

Kernel 5

노트북: SW

만든 날짜: 2018-04-11 오전 10:10

수정한 날짜: 2018-04-11 오후 11:44

작성자: fstopdg

2018. 4. 11 수 - 35회차

과정 : TI, DSP, Xilinx Zynq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈
gcccompil3r@gmail.com
Stu. 정상용
fstopdg@gmail.com

하드디스크를 관리하는 파일시스템 & 가상파일 시스템

파일시스템의 종류에 대하여 알아보고, 각 파일시스템의 구조와 동작에 대하여 알아본다.(디렉토리 엔트리를 통하여)

Memory management techniques and file system are a software which administrates the memory devices(memory and disk)

Chapter 5 : 파일시스템과 가상 파일시스템

-> file system : Everything is a file.

1. 파일시스템 일반
2. 디스크 구조와 블록 관리 기법
3. FAT 파일시스템
4. inode구조
5. Ext2 파일시스템
6. Ext3 파일시스템과 Ext4 파일시스템
7. 가상 파일시스템(Virtual File System)
8. 태스크 구조와 VFS 객체
- PASS
9. 파일시스템 제어 흐름 분석

1. 파일시스템 일반

->

Memory management techniques : main storage(memory)

File system : auxiliary storage(hard disk)

-> There are 2 focuses.

We should think about fragmentation in disk!

The difference between memory and disk are naming! (i.e Memory management techniques and file system are the same software w/o naming)

cf. file system has a lot of kinds.

-> file system - input : name, output(return) : data i want.

which contents do file system save in disk : meta data(inode, super block) & user data(real content)

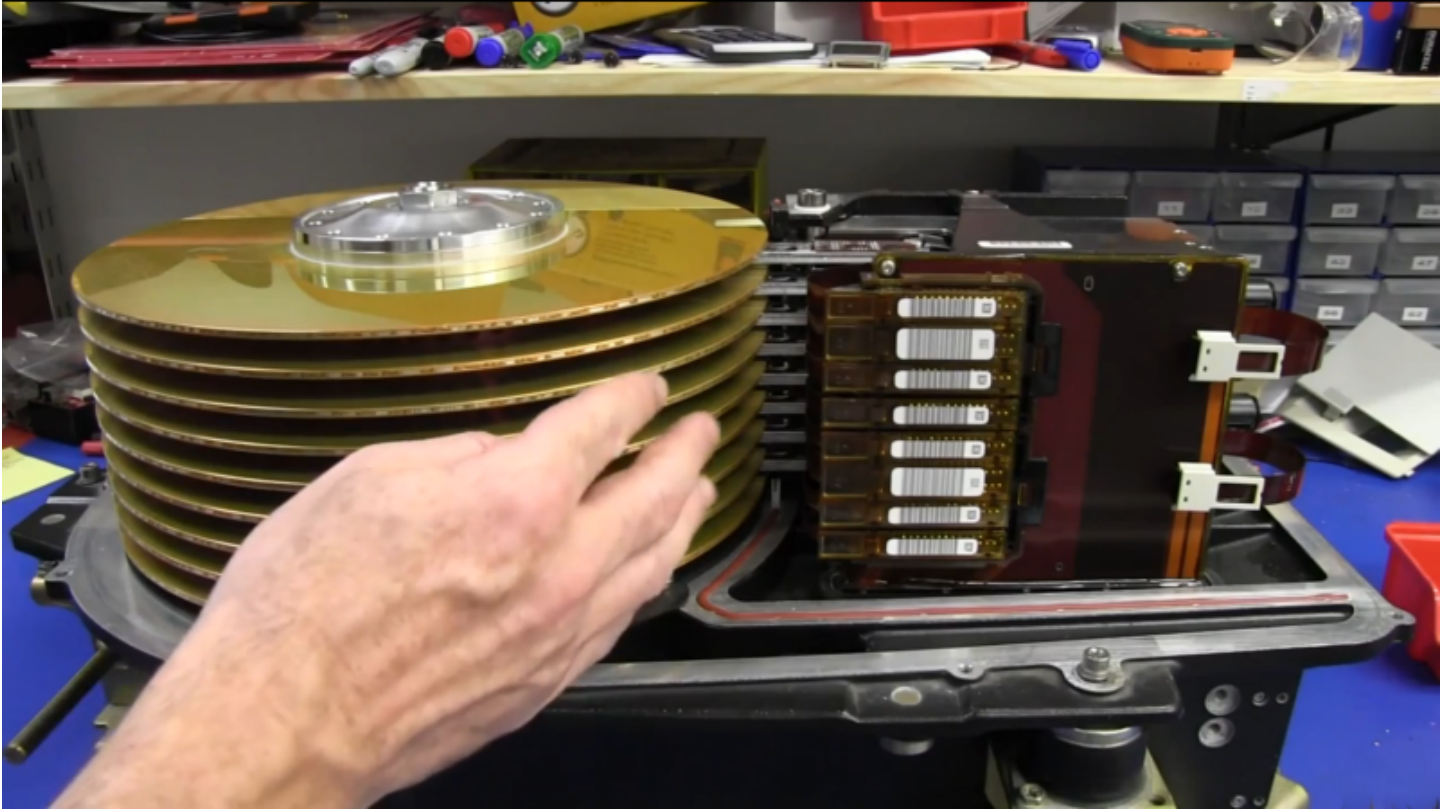
2. 디스크 구조와 블록 관리 기법

-> A physical structure of disk

component : plotter(track, cylinder, sector : 512byte), arm, head

upper and bottom of plotter can be read by head for efficient.

-> if we know about components, we can find out a property of the disk.(i.e size)



ref. <https://hackaday.com/2012/12/07/250000-hard-drive-teardown/>

-> A time that disk accesses to a data.

seek time, rotational latency, transmission time

seek time : 헤드가 트랙을 찾아가는데 시간이 오래 걸림(3가지 시간 중 제일 느림)

rotational latency : 자기장을 발생해야 하므로 회전이 빨라야함(disk : 자기장 기반)

transmission time : 전기적신호이므로 제일 빠름

-> file system views a disk as logical disk block.(Not physical structure like physical memory)

-> Each disk block gets its logic number, 0, 1, 2, 3...

-> Because of a size of disk block, there is a bottleneck. To solve this problem, the size is been increasing nowadays.

-> There are 2ways to allocate a disk block : sequential & non-sequential

sequential : fast, not efficient(reason : it does not need sseek)

non-sequential : slow, efficient

cf. 디스크 조각 모음 : 공간재배치하므로 공간효율이 올라가고 속도도 빨라진다.

-> 연속할당 기법 : 파일의 크기가 증가하였지만 연속하여 디스크 블록을 할당할 수 없는 경우 문제가 생긴다. 연속으로 할당해 주기 위하여 기존에 있는 파일을 다른 위치로 옮겨버릴 경우 비효율적인 비용을 처리해야한다.(i.e 과거 인덱스가 없는 배열로 queue를 구현할 경우, 특정 공간의 데이터를 삭제해주는 상황이 존재하였다. 삭제해준 공간에 뒤의 데이터를 하나씩 앞으로 옮길경우 굉장히 비효율적인 작업이었다. 빈 공간은 그대로 두고 빈 공간을 따로 관리하는 것이 훨씬 효율적이였다. 이것과 같은 상황이라고 보면된다.)

-> 불연속 할당 방법시 비어있는 공간을 관리해야 한다. 이를 위한 방법으로는 3가지가 존재한다.

블록체인 기법, 인덱스 블록 기법, FAT기법

3. FAT 파일시스템

-> super block을 통하여 /에서 하위항목을 하나씩 찾아가는 기법

4. inode 구조

파티션 보는 방법 터미널에서 df-h 또는 mount

ext계열은 서로 호환이 된다(ext2, ext4는 서로 보완가능)

vi -t ext2_indoe

vi -t ext3_indoe :(3이상부터) 지워져도 싹 다 복원이 가능함

-> ext2 : 4tb가 한계
-> but 하루에 pecta씩 싸인다.
-> 우리는 디스크용량을 무한대로 하자. 그것이 바로 ext4이다.

vi -t msdos_dir_entry : fat 굉장히 간단한 구조체

vi -t ext2_sb_info

ext4 : journaling 기법 : 서버에서도사용하므로 : 깨져도 복원가능 절대걱정없음

파일시스템은 블록으로 나누어져있지않지만, 메모리에 올려야해서 디스크블록존재.

디스크블록 사이즈 : 4K 메모리에 올려야하므로