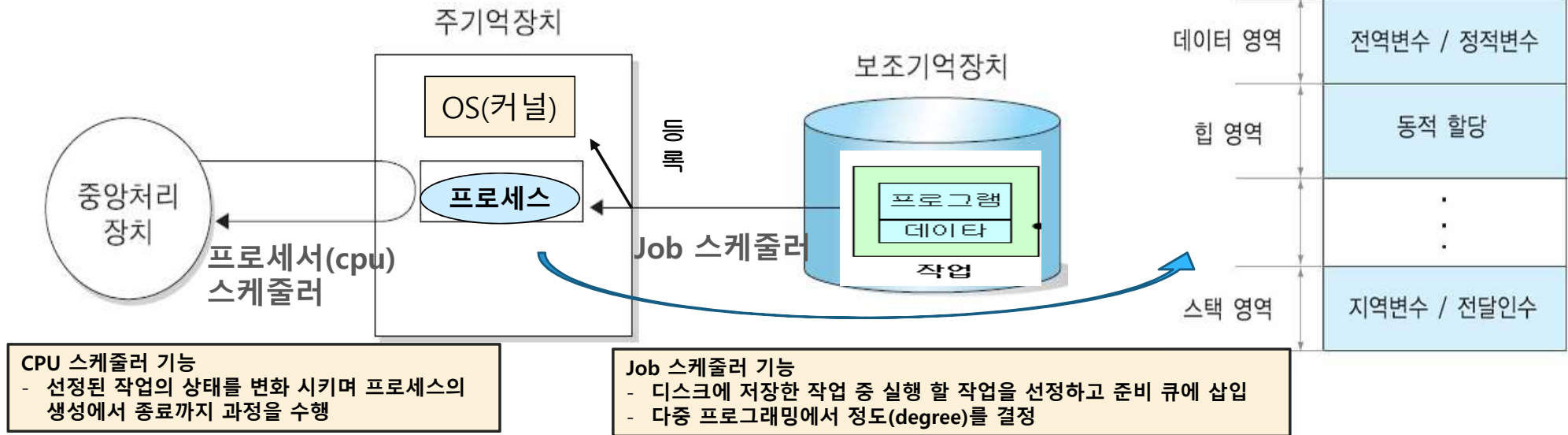
ACC : Accumulator



# 프로세스 개념

- \* 작업 (job)
  - \* 실행 프로그램과 이에 필요한 입력 데이터의 묶음
  - \* 컴퓨터에 실행 의뢰되기 전의 상태
- \* 프로세스 (process)
  - \* 임의의 프로그램이 실행되기 위해서는 반드시 실행되기 전에 주기억장치에 저장되어야 한다.
  - \* 커널에 등록된 작업
  - \* 커널에 등록되어 커널의 관리하에 들어간 작업
  - \* 운영체제에 의해 관리 됨

- \* 프로세스 4가지 구성 요소
  - \* 코드(code) 영역
  - \* 데이터 영역
  - \* 스택(stack) 영역
  - \* 힙 영역



# 프로세스 정의

## \* 프로세스 정의

- \* 실행중인 프로그램 (작업), PCB를 가진 프로그램, 주기억장치에 저장된 프로그램
- \* 프로세서가 할당되는 실체 => PC(Program Counter)레지스터를 지닌 프로그램
- \* 커널에 등록되고 커널의 관리하에 있는 작업
- \* 운영체제가 관리하는 최소 실행 단위
- \* 지정된 결과를 얻기 위한 일련의 계통적 동작
- \* 목적 또는 결과에 따라 발생하는 사건들의 과정
- \* 능동적인 개체로 순차적으로 수행하는 프로그램(비동기적 행위를 일으키는 주체 )
  - \* 실행 중에 각종 자원을 요구, 할당, 반납하며 진행
- 프로세스 제어 블록을 할당받은 개체

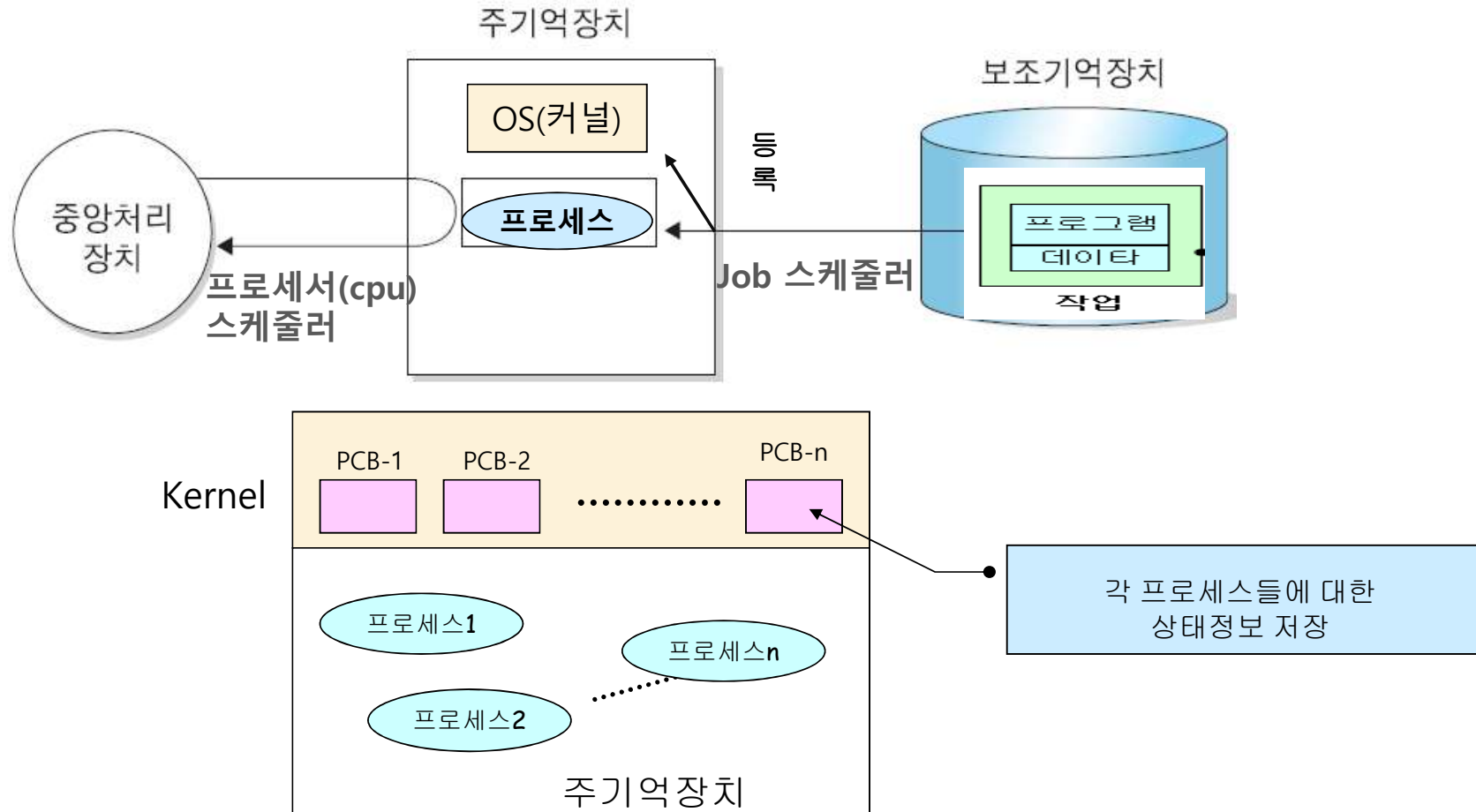
## \* 관련 용어

- \* 프로세스 제어 블록 (PCB : Process Control Block)
  - \* 커널 공간 (kernel space) 내에 존재
  - \* 커널에 등록된 각 프로세스들에 대한 정보를 저장하는 영역
- \* 컴퓨터 시스템의 상태 모델링
  - \* 각종 프로세스들과 각종 자원들의 상호작용(interaction)

# 프로세스 제어 블록(PCB) (1/2)

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

- PCB는 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓은 테이블



# 프로세스 제어 블록(PCB) (2/2)

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

- PCB는 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓은 테이블로 task control block, job control block 이라고도 한다.
- PCB 저장을 위한 공간은 시스템이 최대로 수용할 수 있는 프로세스의 수를 기본으로 하여 동적으로 할당한다.
- 각 프로세스가 생성될 때 마다 고유의 PCB가 생성되고 , 프로세스가 완료되면 PCB는 제거된다.

## ❖ PCB에 저장된 정보

- 프로세스의 현재 상태(실행, 준비완료, 대기 등)
- 프로세스의 고유 이름(identifier)
- 프로세스의 우선순위 및 스케줄링
- 중앙처리장치의 각종 레지스터 상태를 저장하기 위한 공간  
(프로그램 카운터, 인덱스 레지스터, 누산기 등)
- 주기억장치 관리 정보
- 입출력 상태 정보

- ◆ PCB 에 저장되어야 할 정보는 운영체제에 따라 서로 다름
- ◆ 커널의 PCB 영역 참조 및 갱신 속도 개선은 시스템 전체 성능에 중요함

프로세스 식별자
프로세스 상태
프로그램 카운터
레지스터 저장 영역
프로세서 스케줄링 정보
계정 정보
입출력 상태 정보 메모리 관리 정보 ...

프로세스 제어 블록(PCB)



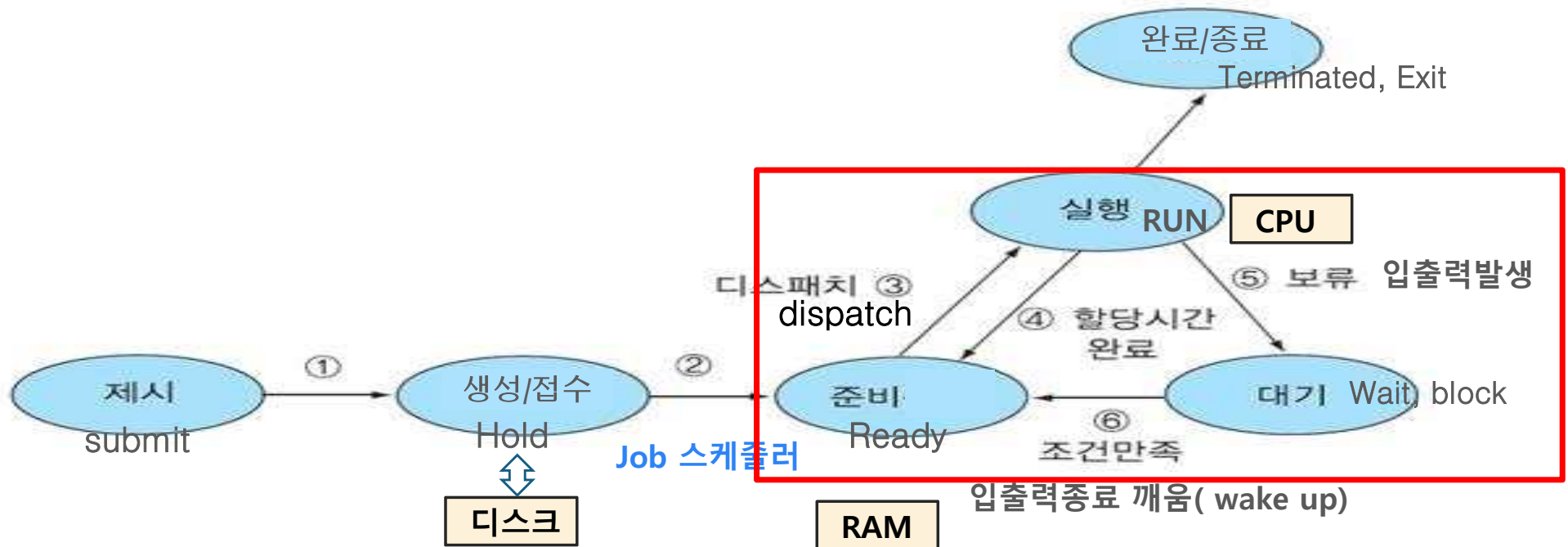
# 프로세스 상태(1/2)

- ❖ 프로세스 상태는 다음과 같이 제시, 생성, 준비, 실행, 대기 상태 5가지이고 이 중 주요 3가지 상태는 준비, 실행, 대기 상태이다.

준비, 실행, 대기 상태는 활동 상태로 주기억 장치를 할당 받은 상태이다.

- ❖ 프로세스 상태의 전이

프로세스 상태 전이는 프로세스가 시스템 내에 존재하는 동안 프로세스의 상태가 변하는 것을 의미하며 하나의 프로세스는 완료될 때까지는 여러 상태의 변화를 거친다.



## 프로세스 상태 (2/2)

❖ 프로세스 상태는 다음과 같이 5가지이고 이 중 주요 3가지 상태는 준비, 실행, 대기 상태이다.

**준비, 실행, 대기 상태는 활동 상태로 주기억 장치를 할당 받은 상태이다.**

❖ 제출 (submit)

- 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태

❖ 접수 (hold)

- 제출된 작업이 스푼 공간인 디스크의 할당 위치에 저장된 상태

❖ 준비(ready)

- 프로세스가 프로세서를 할당받기 위해 기다리고 있는 상태
- 프로세스는 준비상태 큐에서 실행을 준비하고 있다.
- 접수 상태에서 준비 상태로의 전이는 job스케줄러에 의해 수행된다.

❖ 실행(run)

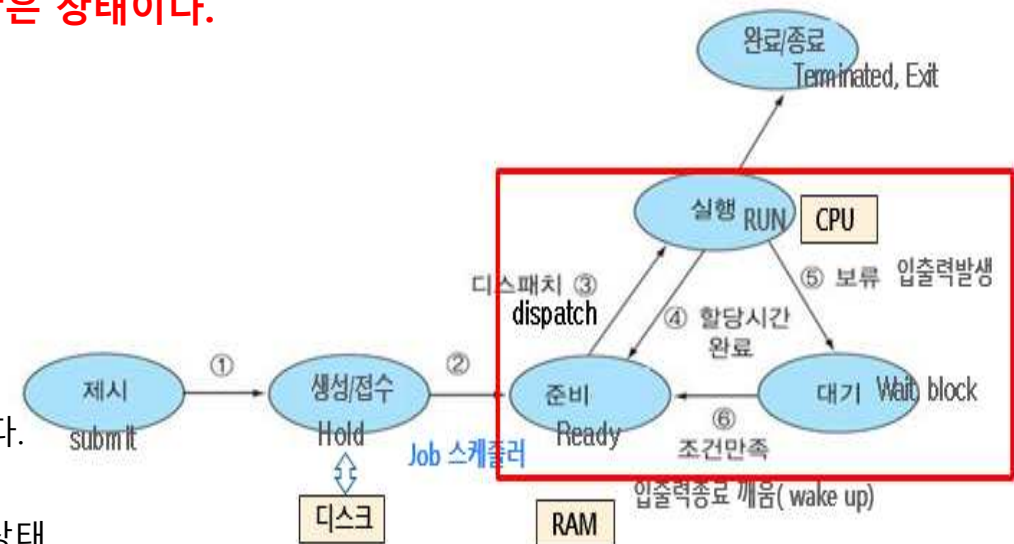
- 준비상태 큐에 있는 프로세스가 프로세서를 할당받아 실행되는 상태
- 프로세스 수행이 완료되기 전에 프로세스에게 주어진 프로세서 할당 시간이 종료되면 프로세스는 준비상태로 전이된다.
- 실행중인 프로세스에 입/출력 처리가 필요하면 실행중인 프로세스는 대기 상태로 전이된다.

❖ 대기, 보류, 블록(wait, block)

- 프로세스에 입/출력 처리가 필요하며 현재 실행중인 프로세스가 중단되고 입/출력 처리가 완료될때 까지 대기하고 있는 상태

❖ 종료(terminated, Exit)

- 프로세서의 실행이 끝나고 프로세스 할당이 해제된 상태



# 프로세스 상태와 전이(1/5)

- ❖ 프로세스 상태는 다음과 같이 제시, 생성, 준비, 실행, 대기 상태 5가지이고 이 중 주요 3가지 상태는 준비, 실행, 대기 상태이다.  
준비, 실행, 대기 상태는 활동 상태로 주기억 장치를 할당 받은 상태이다.

- ❖ 프로세스 상태의 전이

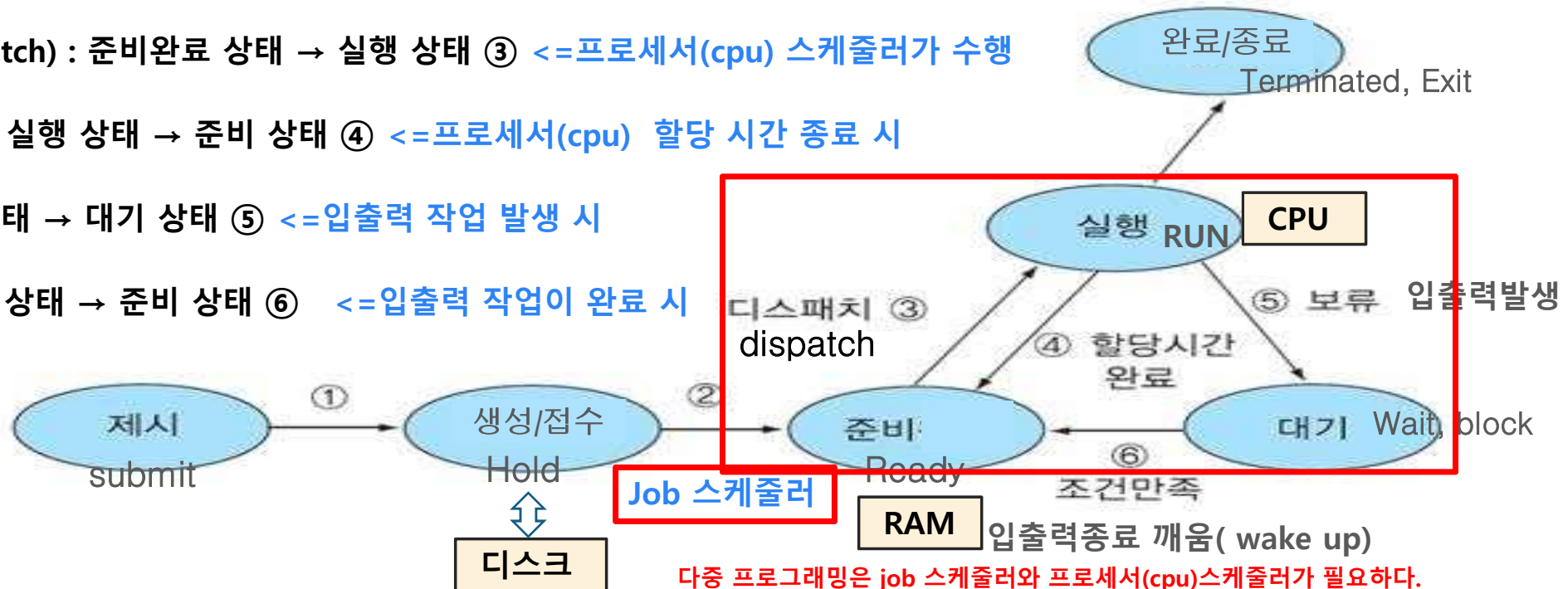
프로세스 상태 전이는 프로세스가 시스템 내에 존재하는 동안 프로세스의 상태가 변하는 것을 의미하며 하나의 프로세스는 완료될 때까지는 여러 상태의 변화를 거친다.

디스패치(dispatch) : 준비완료 상태 → 실행 상태 ③ <=프로세서(cpu) 스케줄러가 수행

timer runout : 실행 상태 → 준비 상태 ④ <=프로세서(cpu) 할당 시간 종료 시

block : 실행 상태 → 대기 상태 ⑤ <=입출력 작업 발생 시

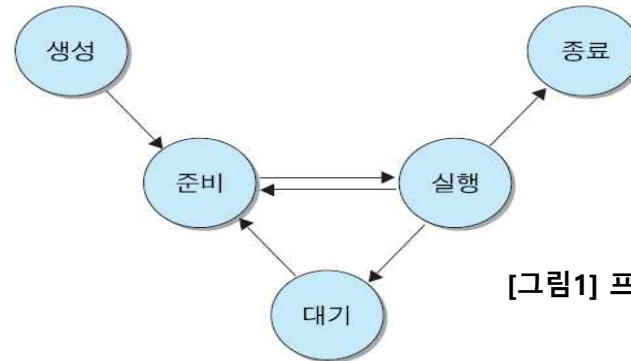
wakeup : 대기 상태 → 준비 상태 ⑥ <=입출력 작업이 완료 시



다중 프로그래밍은 job 스케줄러와 프로세서(cpu)스케줄러가 필요하다.

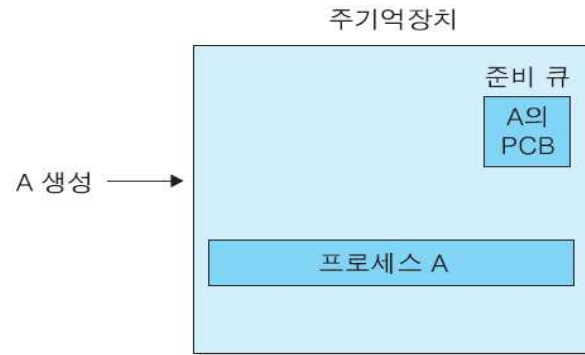
# 프로세스 상태의 전이 (2/5)

## ❖ 프로세스의 상태

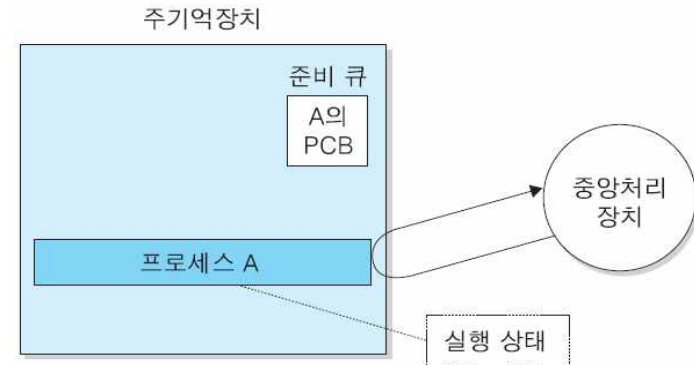


[그림1] 프로세스의 상태

## ▶ 새로운 프로세스 A 생성



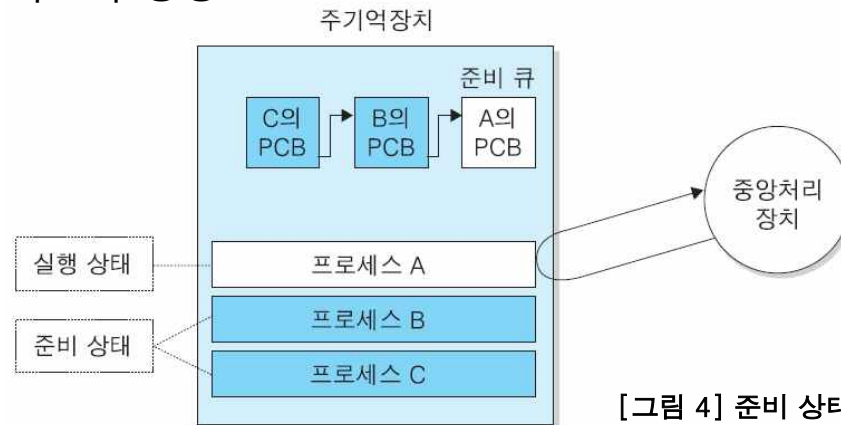
[그림2] 생성된 프로세스의 프로세스 제어 블록이 준비큐에 연결



[그림 3] 중앙처리장치가 프로세스 A를 실행

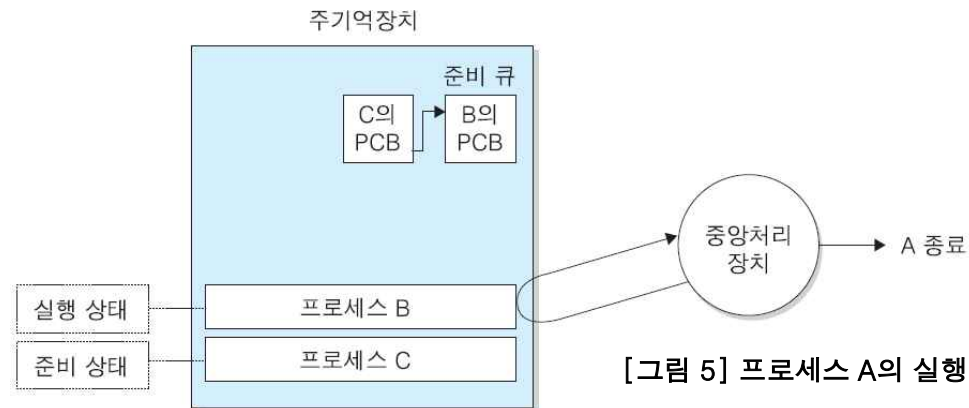
# 프로세스 상태의 전이 (3/5)

- 프로세스 B와 C가 생성



[그림 4] 준비 상태인 프로세스 B와 C

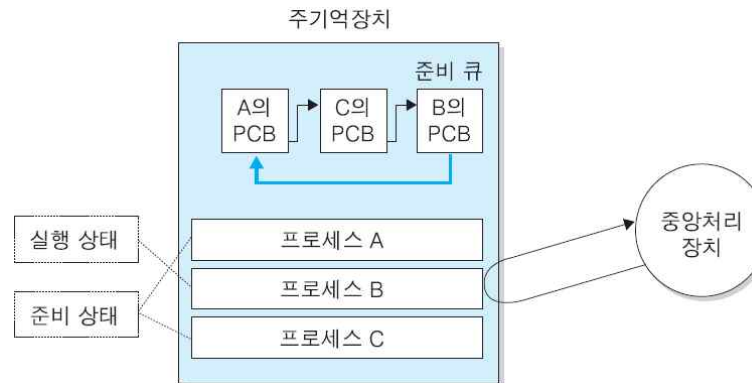
- 준비 상태의 프로세스가 실행 상태가 되는 경우(디스패치(dispatch):프로세스 B)
  - ✓ 프로세스 A의 실행이 종료된 경우



[그림 5] 프로세스 A의 실행이 종료된 경우

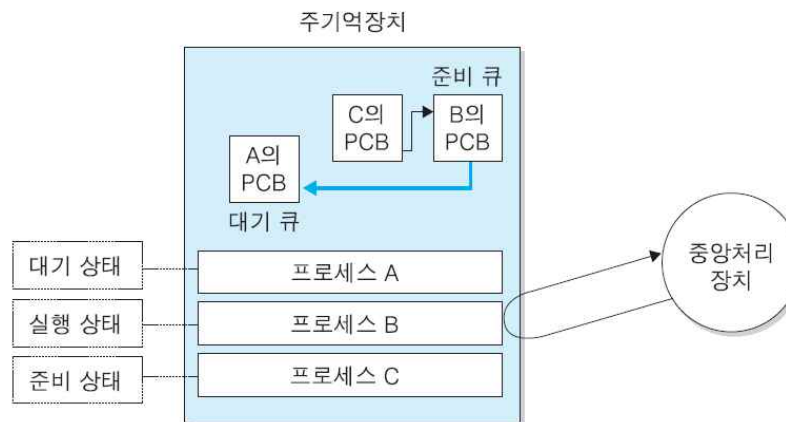
# 프로세스 상태의 전이 (4/5)

- 프로세스 A가 중앙처리장치를 오래 차지하고 있는 경우(timer runout : 프로세스 A)



[그림 6] 프로세스 A가 중앙처리장치를 너무 오래 차지하고 있는 경우

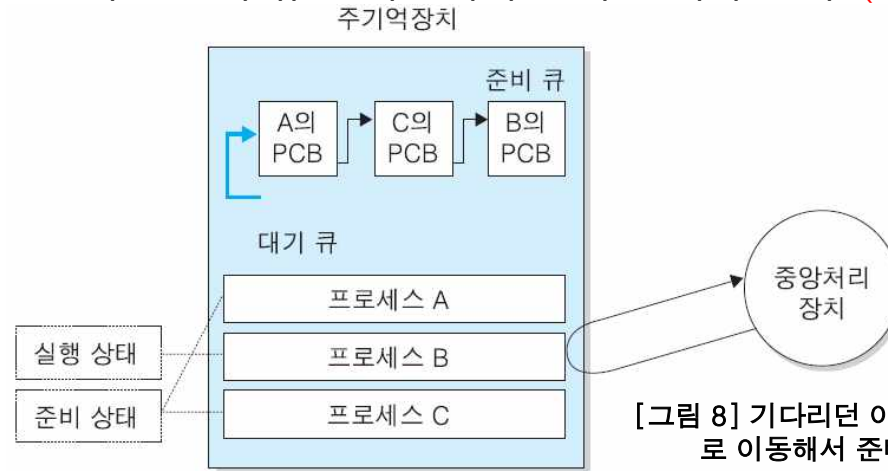
- 프로세스 A에 디스크 입출력 명령이나 sleep 명령이 발생할 경우 (block(입출력발생) : 프로세스 A)



[그림7] 프로세스 A에 이벤트 대기 명령이 발생한 경우

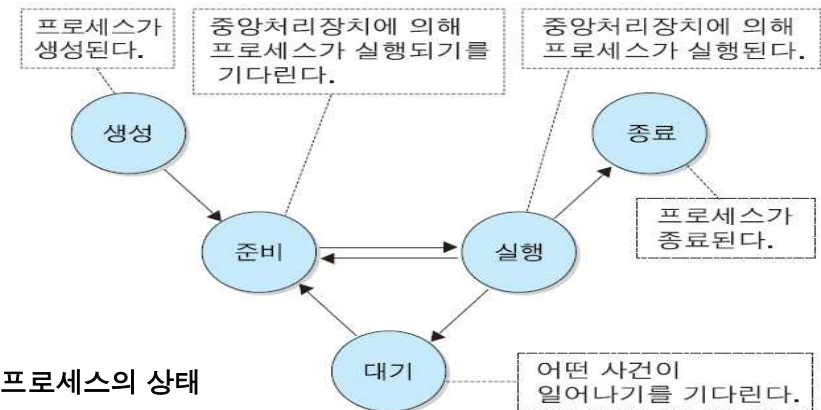
# 프로세스 상태의 전이 (5/5)

- 기다리던 이벤트가 종료되면 준비 큐로 이동해서 준비 상태가 된다. (wake up(입출력완료) : 프로세스 A)



[그림 8] 기다리던 이벤트가 종료되면 준비 큐로 이동해서 준비 상태가 된다

- 중앙처리장치가 실행할 수 있는 프로그램은 한 순간에 하나뿐이므로 하나의 프로세스만 실행 상태가 된다.



[그림 9] 프로세스의 상태

# 프로세스 상태의 전이 정리(1/2)

## ● 준비 → 실행

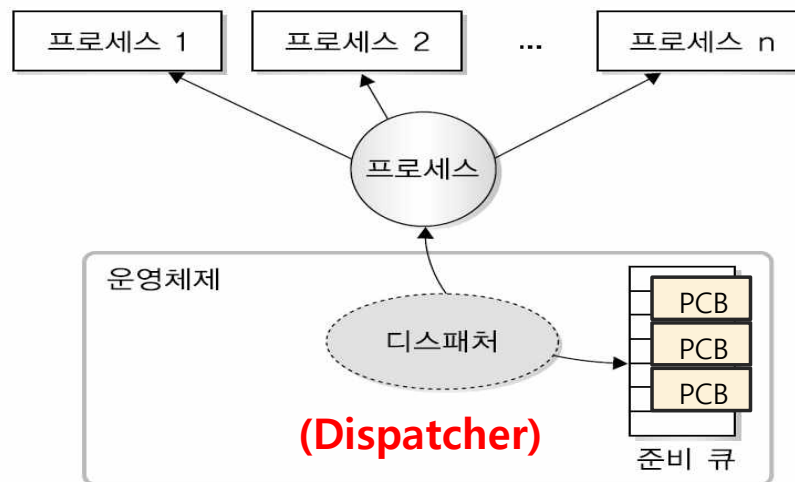
### ▶ 디스패치(Dispatch)

: 준비 리스트 맨 앞에 있던 프로세스가 프로세서를 선택(배당되어 실행)하는 것.

dispatch(프로세스명) : 준비 → 실행

### • 시간 할당

- 실행 프로세스가 프로세서를 일정 시간만 사용할 수 있도록 시간을 제한.
- 특정 프로세스가 프로세서를 계속 독점하는 것을 방지함.



프로세스와 디스패처

여러 프로세스에서 하나를 선택하는 작업을 **스케줄링**  
선택한 프로세스를 CPU에 올리는 것을 **dispatch**  
Dispatch 일을 하는 프로그램을 **dispatcher**  
Dispatcher와 스케줄링을 합쳐서 넓은 의미로 **스케줄링이라 함**



# 프로세스 상태의 전이 정리(2/2)

## ❖ 실행 → 준비

- 프로세스가 프로세서를 점유한 상태를 “프로세스가 실행 상태에 있다”라고 함.
- 운영체제는 프로세스의 독점을 방지하기 위해 인터럽트 클록(Interrupt Clock)을 둠.
  - 특정 프로세스가 일정 시간이 지나도(타임아웃) 프로세서를 반환하지 않으면 클록(Clock)이 인터럽트를 발생, 운영체제가 프로세서 제어권을 가짐.
- 운영체제가 제어권을 가지면 실행 중인 프로세스는 준비 상태로 변화, 준비 리스트의 첫 프로세스가 실행 상태로 변경.

timeout(프로세스명) : 실행 → 준비

## ❖ 대기(보류) → 준비

- 깨움(Wake up) : 프로세스의 마지막 상태 변화로 입출력 작업이 끝났을 때 발생.

wakeup(프로세스명) : 대기(보류) → 준비