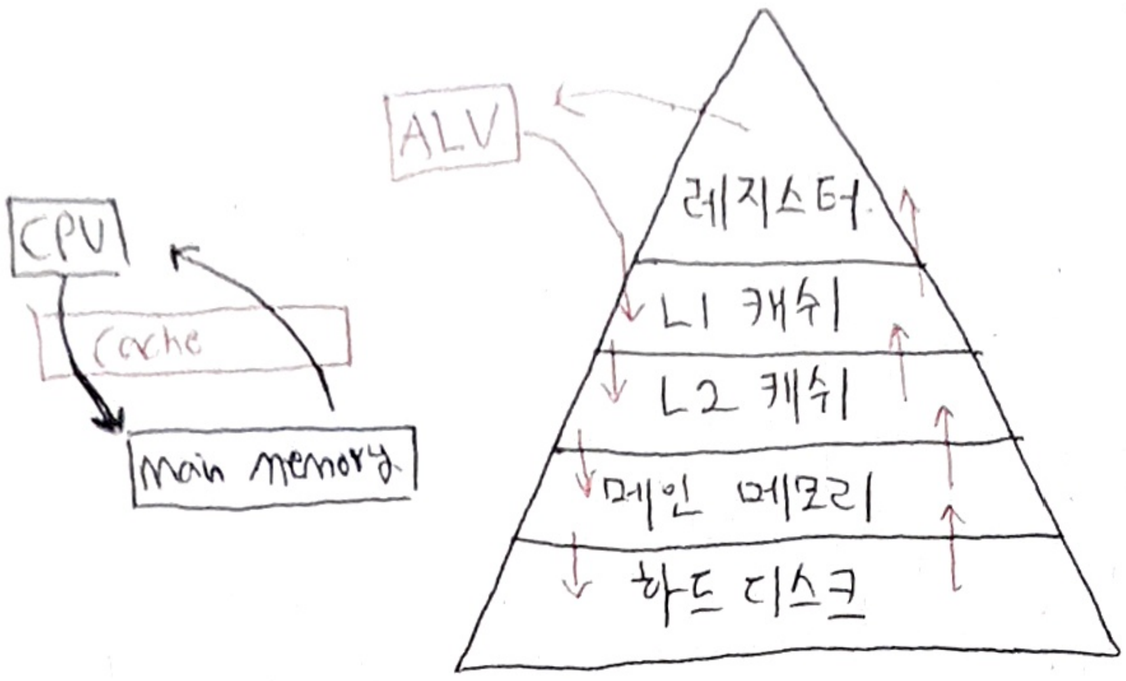
16장	메모리 계층 & 캐쉬 & 가상 메모리

-1/21 이충현



<메모리 계층>

계층적으로 움직인다 (프로그램의 지역적 특성)

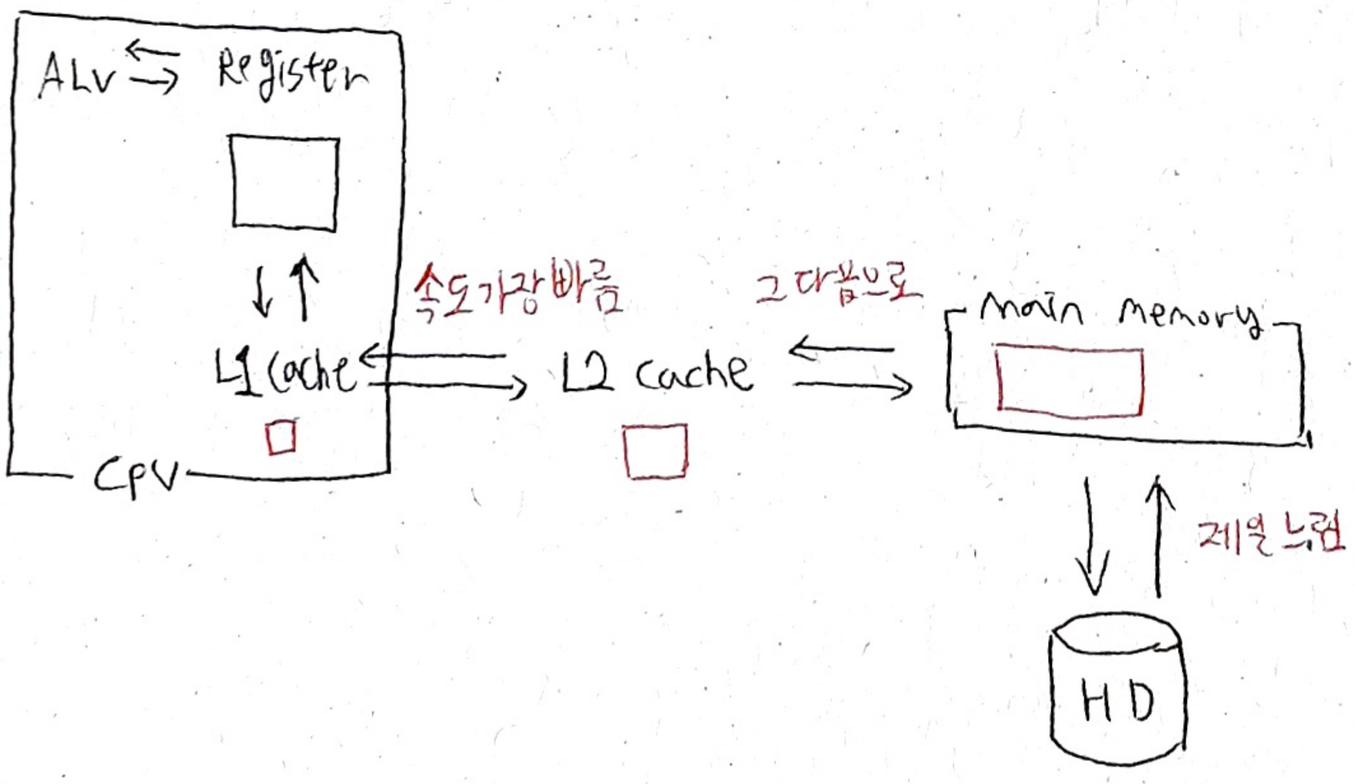
하드디스크 ① 실행의 관점 = Memory 관리
② 저장의 관점 = File system

<캐쉬 & 알고리즘>

프로그램의 ★지역성 특성 때문에 캐쉬는 도움이 된다!!

실제로 CPU가 찾고자 하는 데이터가 캐쉬에 있는 확률은 90% 이다!!

Locality ┬ Temporal : 반복접근 - 한번 접근이 이뤄진 주소의 메모리 영역은 자주 접근
└ ★Spatial : 주변접근 - (블록단위전송) ≒ 이미 접근영역 근처의 확률이 높은 성격



- 아래로 내려갈수록 블록 크기는 커짐, 접근 횟수를 줄이기 위해 (아래로 내려갈수록 속도가 느리다)

〈가상 메모리〉

- 하드 디스크까지 메인 메모리 영역을 확장!

가상 주소가 해결해야 될 두가지

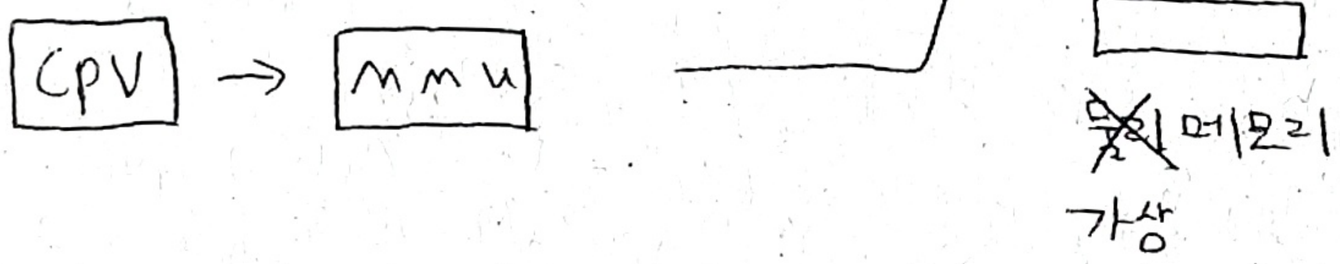
① 선 할당으로 인한 부담 ② 느린 속도의 개선
↓ ↓
HD는 너무 느리다.

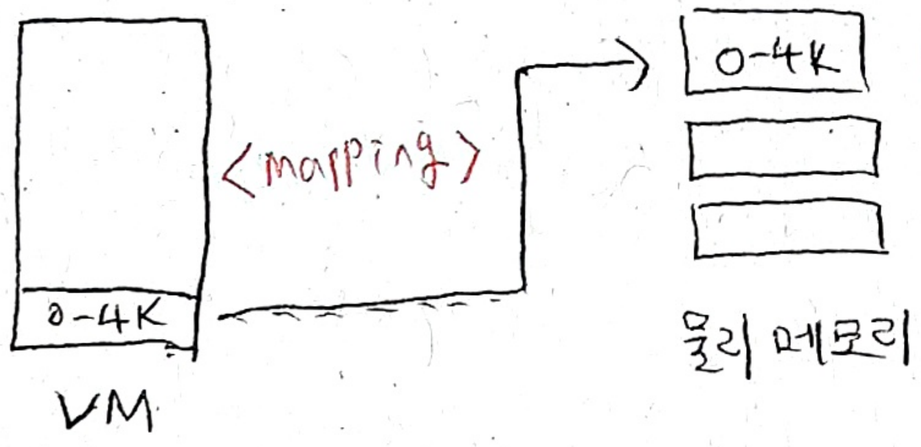
처리 할당하는 것은 조그마한 프로그램
실행할때도 크게 할당되면 메모리 낭비

① 의 해결책

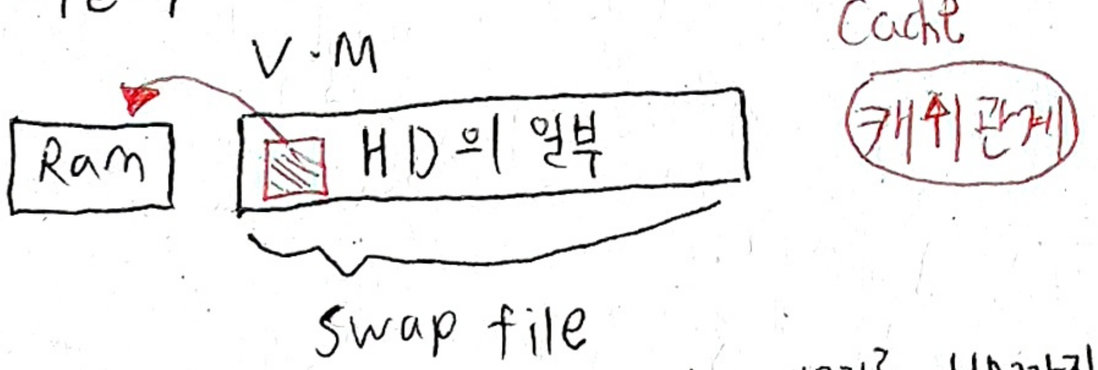
MMU가 SW입장
메모리블록마다 할당! (페이지 ~~프레임~~)

Memory Management Unit

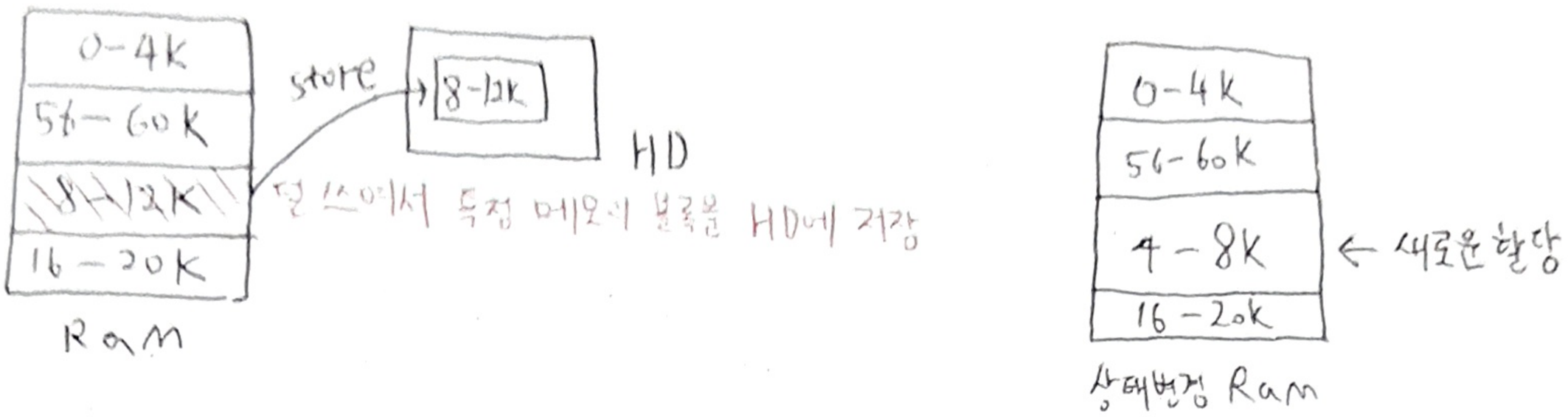




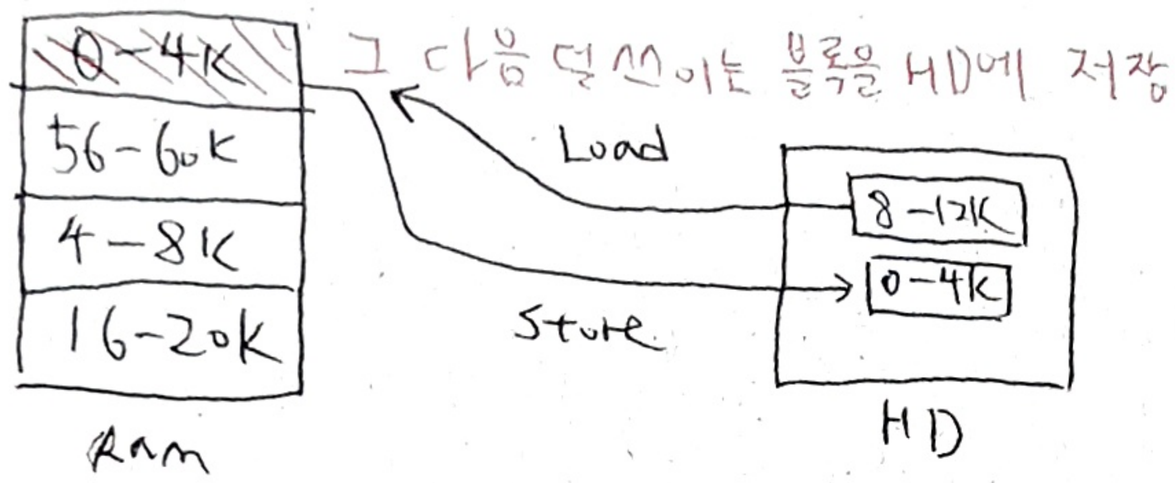
② 의 해결책



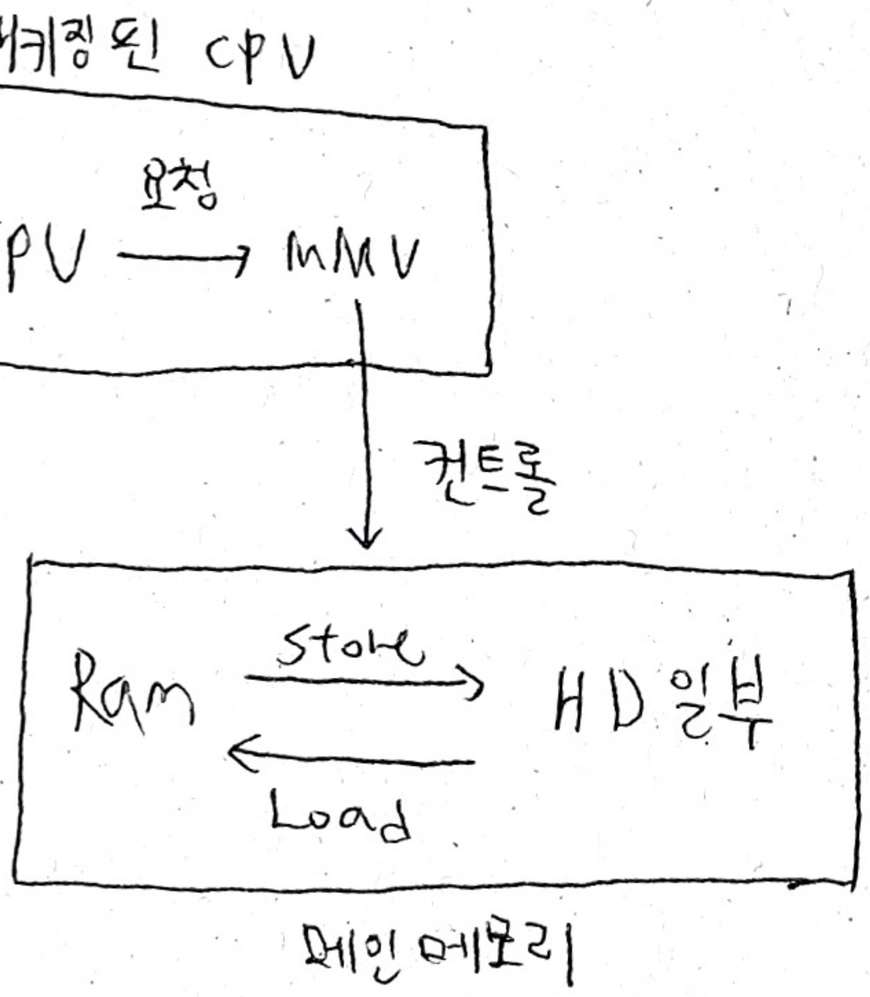
- Spatial / temporal Locality에 의해, 블록단위로 내가 필요한 거 RAM에다 저장.
- Ram에 해당하는 메인 메모리를 HD까지 확장!



다시 8-12k 영역 접근 발생시 어떻게 다시 꺼내올까?



〈정리〉



MMU - 가상메모리와 실제물리 메모리 사이에서
주소의 변환 담당

- VM 구현은 MMU라는 HW 블록의 도움덕분!