**3차 과제**

1. **섹터와 블록 매핑 알고리즘 구현**

- 가정해야 할 부분이 존재한다면 가정할 것 (여분 블록, 추가 DRAM 등 가정 가능)

*구현 함수:*

*init* ( );

*Print\_table* ( ) ; **//매핑 테이블 출력 각 Table 마다 Object 다르게 할 것.**

*FTL\_read* ( ); //Sector Mapping -> int array, size = sectorsize,

//sectorsize \* 4 byte + vector overhead

*FTL\_write* ( ); //Block Mapping -> int array size = sectorsize / 32, + vector overhead

// sectorsize / 32 \* 4 + vector overhead

Trusted Calculation allocation! // strict allocation

*Flash\_read* ( ); //플래시 메모리 하드웨어 구성 read, write, erase

*Flash\_write* ( );

*Flash\_erase* ( );

*Input 명령어:*

init megabytes //x megabytes 플래시 메모리 생성

read LSN

write LSN data

*결과값:*

init의 ouput: x megabytes flash memory ;

read의 output: 해당되는 PSN와 데이터 출력

write의 output: done,

각 Operation 실행 후

실제로 수행된 FLASH\_read 횟수

실제로 수행된 FLASH\_write 횟수

실제로 수행된 FLASH\_erase 횟수 및 PBN

예) 쓰기 완료,, 쓰기 5회, 지우기 연산 (PBN 13)

***유의 사항*** 구현하는 섹터/블록 매핑알고리즘의 DRAM 용량은 이론적 계산과 완전히 일치할 필요는 없습니다. 다만 여러분의 프로그램에서 필요로 하는 DRAM 용량을 보고서에서 계산/설명할 수 있어야 합니다.

**C. 보고서 작성**

**1. 요구 분석 (2장이상)**

**2. 설계 (3장이상)**

Flowchart로 보여주고, 각 과정 설명

**3. 구현 (3장이상)**

각 함수 설명 ;

input:

output:

함수 역할

중요 소스코드 라인 설명

**4. 테스트 결과 (3장이상)**

결과 분석

* 섹터 매핑과 블록 매핑 알고리즘 비교분석
  + DRAM
  + 성능 비교 (원인 분석)
  + 성능을 더욱더 향상시킬 수 있는 방향 고려