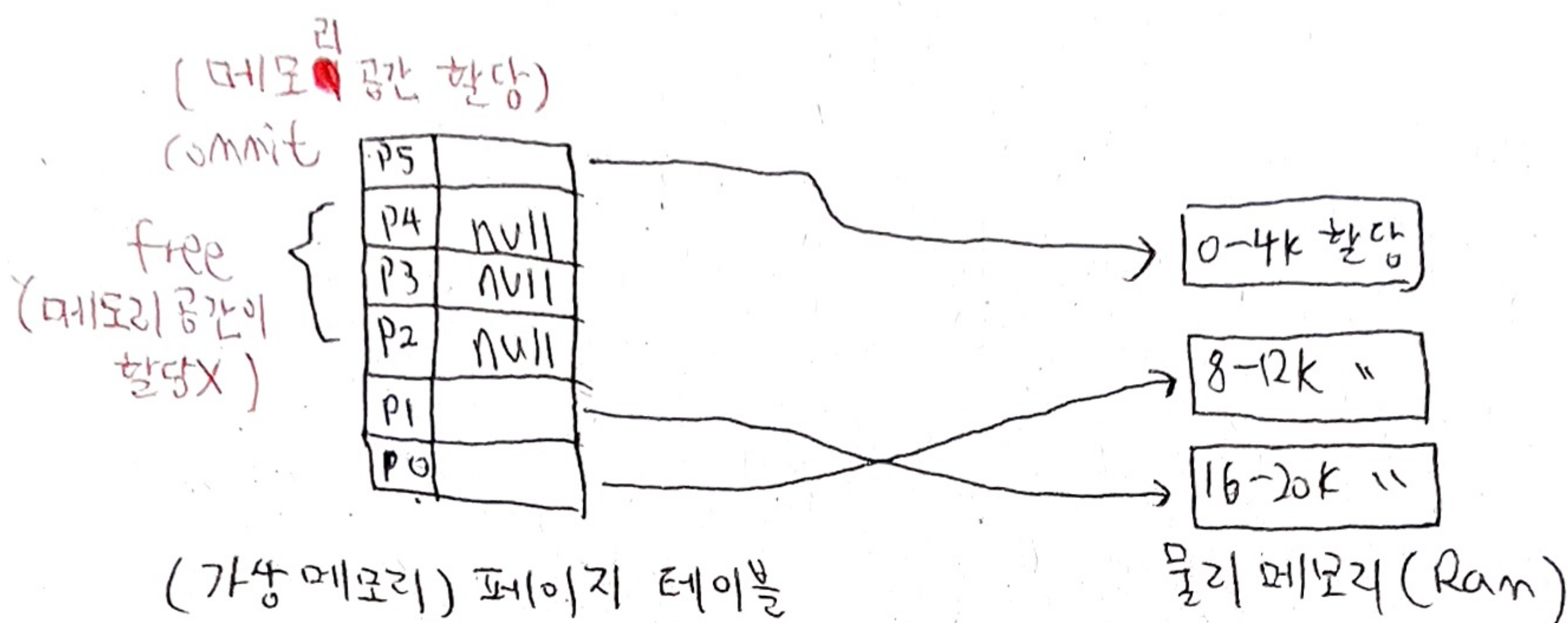


<가상 메모리 관리>



Commit → malloc 함수 호출

메모리 효율 증대 → Reserved 이용하므로써 다른 메모리 할당 함수에 의해 해당번지가 할당되지 못하도록 선언 (예약, 할당완료X)

메모리 할당기준

- ① 메모리 할당의 시작 주소
- ② 할당할 메모리의 크기

<Heap 컨트롤>

Heap 메모리에 할당시, 디폴트 Heap 영역에 할당된다 (프로세스 생성시)

기본 크기 = 1M 그러나 필요에 따라 늘어난다 (윈도우 시스템에서 알아서 늘려줌)

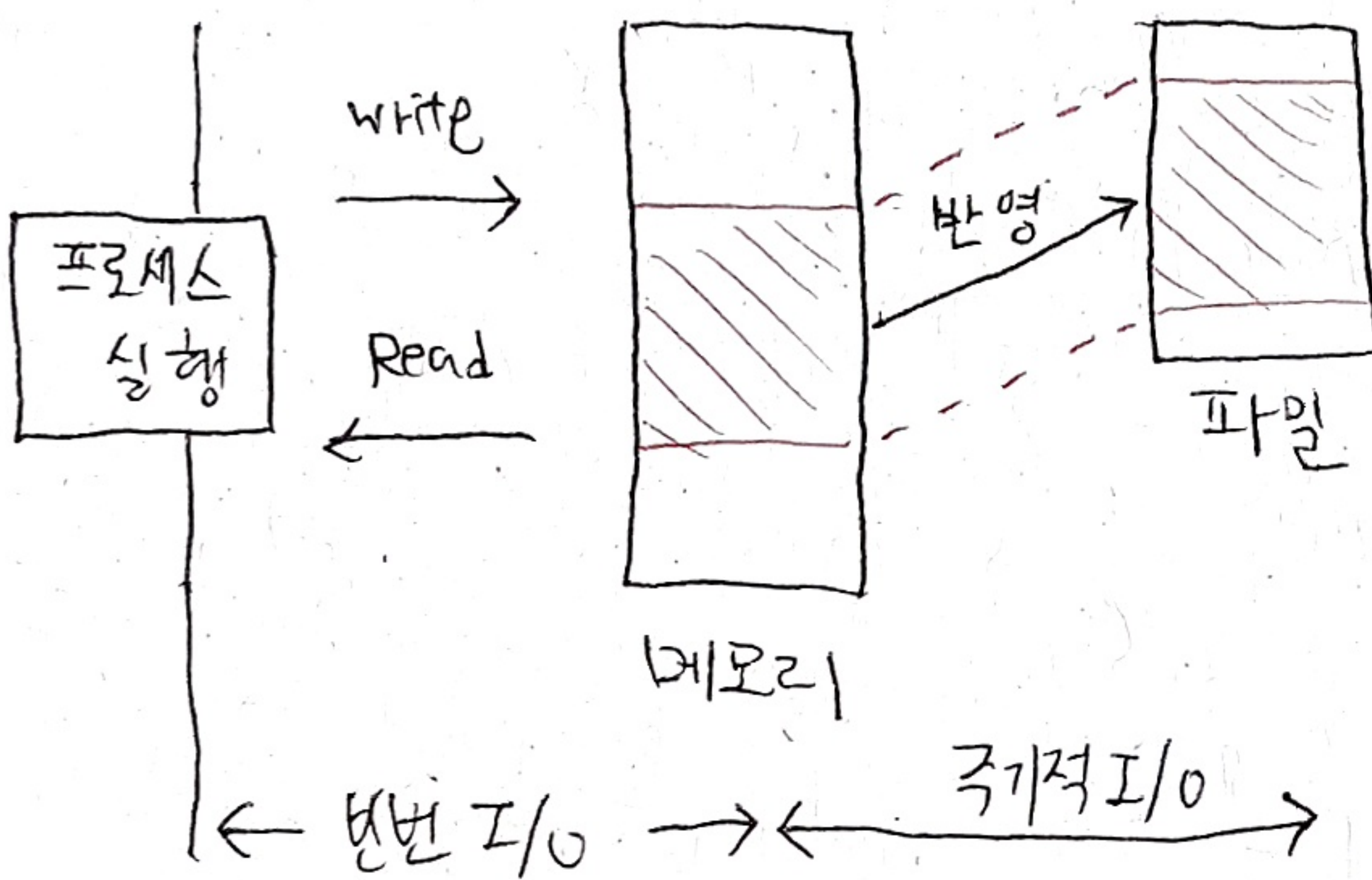
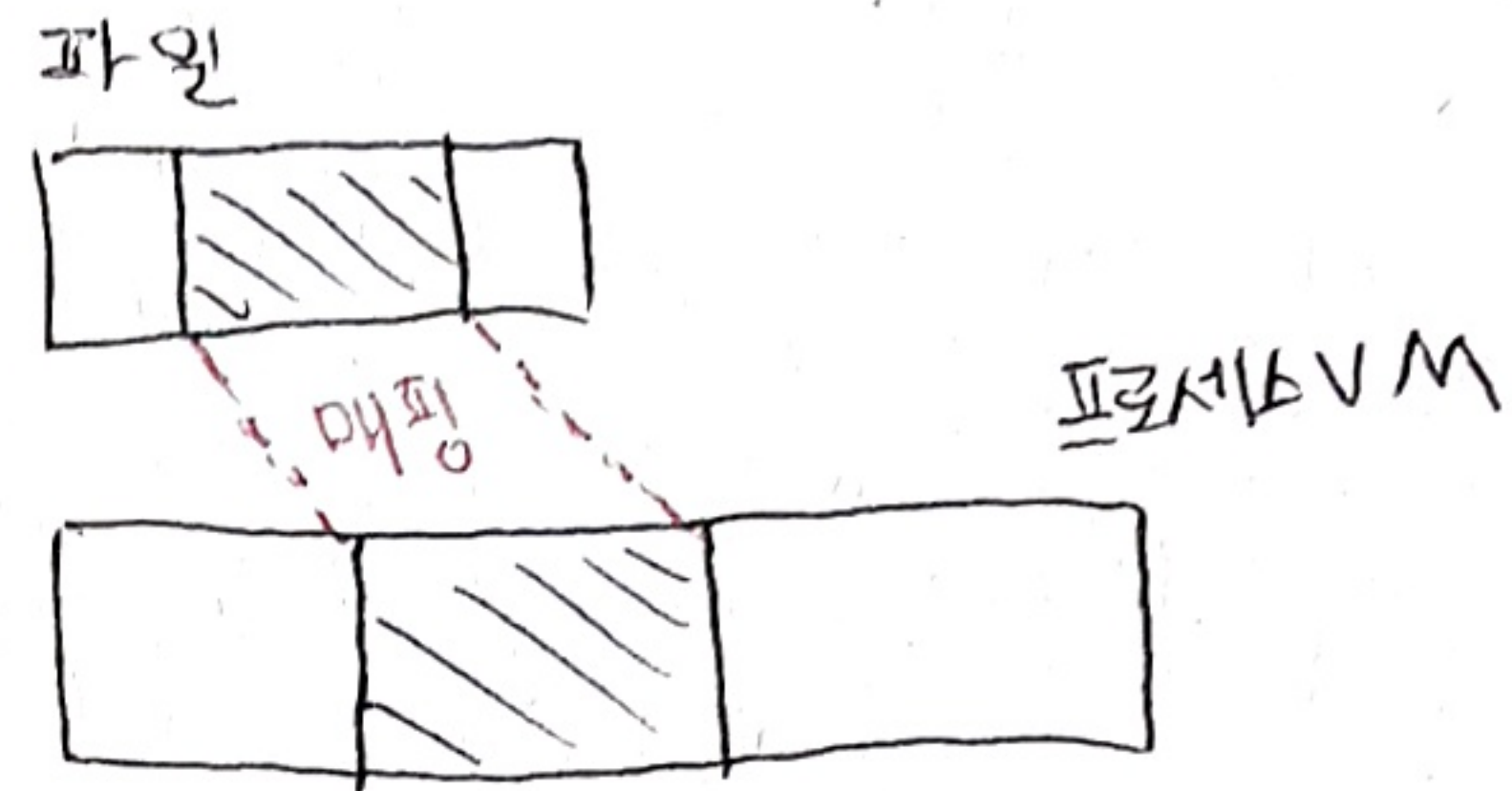
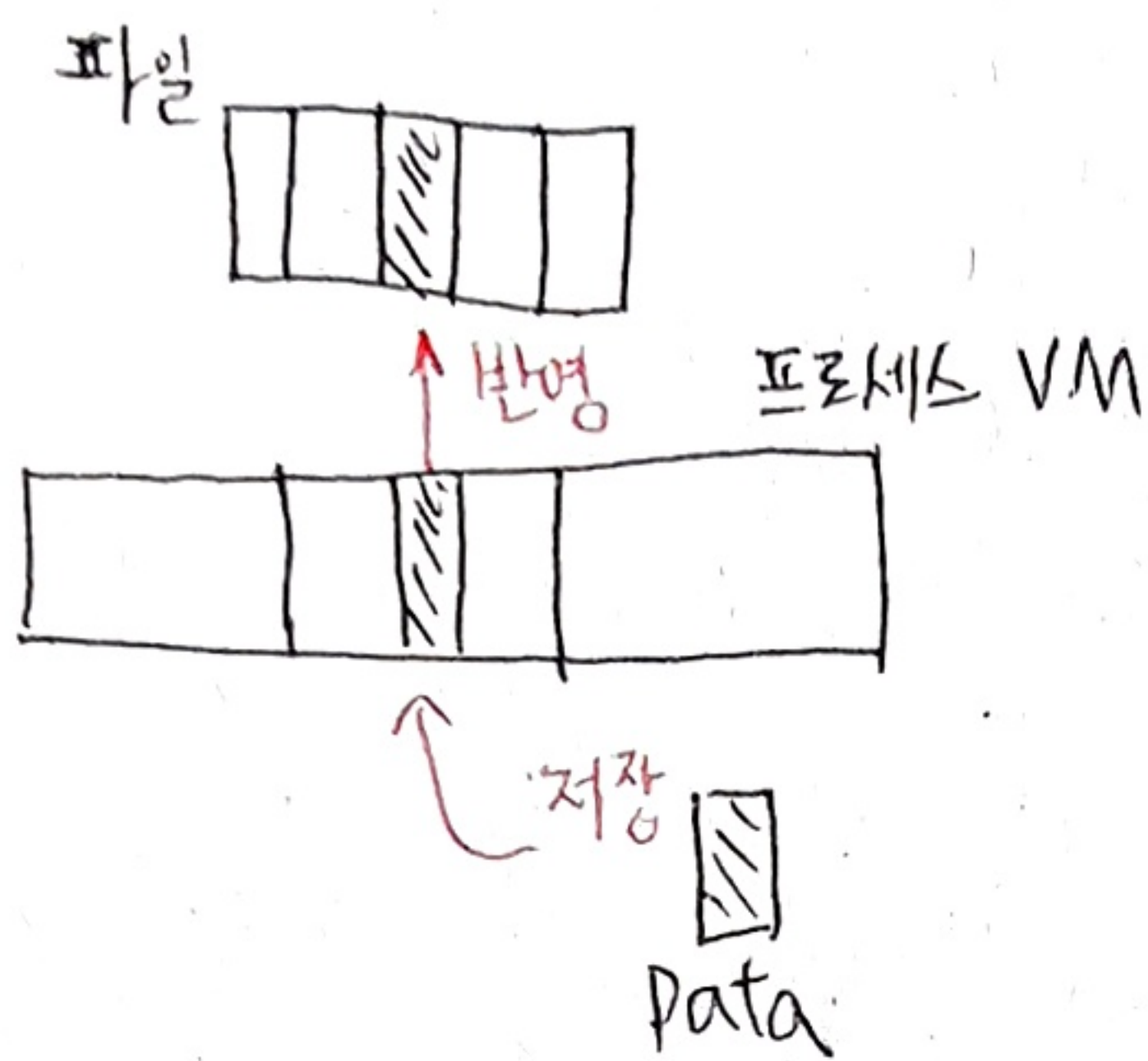
Window 시스템 함수 호출을 통해 생성되는 힙 = 동적 Heap

이점

- 메모리 간편화 해소
- 동기화 문제에서 자유 → 스레드별로 힙 생성 가능

<MMF (memory mapped file)>

파일의 일부 영역을 가상 메모리 일부에 연결시키는 메커니즘



직접 파일에 접근 안해도 됨
⇒ 성능 향상

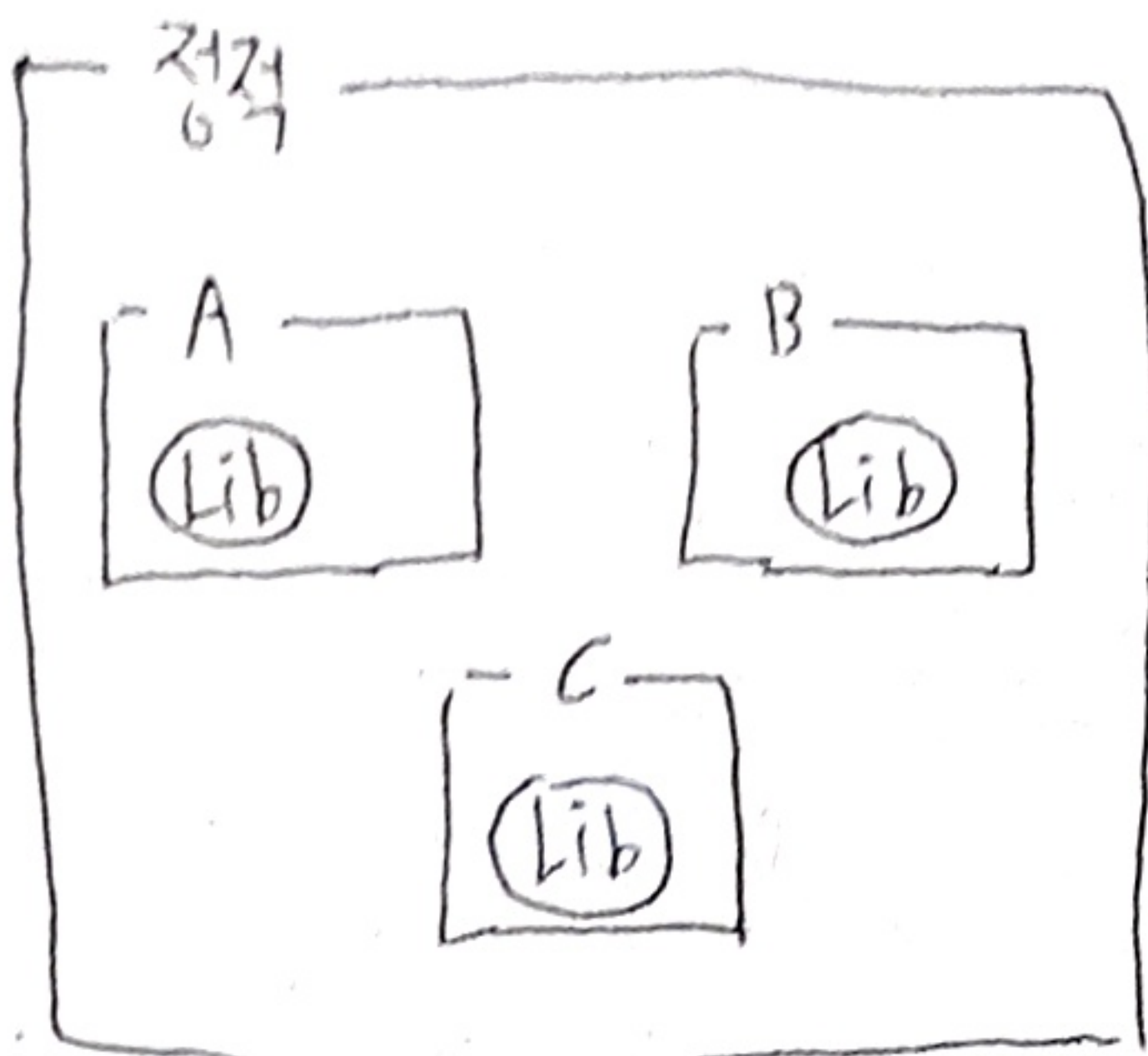
구현 과정

파일 생성 → 파일 연결 obj 생성 → VM에 파일 연결

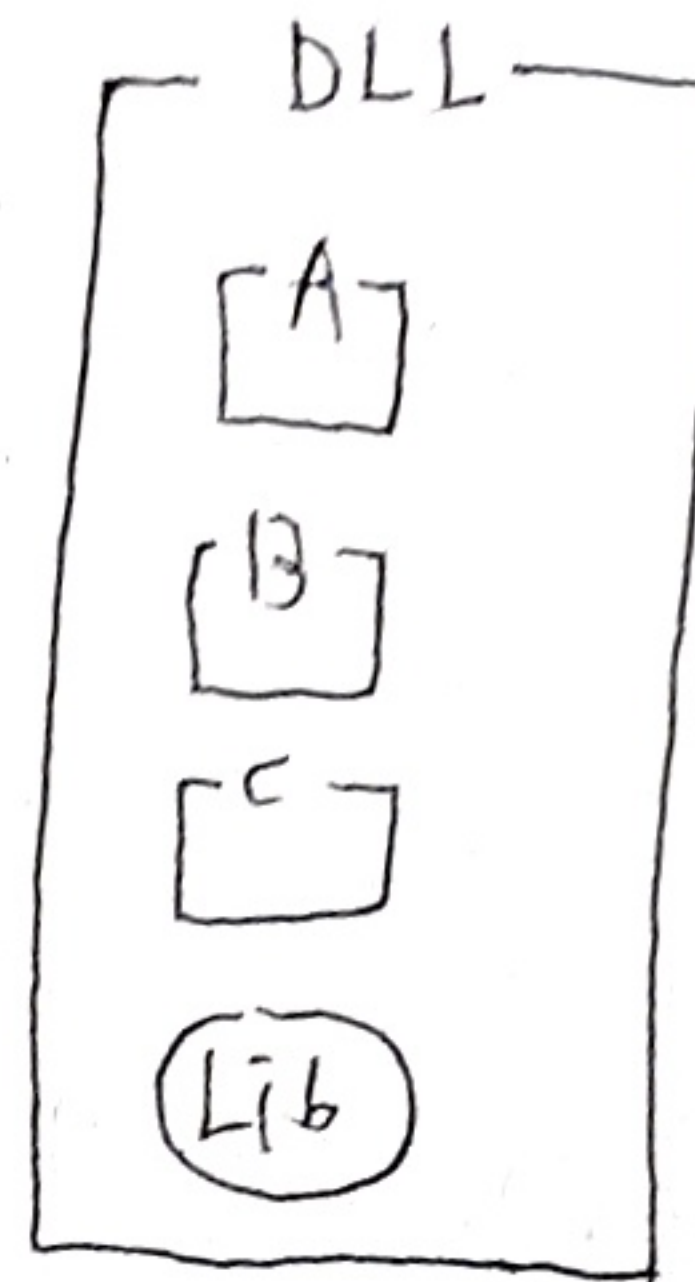
2장 Dynamic Linking Library (DLL)

- 1/28 이충현

< DLL과 정적 라이브러리의 차이점 >



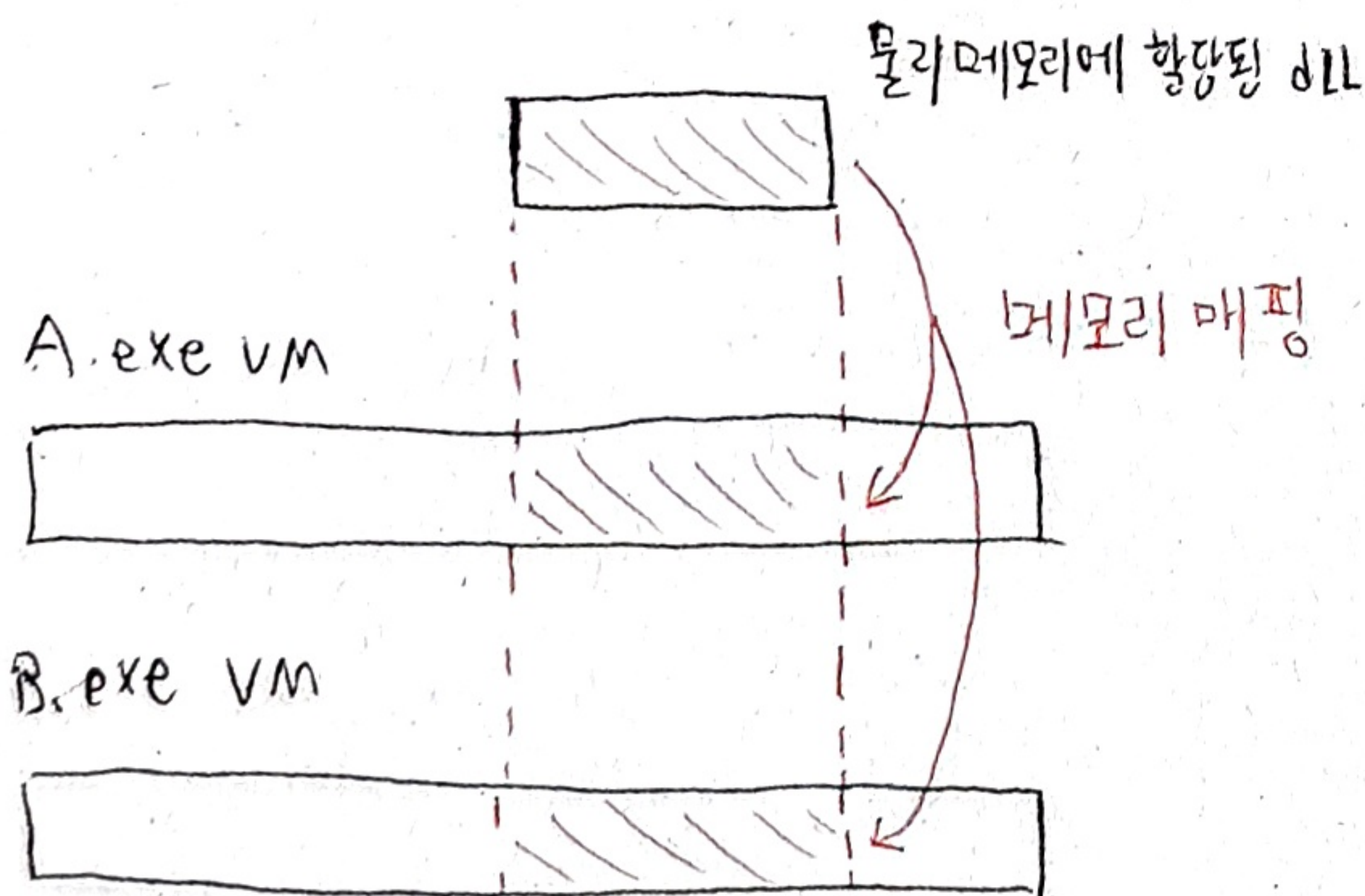
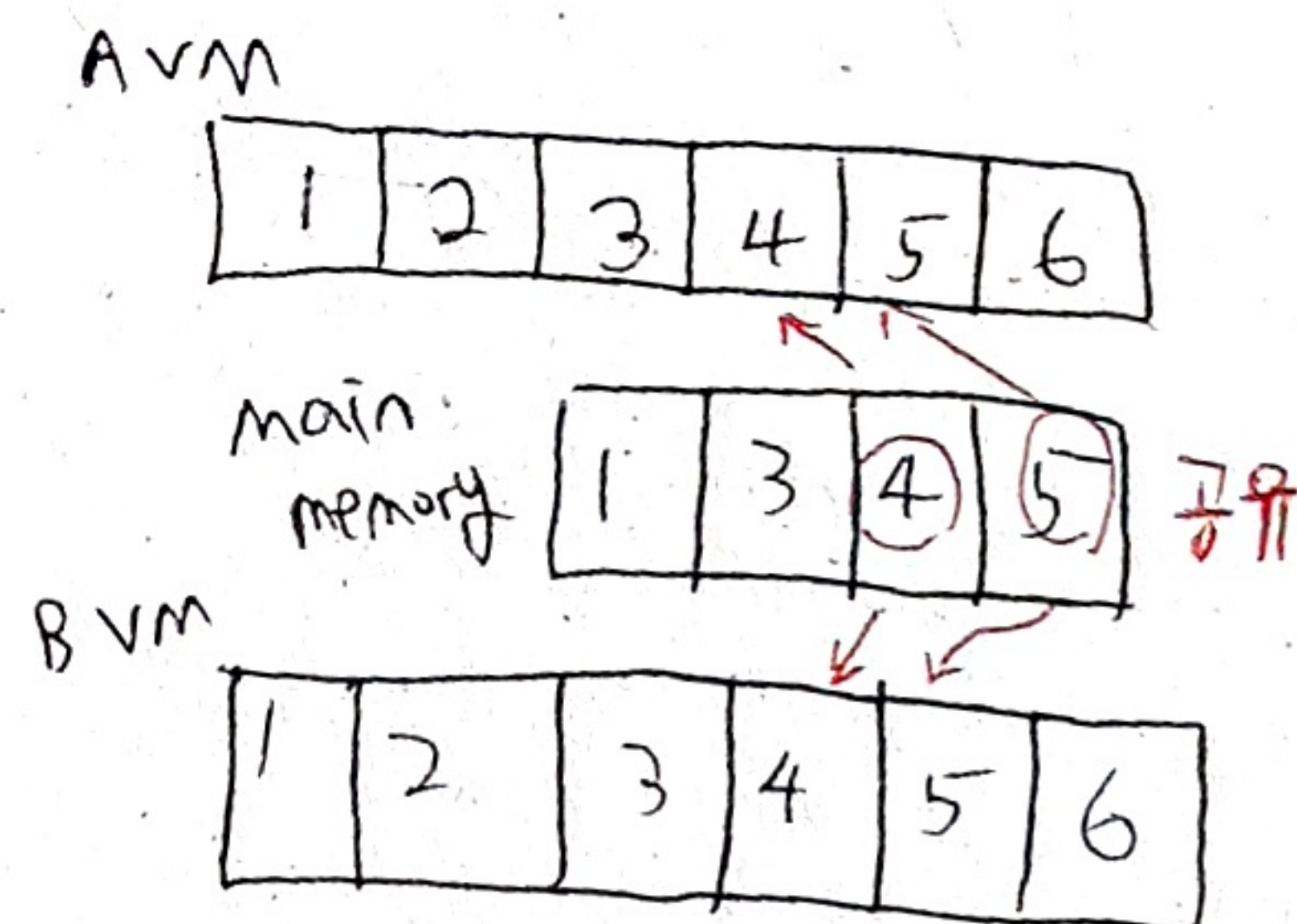
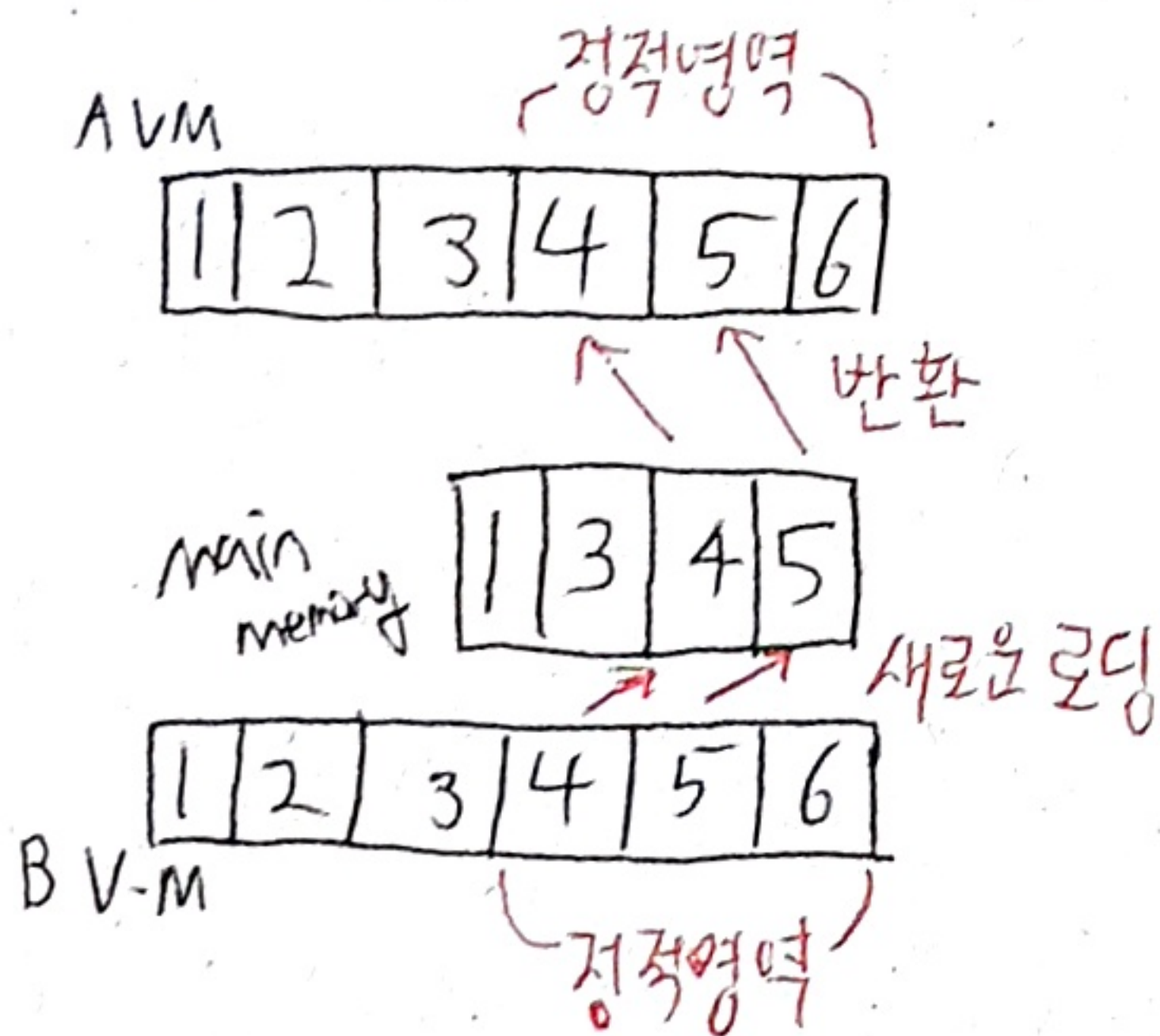
- 모든 실행파일이 동일한 라이브러리 포함
→ 메모리 공간을 더 차지함



- 라이브러리를 공유
→ 메모리 절약

< DLL 특성 >

둘 이상의 프로세스가 동일한 DLL 공유시, 메인 메모리에서 페이지 단위로 공유가 이루어진다.
↔ 정적은 메인 메모리에 있는 페이지를 교체해야 된다.



두 프로세스가 동일한 DLL을 동일한
가상 주소로 매핑했기 때문에
페이지 단위로 공유가능!