

1. Introduction

- 과제를 통해 OpenSSD Platform 사용에 익숙해지고, Flash Translation Layer (FTL) 를 직접 작성해 본다.

2. Overview

- 이번 과제에서는 OpenSSD Platform 상에서 Log Block FTL 를 작성하는 것이다.
- Log Block FTL 에 대한 자세한 내용은 아래의 논문을 참고하도록 한다.
J. Kim, J. M. Kim, S. Noh, S. L. Min, and Y. Cho, "A Space-efficient Flash Translation Layer for CompactFlash Systems," IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, pp.366-375, 2002.

3. Specification

- OpenSSD-1.1.1 을 기반으로 Log Block FTL 을 구현한다.
 - OpenSSD 와 NAND Module * 2 개로, 64 GB SSD 를 개발한다.
 - ./ftl_logblock/ftl.c 와 ./ftl_logblock/ftl.h 파일만 수정하여, Log Block FTL 을 구현한다.
 - ftl_read 와 ftl_write 를 통해 접근 가능한 LBA 는 0 ~ (NUM_LSECTORS-1) 이다.
 - ✓ NUM_LSECTORS 는 ./include/jasmine.h 에 정의되어 있다.
 - ✓ 64 GB SSD 의 경우, NUM_LSECTORS 는 124822461 이다.
 - Block #0 은 Firmware 및 Initial Bad Block List 등을 저장하는 용도이다.
 - Block #(VBLKS_PER_BANK) 부터 SPARE_VBLKS_PER_BANK 만큼의 Block 은 Reserved Block 으로 관리한다.
 - ✓ SPARE_VBLKS_PER_BANK 는 ./include/jasmine.h 에 정의되어 있다.
 - ✓ 64 GB SSD 의 경우, SPARE_VBLKS_PER_BANK 는 28 이다.
 - 그 외에 Block 은 Data Block, Log Block, Meta Block 등으로 사용한다.
- ftl_open 함수는 다음과 같이 구현한다.
 - ftl_open 에서는 format 을 수행한다.
 - ✓ FTL Metadata 를 초기화 한다.

- ✓ 각 Data Block 을 모두 매핑해준다.
- ✓ Initial Bad Block 은 Reserved Block 과 Remap 한다.
- **Bonus** – Run-time 시 Program 또는 Erase fail 이 발생하는 경우, 해당 Block 을 Reserved Block 과 Remap 한다. Program fail 에서 Remap 하는 경우에는, Program fail 이 발생한 Page offset 이전에 write 했던 data 는 Reserved Block 으로 Copy 해야 한다.
- ftl_read 함수는 다음과 같이 구현한다.
 - 한번도 write 하지 않은 LBA 를 ftl_read 로 read 하는 경우에는, 해당 LBA 는 반드시 0 으로 읽혀야 한다.
 - LBA 가 write 된 적이 있다면, 반드시 가장 최근에 write 했던 data 가 읽혀야 한다.
- ftl_write 함수는 다음과 같이 구현한다.
 - 하나의 Data Block 은 0 또는 1 개의 Log Block 만을 갖는다.
 - Data Block 은 Block 단위로, Log Block 은 Page 단위로 Mapping 한다.
 - Log Block 이 부족하거나, Log Block 의 page 를 모두 write 한 경우 Merge 가 발생한다.
 - ✓ Merge 는 Switch, Partial(Copy), Full Merge 를 모두 구현한다.
- **Bonus** – Power Off Recovery 를 구현한다.
 - 정상적으로 Power Off 를 할 경우에는 ftl_flush 함수가 호출되는데, 이 때 FTL Metadata 를 저장한다.
 - ftl_open 함수에서는 FTL Metadata 가 저장된 적이 있으면, 가장 최근의 Metadata 를 통해 복구하고, 저장된 적이 없는 경우에만 format 을 수행한다.
 - FTL Metadata 를 저장하기 위한 Block 은 #0 과 Reserved Block 이외의 Block 을 사용한다.

4. Restriction

- 과제는 윈도우 환경에서 수행하고, 과제 보고서에 컴파일하고 실행한 화면을 캡처하여 추가한다.
 - 실행 화면에는 Iometer 에서 Sequential Write 64KB , Sequential Read 64KB, Random Write 4KB, Random Read 4KB 를 각각 1 분씩 돌린 결과를 포함한다.
 - ✓ (Iometer의 Result 는 Update Frequency 는 1 초로, Start of Test 로 설정한다)
- Log Block 의 개수를 조절할 수 있도록 한다.
- Read 한 data 는 반드시 가장 최근에 Write 한 data 와 같아야 한다.

- 전원을 연결하고 5 초 이내에 ftl_open 함수의 수행을 종료해야 한다.

5. Hand in instructions

- 작성한 프로그램 코드 상단에 이름과 학번을 주석으로 표기한다.
- 과제 및 보고서는 제출시 “학번1_학번2.zip” 파일로 압축하여 제출한다. (예: 2011711283_2010400299.zip)
- 프로그램의 설계, 구현에 관한 내용을 담은 보고서를 제출한다. 보고서에는 프로그램의 구조를 그림으로 삽입하도록 한다. 보고서는 워드, 한글 등의 형식도 상관 없지만 가능하면 PDF 포맷으로 제출한다.
- 과제는 yjwoo@csl.skku.edu로 보내고, 메일의 제목은 아래와 같은 형식을 따른다.
[ICE3028] project1, 학번1, 이름1, 학번2, 이름2

6. Logistics

- 과제 제출 결과는 <http://csl.skku.edu/ICE3028S12> 에서 확인할 수 있다.
- 과제 제출 시간은 메일 도착시간을 기준으로 하며, 기한 이후 24시간 내에 제출할 경우 30%, 48시간 내에 제출할 경우 60% 감점한다. 그 이후에는 0점 처리한다.
- 과제에 대한 의논은 함께 할 수 있으나, 프로그램 소스코드 작성은 스스로 해야 한다.
- 다른 사람의 과제를 copy 한 경우, 두 팀 모두 0점 처리한다. 인터넷 등에서 찾은 소스코드를 그대로 copy 한 경우에도 0점 처리한다. 두 번 이상 이와 같은 이유로 0점 처리된 경우 F 학점을 받을 수 있다.