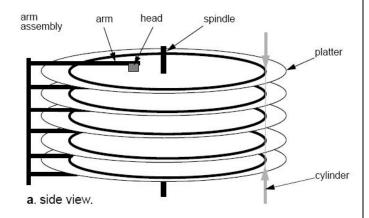
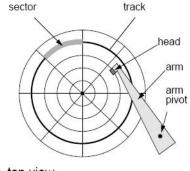
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 최대성
c3d4s19@naver.com

파일 시스템

하드디스크의 물리적 구조





b. top view.

Track

-> 디스크 상에 있는 1개의 데이터 인터럽터 영역 (보통 트랙 1매에 200~750 Cylinder 존재)

Sylinder

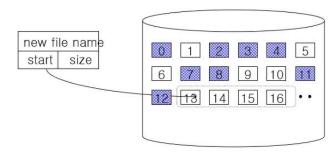
-> 세로 방향의 Track의 집합, Head이동 없이 읽혀질 수 있는 트랙군 의미

Sector

-> 각 Track을 일정한 길이로 분할 했을 때 일부분 (보통 1Sector = 512byte) 하드디스크 데이터 할당 방법

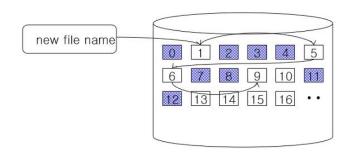
하드디스크는 4Kbyte씩 가상의 디스크 블록으로 나누어 데이터를 관리한다.

1. 연속 할당 기법



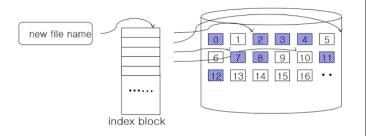
-> 연속 할당의 경우 데이터를 접근하는 데 걸리는 물리적 탐색 시간(Seek Time)을 줄일 수 있어 속도가 빠르지만 공간을 효율적으로 사용하기 힘들다

2. 블록체인 할당 기법



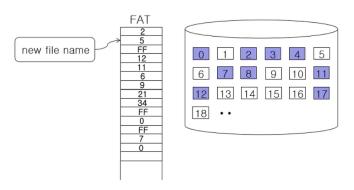
-> 같은 파일에 속한 디스크 블록들을 Linked list방식으로 연결해서 공간을 효율적으로 사용하지만 중간 블록 하나가 유실되는 경우 나머지 데이터까지 모두 잃게된다는 단점이 있다.

3. 인덱스 블록 기법



-> 블록에 대한 위치 정보를 기록한 인덱스 블록을 이용하여 필요한 데이터만 읽을 수 있지만 인덱스 블 록이 유실되면 파일 전체 데이터가 소실된다

4. FAT(File Allocation Table) 기법



-> 블록 위치를 FAT이라는 자료구조에 기록한다

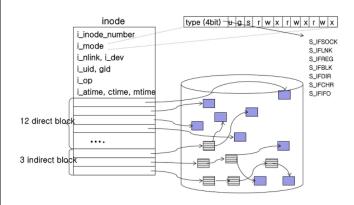
FAT에는 다음으로 읽어야할 디스크 블록 번호이기도
하며 다음 FAT 인덱스 위치인 숫자값이 들어있다

FF는 파일의 끝을 의미하고 0은 FREE상태를 의미한다

마찬가지로 FAT구조 유실시 모든 파일 유실될 수 있다

따라서 요즘 대부분의 FAT구조 파일 시스템은 FAT내용
을 중복하여 관리한다

5. inode 구조



i_blocks : 해당 파일의 데이터 블록의 갯수

i_mode: 파일의 속성 및 제어 정보

i links count : inode를 가르키는 파일수 또는 링크수

i_uid, i_gid: 파일을 생성 유저, 그룹 ID

i_atime, i_ctime, i_mtime: 파일 접근, 생성, 수정 시간

i_blocks 필드는 12개의 direct block과 3개의 indirect block으로 나뉜다

direct block은 실제 파일의 내용을 담은 디스크의 데 이터 블록을 가리키는 포인터이다

indirect block은 인덱스 블록(디스크 블록을 가르키는 포인터 테이블)을 가르키는 포인터이다

첫번째 indirect block 인덱스 블록 1개 가짐 (4KB)

두번째 indirect block은 인덱스 블록 2개 가짐 (4MB)

세번째 indirect block은 인덱스 블록 3개 가짐 (4GB)