

# Page & Segment

OS-Challenge 2주차

배서은

# Page의 이해

## Page

- Page : 프로세스를 일정한 크기로 나눈 블록 / 가상메모리의 블록(단위)
- Frame : 물리메모리를 일정한 크기로 나눈 블록
- Page number : 가상메모리의 인덱스

# 가상메모리의 배경

## 가상메모리

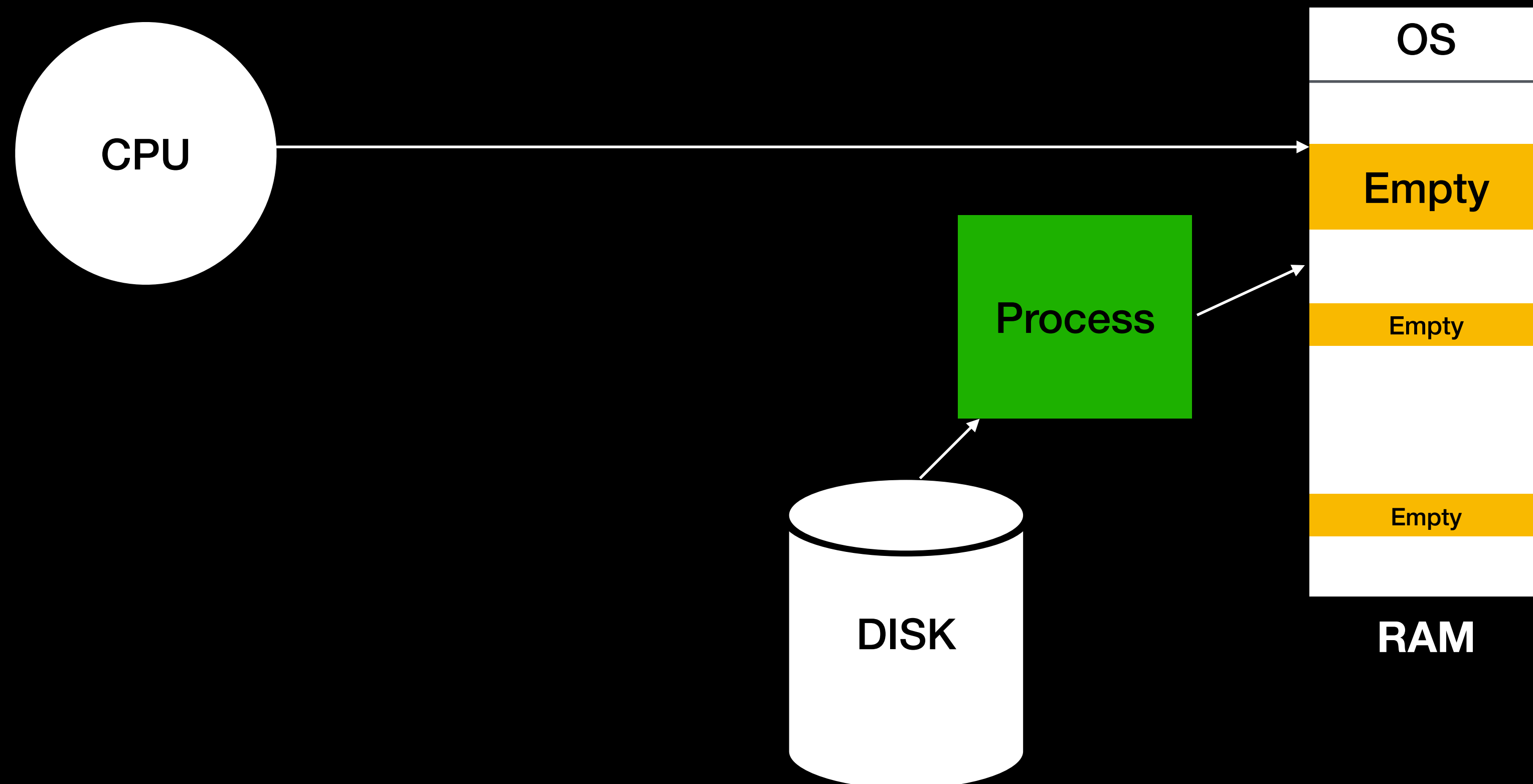
- 초창기 컴퓨터에서는 사용가능한 RAM의 용량이, 가장 큰 실행 애플리케이션의 주소공간보다 커야했다.
- 그 외에도 메모리의 외부단편화 문제 발생->버려지는 메모리가 많았다
- 가상메모리기법이 등장
- 애플리케이션의 주소 공간의 크기에 대해 RAM이 구애받지 않도록 한다.

# 가상메모리란?

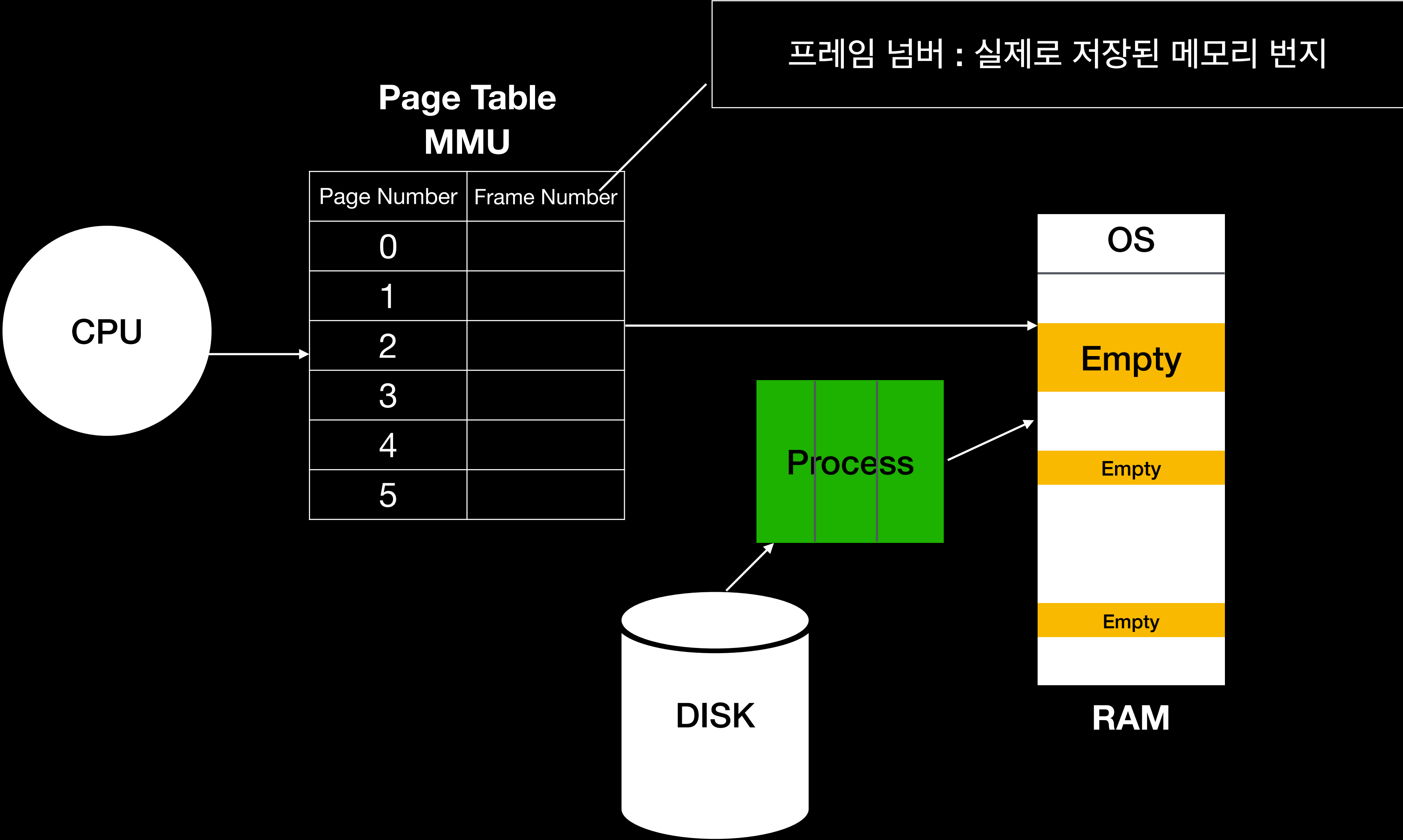
가상메모리

- 가상메모리기법
- 애플리케이션을 실행하는데 필요한 최소한의 메모리 크기에 집중!
  - 메모리접근은 순차적이고 지역화되기때문에 가능한 방식.
  - 애플리케이션의 일부만 메모리에 적재/나머지는 디스크에 위치
  - 즉, 디스크덕분에 가능

# Page의 이해



# Page의 이해



### 3. 가상 메모리란?

#### NOTE

가상 메모리는 메모리가 실제 메모리보다 많아 보이게 하는 기술로, 어떤 프로세스가 실행될 때 메모리에 해당 프로세스 전체가 올라가지 않더라도 실행이 가능하다는 점에 착안하여 고안되었음.

- 애플리케이션이 실행될 때, 실행에 필요한 일부분만 메모리에 올라가며 애플리케이션의 나머지는 디스크에 남게 됨. 즉, 디스크가 RAM의 보조 기억장치(backing store)처럼 작동하는 것임.
  - 결국 빠르고 작은 기억장치(RAM)을 크고 느린 기억장치(디스크)와 병합하여, 하나의 크고 빠른 기억장치(가상 메모리)처럼 동작하게 하는 것임.
- 가상 메모리를 구현하기 위해서는 컴퓨터가 특수 메모리 관리 하드웨어를 갖추고 있어야만 함. ⇒ 바로 **MMU(Memory Management Unit)**!

# MMU

## MMU

### - 하드웨어

- MMU는 가상주소를 물리주소로 변환하고, 메모리를 보호하는 기능을 수행함.
- MMU를 사용하게 되면, CPU가 각 메모리에 접근하기 이전에 메모리 주소 번역 작업이 수행됨.
- 그러나 메모리를 일일이 가상 주소에서 물리적 주소로 번역하게 되면 작업 부하가 너무 높아지므로, MMU는 RAM을 여러 부분(페이지, pages)로 나누어 각 페이지를 하나의 독립된 항목으로 처리함.
- 페이지 및 주소 번역 정보를 기억하는 작업이 가상 메모리를 구현하는 데 있어 결정적인 절차임.



# MMU 내부

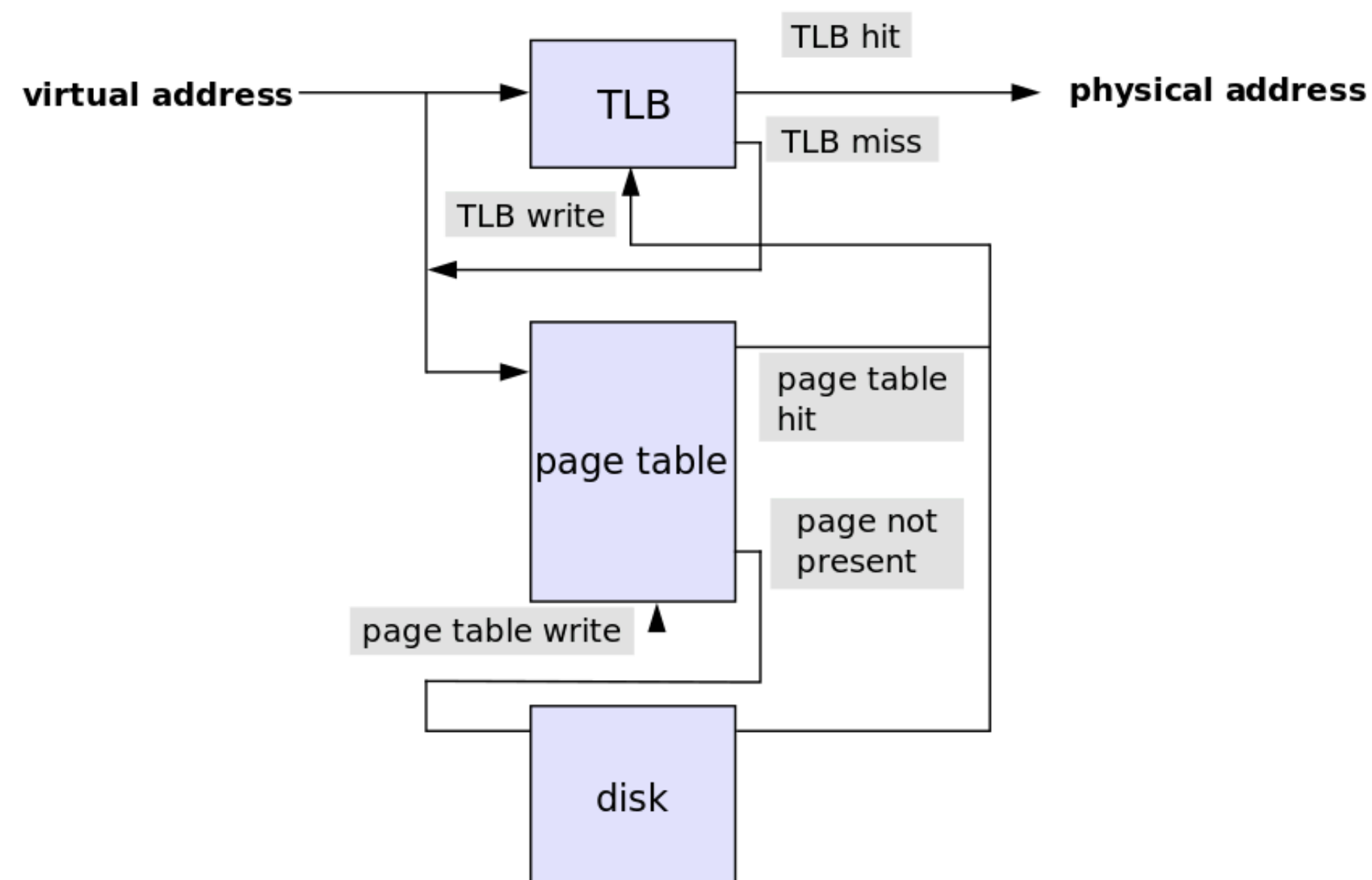
MMU는 TLB라는 캐시를 저장하고 있다. 가상주소가 물리 주소로 변환되어야 할 때, TLB에서 우선 검색된다.

해당 되는 주소가 있으면 (TLB hit) 물리주소가 리턴되고 메모리에 접근한다. 하지만, TLB에서 해당되는 주소가 없을 경우 (TLB miss) 페이지 테이블에서 �핑이 존재하는지 찾는다. 존재할 경우에 (page table hit) 이 값을 다시 TLB에 쓰이고 그 주소를 갖고 물리 주소로 변환 후, 메모리에 접근한다.

페이지 테이블에서도 찾지 못할 경우에는 disk에서 찾게 되고 그 값을 다시 page table에 쓰이고 TLB에 쓰이고 물리주소로 변환 후 메모리에 접근한다.

다시 정리하면, 가상 주소를 갖고 물리 주소에 접근할 때 TLB -> page table -> disk 순으로 접근한다고 생각하면 된다.

TLB는 processor 안에 있으며 page table에 경우 주로 메모리에 있으나 운영체제마다 다르다.



# Segment의 이해

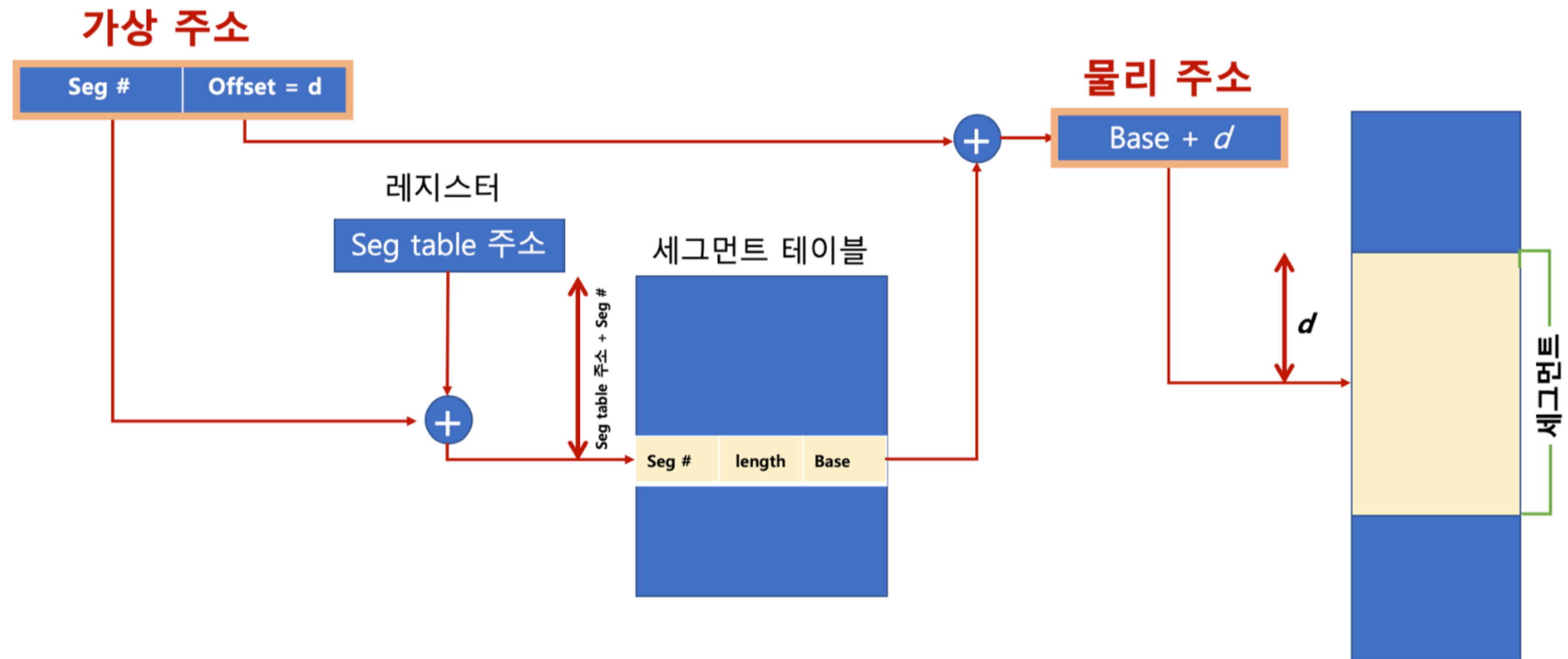
## Segment

- Process를 논리적으로 자른 단위 (일정한 크기가 아님!)
  - Ex. 코끼리 = 머리, 몸, 꼬리
- Page Table = Segement Table
  - 차이점 : **limit값이** 추가된다(범위에 한계를 주기 위함)
- Page Number = Segment Number

# Segment의 이해

## Segment

- $v = (s, d)$ :  $s$ 는 세그먼트 번호,  $d$ 는 블록 내 세그먼트의 변위



# 출처

1. KOCW 운영체제 양희재 교수님 강의영상(페이지 , 세그먼트 )
2. [MMU 자료]<https://about-myeong.tistory.com/35>
3. [가상메모리 자료]
  - a) <https://ahnanne.tistory.com/15#3.%20%EA%B0%80%EC%83%81%20%EB%A9%94%EB%AA%A8%EB%A6%AC%EB%9E%80?>
  - b) <https://velog.io/@codemcd/%EC%9A%B4%EC%98%81%EC%B2%B4%EC%A0%9C%OS-15.-%EA%B0%80%EC%83%81%EB%A9%94%EB%AA%A8%EB%A6%AC>