

과제 3: Flash Device Driver 활용

1. 개요

“Flash Memory” 강의에서 배운 Flash device driver에 대한 이해를 높이고 이를 활용하는 프로그램을 구현하며, 다음과 같은 제약 사항을 따라야 한다.

- 파일 I/O 연산은 system call 또는 C 라이브러리만을 사용한다.
- 아래의 (1), (2), (3), (4)의 기능을 ftlmg.c에 구현한다.
- 아래 (2), (3), (4)의 기능은 fdevicedriver.c의 인터페이스를 이용하여 구현한다.
- fdevicedriver.c는 주어진 그대로 사용하며 수정해서는 안된다.

(1) flash memory emulator

Flash memory 저장 장치를 모방하는 flash memory 파일을 생성한다. 여기에는 n 개의 블록과 각 블록에 m 개의 페이지가 존재하며, 각 페이지는 하나의 512B 섹터영역과 16B 스페어영역(spare area)으로 구성된다고 가정한다. 또한 블록은 4 개의 페이지로 구성되어 있다고 가정한다 (즉, m=4). 아래의 명령어를 실행시키면 블록의 수 <#blocks>로 구성되는 flash memory 파일 <flashfile>을 생성한다.

n개 블록, 각블록에 4개 페이지(각 페이지:512B 섹터, 16B 스페어)

a.out c <flashfile> <#blocks>

<예시>

a.out c flashmemory 100

옵션으로 c를 사용하며, 100 개의 블록과 각 블록은 4 개의 페이지로 구성되는 flashmemory 파일을 생성한다 (파일 크기: $100 \times 4 \times (512 + 16) = 211,200B$). 이때 모든 블록의 각 바이트는 0xFF로 초기화한다 (파일 전체를 0xFF로 채운다). Flash memory에서 'erase' 연산도 결국 초기화 작업이기 때문에 파일에서 0xFF로 초기화하는 방법은 fdevicedriver.c의 dd_erase() 함수를 참고하여 프로그래밍한다.

<주의>

- 생성한 flash memory 파일은 아래 (2), (3), (4)에서 사용한다.

(2) 페이지 쓰기

Flash memory 저장장치에 페이지 단위로 데이터 쓰기를 수행한다. 아래 명령어를 실행시키면 <flashfile> 파일의 <ppn>의 물리적 페이지 번호를 가지는 페이지에, 섹터영역에는 <sectordata>를 스페어영역에는 <sparedata>를 저장한다. 만약 <flashfile>의 블록의 수가 100이고 페이지 수가 4이면 ppn=0, 1, 2, ..., 399가 된다. <sectordata>와 <sparedata>는 각각 512B와 16B가 되어야 하나 그렇지 않아도 무방하다. 화면에 실행 결과를 출력할 필요가 없다.

// flash memory 파일 생성: 위에서 선언한 flashfp를 사용하여 flash 파일을 생성한다. 그 이유는 fdevicedriver.c에서 flashfp 파일포인터를 extern으로 선언하여 사용하기 때문이다.

// 페이지 쓰기: pagebuf의 섹터와 스페어에 각각 입력된 데이터를 정확히 저장하고 난 후 해당 인터페이스를 호출한다

// 페이지 읽기: pagebuf를 인자로 사용하여 해당 인터페이스를 호출하여 페이지를 읽어 온 후 여기서 섹터 데이터와 스페어 데이터를 분리해 낸다

// memset(), memcpy() 등의 함수를 이용하면 편리하다. 물론, 다른 방법으로 해결해도 무방하다.

a.out w <flashfile> <ppn> <sectordata> <sparedata>

<예시>

a.out w flashmemory 15 "abcd12345@%\$" "5"

옵션으로 w를 사용하며, ppn=15 즉, 16 번째 페이지에 큰따옴표로 묶여있는 abcd12345@%\$를 저장한다. 주어진 flashmemory 파일은 이미 생성되어 있어야 하며, 이 파일에는 최소한 16 개 이상의 페이지가 존재해야 한다. 주어진 섹터데이터를 해당 페이지에 처음부터 차례로 저장한다. 만약 주어진 섹터데이터가 512B보다 적을 경우 나머지 공간은 0xFF로 채운다. abcd12345@%\$를 16 번째 페이지에 처음부터 채우고 나머지 500B는 0xFF로 채운다. 스페어의 경우도 이와 같은 방식으로 처리한다.

<주의>

- <sectordata>와 <sparedata>는 blank를 포함할 수 있기 때문에 큰따옴표로 데이터를 묶어서 입력한다. 그렇지 않는 경우 채점 시 정상적으로 동작하지 않는다.
- 섹터 데이터와 스페어 데이터는 키보드로 입력할 수 있는 문자로 표현한다.
- 같은 페이지에 여러 번 쓰기 연산을 수행시킬 수 있다 (즉 갱신).

(3) 페이지 읽기

Flash memory 저장장치에서 페이지 단위로 페이지 읽기를 수행한다. 아래 명령어를 실행시키면 <flashfile> 파일의 <ppn>의 물리적 페이지 번호를 가지는 페이지에 저장되어 있는 섹터 데이터와 스페어 데이터를 화면에 출력한다.

a.out r <flashfile> <ppn>

<예시>

a.out r flashmemory 15
abcd12345@%\$ 5

옵션으로 r를 사용하며, flashmemory 파일의 ppn=15 페이지에서 섹터영역의 데이터와 스페어영역의 데이터를 읽어서 화면에 출력한다. 위의 (2)의 예시와 같이 데이터를 저장하였다고 가정한 것이며, 섹터 데이터와 스페어 데이터는 각각 첫 번째 0xFF 전까지의 의미 있는 데이터만을 출력한다.

<주의>

- 읽어야 할 페이지에 의미있는 데이터가 존재하지 않는 경우 (섹터와 스페어에 0xFF만 저장되어 있는 경우) 화면에 출력할 필요가 없다.

(4) 블록 소거(erase)

Flash memory 저장장치에서 블록 단위로 블록 소거를 수행한다. 아래 명령어를 실행시키면 <flashfile> 파일의 <pbn>의 물리적 블록번호를 가지는 블록을 소거한다. 화면에 실

행 결과를 출력할 필요가 없다.

```
a.out e <flashfile> <pbn>
```

<예시>

```
a.out e flashmemory 3
```

옵션 e를 사용하며, flashmemory 파일의 pbn=3 즉, 4 번째 블록을 소거한다.

2. 개발 환경

- OS: Linux 우분투 버전 18.0.4
- 컴파일러: gcc 7.5

**** 반드시 이 환경을 준수해야 하며, 이를 따르지 않아서 발생하는 불이익은 본인이 책임져야 함**

3. 제출물

- 프로그래밍한 소스파일 ftlmgc.c를 하위폴더 없이(최상위 위치에) zip파일로 압축하여 myclass.ssu.ac.kr 과제 게시판에 제출한다 (모든 제출 파일들의 파일명은 반드시 소문자로 작성). **제공된 flash.h와 fdevicedriver.c는 제출할 필요가 없음.**
- 압축한 파일은 반드시 학번_3.zip (예시 20061084_3.zip)과 같이 작성하며, 여기서 3는 세 번째 과제임을 의미함
- ** 채점 프로그램상 오류가 날 수 있으니 꼭 위 사항을 준수하기 바라며, 이를 따르지 않아서 발행하는 불이익은 본인이 책임져야 함**