

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 윤연성

whatmatters@naver.com

버디

슬랩

```
ys@ys-Z20NH-AS51B5U:~/my_proj/kernel/linux-4.4$ vi -t ext4_inode
```

```
ys@ys-Z20NH-AS51B5U:~/my_proj/kernel/linux-4.4$ vi -t ext4_inode
```

파티션 보는방법 df- h

mount

파일시스템과 가상 파일 시스템

리눅스가 제공하는 대표적인 객체 2 가지
태스크와 파일

태스크 = **CPU** 를 추상화시켜 프로그램에게 생명을 부여

파일 = **디스크**를 추상화시켜 프로그램에 장소를부여

1. 파일시스템 일반

기억장치는 한정적자원 RAM, 하드디스크

그렇기에 메모리 관리 기법과 파일시스템 모두 내부/외부 단편화를 최소화 해야
함

두 기법모두 자신만의 할당/해제 정책이 존재

메모리 관리기법과 파일시스템간의 차이점? = 해답은 “이름” 이라는 특성임

결국 파일시스템은 사용자에게 이름이라는 속성으로 접근되는 추상적인 객체인
파일의 개념을 제공

영속적인 객체의 저장을 지원하는 소프트웨어

즉 “이름” 을 입력받아 해당 데이터를 리턴해주는 소프트웨어가 파일시스템

메타데이터 : 파일의 속성 정보나 데이터 블록 인덱스 정보

유저데이터 : 사용자가 실제 기록하려 했던 내용

사용자가 파일을 하나 생성하고 약속시간을 저장하면 (4KB 문자열)

사용자가 위 내용을 확인하기 위해 읽으려하면 사용자입장에서는 “약속” 이라는 이름을 통해 접근하고싶어할것이기 때문에 파일시스템은 처음데이터를 기록할 때 실제데이터 + 파일이름 + 파일의 생성시간 + 실제데이터블록인덱싱 정보를 기록 해둔다

2. 디스크 구조와 블록관리기법

디스크는 원판, 팔, 헤드로 구성됨

원판에는 원형의 트랙이 존재하고 이 트랙은 몇개의 섹터로 구분됨

섹터는 디스크의 데이터를 읽고 쓰는 기본단위

일반적 크기는 512 바이트임

디스크에서 데이터를 접근하는데 걸리는 시간은

탐색시간, 회전지연시간, 데이터 전송시간으로 구분됨

탐색시간이 가장오래걸림(상식적으로도)

디스크 블록의 할당과 회수방법

디스크블록 할당은 연속할당과 불연속할당 (자료구조처럼 연결)

연속할당 = 파일에게 연속된 디스크블록을 할당하는 방법

불연속할당 = 파일에 속한 디스크블록들을 연속적으로 저장하지 않음

-파일이 속한 디스크블록이 어디에 위치하는지에 대해 정보를 기록해야함

블록체인기법 : 디스크블록들을 체인처럼 연결(링크드리스트)해서 다음블록의 위치를 기록

첫번째 디스크 블록에 가면 포인터를 이용해 다음블록의 위치를 찾을수 있음

단점 : 파일의 끝 부분을 읽으려는 경우 어쩔수없이 앞부분의 데이터블록을 읽어야되고

중간에 블록이 유실되면 나머지 데이터 모두읽게 된다

이덱스블록기법 : 블록들에 대한 위치정보를 기록한 인덱스 블록을 따로 사용하는 방법임

단점 : 인덱스 파일 유실시 파일 데이터 전체가 소실되며 인덱스 블록을 위한 별도의 공간이 필요하다

파일이 커져서 인덱스 블록이 가득차면 방법이없음

FAT 기법 : 같은 파일에 속한 블록의 위치를 FAT 라는 자료구조에 기록해 놓는 방법 FAT 기법은 파일시스템 전체적으로 하나의 FAT 존재

FF 는 파일의 끝을 의미

inode

ext2 , 4 아직 못했습니다 π