TI DSP, MCU 및 Xilinx Zynq FPGA
프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – 하성용
accept0108@naver.com

35 일차
chapter5

- 파일 시스템

메모리 관리 기법과 파일 시스템의 차이점은?
‘이름’(예를들어 약속.txt)이라는 특성이다
파일 시스템은 이 기능을 사용하기위해 메타데이터(‘이름’과 파일정보,인덱싱 정보)와
사용자 데이터(사용자가 실제 기록하려 했던 기록)를 하드디스크에 저장한다

- 디스크 구조와 블록 관리 기법

디스크는 자기장으로 동작하는데
디스크가 찍히는순간 자기장패턴을 읽게됨(ex.레코드)

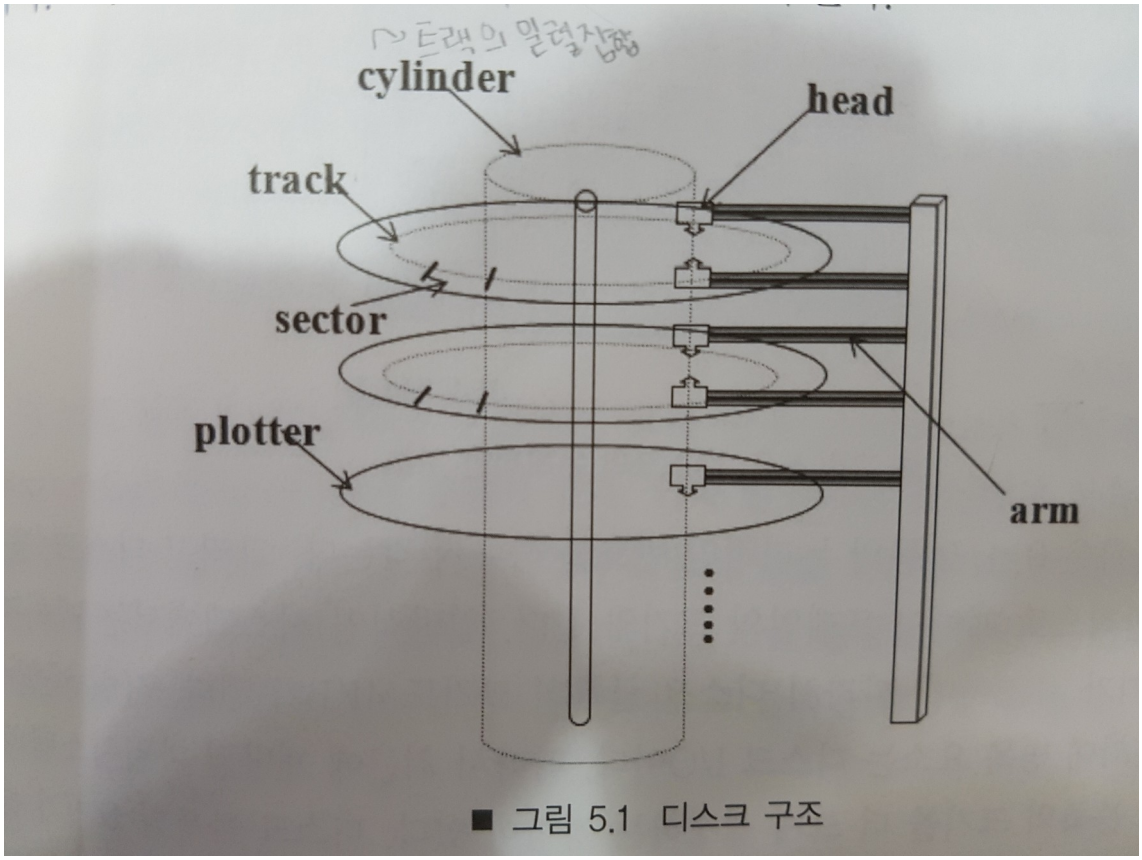
디스크는 원판(plotter), 팔(arm), 헤드(head)로 구성
원판(plotter)에는 원 모양의 트랙(track)들이 존재하며,
모든 원판에서 같은 위치를 갖는 트랙들의 집합을 실린더(cylinder)라고 한다.
트랙은 다시 몇 개의 섹터(sector)로 구분된다.
섹터는 디스크에서 데이터를 읽거나 기록할 때 기본 단위가 되며 일반적으로 섹터의 크기는 512byte 이다.
헤드는 각 원판의 읽기/쓰기가 가능한 면마다 하나씩 존재한다
따라서 헤드가 몇개, 트랙이(또는 실린더가) 몇개, 각 트랙마다 섹터가 몇개 등이 결정되면
디스크의 전체 용량 등 해당 디스크의 물리적 특성을 결정 할 수 있게 된다.

탐색시간(seek time) :
디스크에서 데이터를 접근하는 데 걸리는 시간

회전지연시간(rotational latency) :
요청한 섹터가 헤드 아래로 위치 될 때까지 디스크 원판을 회전시키는데 걸리는 시간

데이터전송시간(transmission time) :
헤드가 섹터의 내용을 읽거나 또는 기록하는데 걸리는 시간

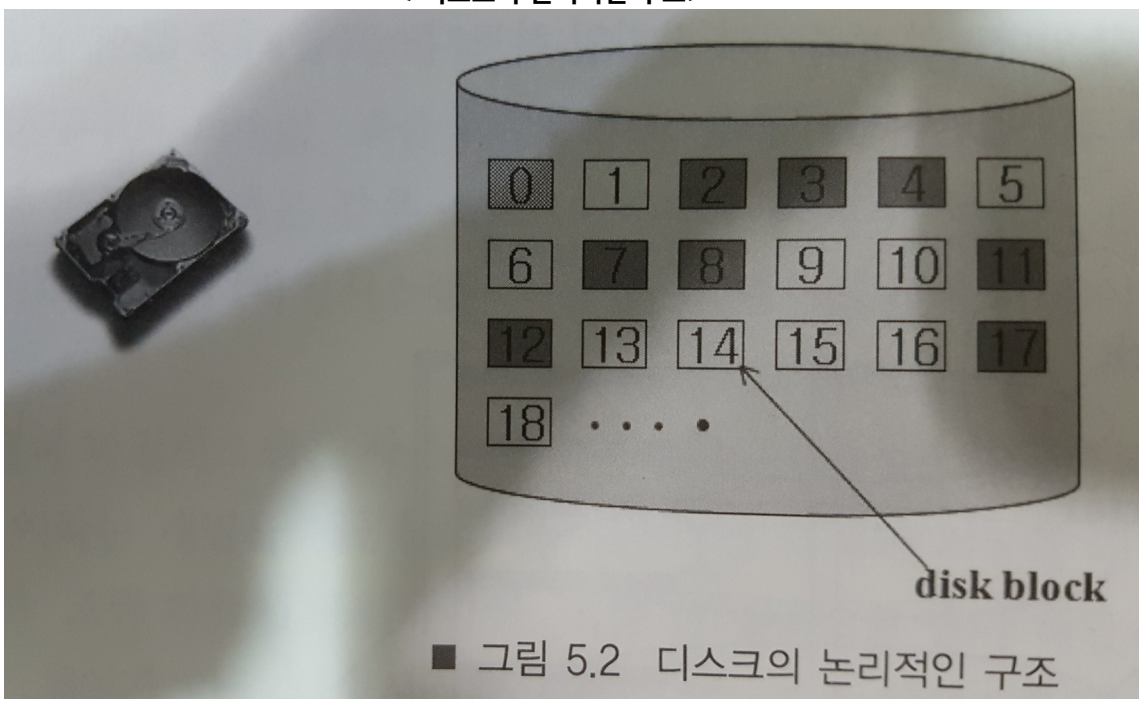
< 디스크의 일반적인 구조 >



■ 그림 5.1 디스크 구조

파일시스템은 디스크를 물리적인 구조로 보지않고 논리적인
디스크 블록(disk block)들의 집합으로 본다

< 디스크의 논리적인 구조>



■ 그림 5.2 디스크의 논리적인 구조

디스크 블록은 0, 1, 2 등의 논리적인 번호를 하나씩 가지고 디스크블록 하나의 크기는 4KB 이다
사실 파일시스템 성능의 최대 병목 요소는 디스크 I/O 인데 최근에는 디스크블록의 크기를 크게 설정해서 한번의 Disk IO 로 더 많은 데이터를 메모리로 읽어 들이게하기도 한다
그러나 디스크 블록의 크기와 '속도'측면에서의 성능은 비례하지만,
'공간효율성'면에서의 성능은 반비례한다.

- 디스크 블록의 할당과 회수방법
예를들어 파일 시스템이 14KB 크기의 파일생성요청을 받았다고 가정하면 각 디스크의 블록의 크기가 4KB 라고 가정하였을때 총 4 개의 디스크블록이 필요하며
free 상태의 블록이 1, 5, 6, 9, 10, 13, 14, 15,16 이 있다

디스크 블록을 할당하는 방법에는 연속(sequential)할당과 불연속(non-sequential)할당 방법이 있는데 공간을 넓게 쓰고 싶다면 불연속할당으로 1, 5, 6, 9 번에 디스크블록을 할당할수도있다

불연속할당을 할경우에는 파일에 속한 디스크 블록들이 어디에 위치하고 있는지에 대한 정보를 기록해 두어야한다.
연속할당으로 13,14,15,16 을 할경우에는

136
vi -t ext2_inode
vi -t ext3_inode// 안보이고
vi -t ext4_inode

exit2 와 4 나눠놓은이유
2 는 4 테라정도가 한계
3 부터는 모든걸 복구가능
4 는 리눅스가 수용해야할 데이터가 방대함
하루에 정보가펙타씩 쌓임
수용할수있는 정보의 한계를 무한대로하자해서 나온게 4

msdos_dir_entry // FAT 파일시스템

vi -t ext2_sb_info
vi -t ext4_sb_info

exit4 는 구글에서 쓰이는데 Jouraling 에서 쓰고있는데 정보가 날아가면 복구가능

p149
디렉토리 관련된 파일시스템
함수포인터로 선언해놓고 파일시스템에 따라서 적절하게 래핑을 해주는것
함수포인터를 쓰는 이유 파일시스템이 수천개가 되어서

EXT4 가 NTFS 에 무엇을 쓰고싶을때 쓸수있음
함수포인터가 다있기때문에
ext4_opne ntfs_open
ext_read ntfs_write
ext_close nfts_close