



# TI DSP, MCU 및 Xilinx Zynq FPGA 프로그램 전문가 과정

Innova Lee(이상훈) 강사  
gcccompil3r@gmail.com

은태영 학생  
zero\_bird@naver.com

# Filesystem

메모리 관리 기법과 파일 관리 기법의 차이는 Naming 을 볼 수 있다.

파일은 meta data 와 user data 로 구분된다.

여기서 meta data 내부에는 inode(superblock) 을 통해 파일의 속성 정보나 데이터 블록 인덱스 정보 등을 관리한다.

메모리와 다르게 Naming 관리를 해 주는 이유로는 사용자 입장에서 가독성 및 직관성이 부족하기 때문이다.

디스크는 plotter, arm, head 로 구성된다.

plotter 안에는 track 들이 존재하며, 이것들의 집합을 cylinder 라고 한다.

track 은 sector 로 구분된다. 1개의 sector 는 512byte 를 나타내며, 이는 4kb 를 관리하기 위해 8개의 sector 가 사용된다는 말이다.

arm 은 head 가 읽을 곳으로 물리적으로 이동시켜주는 역할을 한다.

head 는 plotter 를 읽고 쓰는 역할을 한다.

head나 track 의 숫자나, 한개의 track 에서 몇개의 sector를 관리하지를 알면 디스크의 전체 용량을 알 수 있다.

해당 정보는 Device Drive 가 갖고 있다.

디스크는 자기장 패턴을 사용하며, 이는 회전을 통해 읽을 수 있다.

디스크에서 데이터에 접근하는데 걸리는 시간을 탐색 시간(seek time : 물리적으로 탐색하는 시간), 회전 지연 시간(rotational latency : 원판의 각속도와 관련), 데이터 전송 시간(transmission time : 읽은 정보를 전송하는 시간) 이 존재한다.

속도는 탐색 시간 < 회전 지연 시간 < 데이터 전송 시간 순으로 느리다.

# Filesystem

디스크 블록은 논리적인 번호를 갖고 있다.

페이지 프레임의 크기가 4kb이기 때문에 디스크 블록의 크기도 4kb 이다.

(하드 디스크는 실제 4kb 로 구분되어 있는 것은 아니다.)

디스크 블록의 할당 방법에는 연속 할당과 불연속 할당이 존재한다.

연속 할당의 경우 속도 면에서 장점을 갖지만 공간 효율이 떨어지고, 불연속 할당의 경우 공간 효율이 좋지만 속도가 느리다.

(디스크 조각 모음은 즉, 기존 정보를 연속 할당으로 재배치 하는 것이다.)

불연속 할당의 경우 블록체인 기법, 인덱스 블록 기법, FAT 기법이 존재한다.

블록체인 기법은 파일 헤더에 시작 위치와 크기를 저장하고 각 블록에 연결리스트를 두어 연결한 방식이다.

이러한 방식은 lseek 와 같이 파일의 끝을 읽으려면 모든 데이터를 읽어야 하는 일이 발생하고, 중간에 데이터가 유실될 그 뒤의 모든 데이터를 잃게 된다.

인덱스 블록 기법은 위치 정보들을 기록한 인덱스 블록(파일마다 존재)을 따로 사용한다.

lseek 로 인한 단점은 보완되지만, 인덱스 블록이 유실되면 데이터 전체가 소실되는 문제가 있다.

# Filesystem

FAT 는 하나의 FAT 을 통하여 다른 파일들을 모두 관리한다.

FAT 내부 데이터에서 0 은 free 상태, FF 파일의 끝을 나타낸다.

FAT 구조의 문제점은 FAT 구조의 유실 시, 파일 시스템 내의 모든 파일이 소실되는 문제점이 있다.

이를 해소하고자 Ext3 이후 FAT 내용을 중복하여 관리한다. 그것을 통하여, 데이터 복원이 가능하게 되었다.

FAT 의 meta data 는 FAT 테이블, 디렉토리 엔트리, 슈퍼 블록으로 구성되어 있다.

디렉토리 엔트리 는 파일 시스템들마다 구조체를 선언해 두고, 파일 생성이 내용을 채워서 저장한다.

즉 특정 파일을 검색하고자 할 때 이용된다.

디렉토리에서 검색 방법으로는 ‘/’ 디렉토리에서 이름을 통해 해당 블록 번호를 찾고, 그 블록으로 이동하여 추적하는 방식을 사용한다.

여기서 ‘/’ 를 나타내는 것이 슈퍼블록이다.

# Filesystem

inode 의 구조를 살펴보자.

i\_blocks : 몇개의 블록 데이터를 갖고 있는지 나타낸다.

i\_mode : 파일의 속성 및 접근 제어 정보를 나타낸다.

i\_links\_count : inode 를 가리키고 있는 파일의 수를 나타낸다.

i\_uid / i\_gid : 유저 ID 와 그룹 ID 를 나타낸다.

i\_atime / i\_ctime / i\_mtime : 각각 접근 시간, 생성 시간, 수정 시간을 나타낸다.

하부에 i\_block[15] 이 존재한다.

즉 15개의 엔트리가 존재하는데, 이중 12개는 직접 블록 이고, 3개는 간접 블록이다.

직접 블록은 실제 위치를 직접 나타내는 것으로 48kb 의 범위를 갖고있다.

3개의 간접 블록은 단일 간접 블록, 이중 간접 블록, 삼중 간접 블록으로 이루어져 있다.

각각의 간접 블록은 내려갈수록 인덱스 블록 한 개가 추가된다.

즉 4kb, 4mb, 4gb, 4tb 로 구성되어 있다.

# Filesystem

Ext 시스템.

회로 차이에 따라 IDE 방식과 SCSI 방식으로 구분되며, 각각 hd 와 sd 로 확인할 수 있다.

디스크는 사용자가 원하는 갯수만큼 분할 할 수 있는데, 이를 파티션 분할은 최대 64개까지 지원한다.

...