

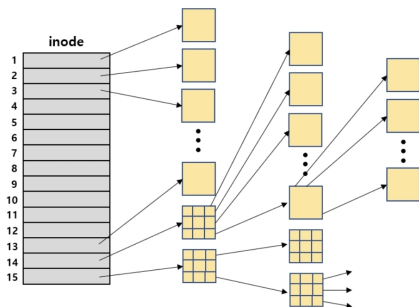
ext4 파일 시스템

0. ext 파일 시스템

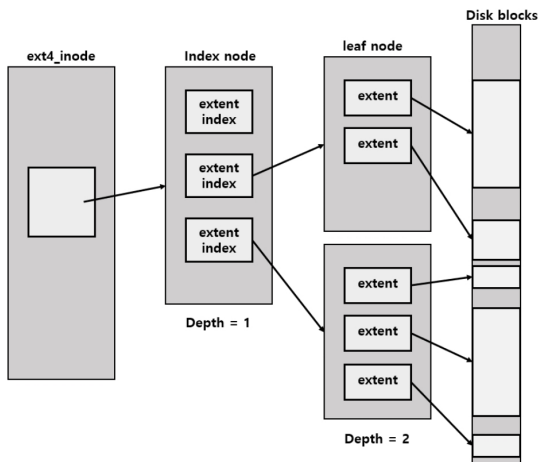
각각의 운영체제들을 저마다 파일들을 관리하는 파일 시스템을 채택하고 있다. 대표적으로 윈도우는 NTFS를, 맥 os는 APFS를 사용한다. 리눅스의 경우 ext 파일시스템을 사용한다. 대부분의 안드로이드 스마트폰과 IOT 기기의 경우에 리눅스를 기반으로 돌아가기 때문에 역시 ext 파일 시스템을 많이 사용하게 된다. 즉 한마디로 ext 파일 시스템이 널리 쓰인다는 것이다.

1. ext 의 역사

우선 1993년 나온 ext2부터 알아볼 필요가 있다. 리눅스 운영체제에서 사용할 수 있는 파일 시스템으로, inode 하나 당 하나의 파일을 저장한다. 가장 중요한 특징으로는 파일을 저장할 때, Block Mapping 방식을 사용한다는 것이다.

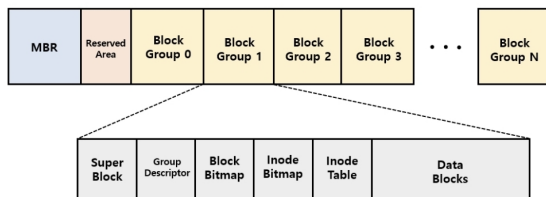


위의 그림처럼 inode 1번부터 12번까지는 하나의 파일에 매핑을 한다. 하지만, 실제 시스템에서 사용하는 파일의 수는 수백, 수천개가 넘어가므로 이렇게 표현해서는 역부족이다. 따라서 13번 inode부터는 Block Mapping 방식을 사용하여 하나의 inode가 여러개의 블록을 가리키도록 하여 많은 파일들을 효율적으로 저장할 수 있도록 하였다. 다음으로 ext3에 대하여 알아보자. ext3는 ext2와 거의 유사하지만 저널링(Journaling) 기능이 생겼다는 특징이 있다. 저널링 기능은 파일 시스템의 변경 사항을 추적하기 위해서 만들어진 기능으로, 삭제된 파일을 복구하는데 사용되기도 한다. 마지막으로 ext4에 대하여 알아보자. ext4는 ext3와 거의 유사하지만, Block Mapping 방식이 아닌, extents 트리를 이용하여 파일을 관리한다.



위의 그림이 extents 구조를 나타낸 것을 나타낸 것으로, extents 구조를 활용하면 depth가 2를 넘어가지 않아도 56TB의 파일을 관리할 수 있다. 물론 ext2, ext3, ext4가 모두 조금씩 차이점이 있지만, ext4만 제대로 이해한다면 나머지는 쉽게 이해할 수 있다.

2. ext4의 구조



위의 그림이 ext4의 전체적인 구조를 나타낸 것이다. 부팅과 파티션 관리를 위한 MBR영역이후에 여러개의 Block Group (그냥 Group으로 표현하기도 함) 으로 나누어진 형태이다. 각각의 Block Group 속에는 6개의 영역으로 나뉘어진다. 앞서 하나의 파일시스템이 여러개의 Block Group으로 나누어진다고 했다. 따라서, 여러개의 Block Group 마다 각각 6개의 영역이 존재한다. 이 6개의 영역을 하나씩 살펴보자.

1) Super block Super block은 전체적인 파일 시스템의 정보를 담고 있다. 총 inode 개수, 총 블록 개수, 블록 라벨명, 블록 사이즈 등의 전체적인 파일 시스템의 틀을 규정하고 있다고 보면 된다. Super block은 여러개의 Block Group에 모두 동일한 값으로 저장되어 있다. 참고로 Block Group과 Block과 Group을 혼동하기 쉬운데, Block Group = Group이고, Block은 전혀 다른 개념이다. Block은 전체 파일 시스템의 눈금자라고 생각하면 쉽다. 예를 들어 Block의 크기가 1024bytes라고 하면 (Super block에 크기가 나와있다) 전체 파일 시스템을 1024bytes 단위로 쪼개놓고 "45번째 블록" 이럴 경우, 45번째에 있는 block을 찾아가면 된다.

2) Block descriptor table Block descriptor table은 각 Block Group에 있는 Data block bitmap, inode bitmap, inode table의 위치를 알려준다. 각각의 Block Group에 포함된 위치를 알려주므로, Block Group의 개수만큼의 엔트리가

존재한다. Block descriptor table 역시 모든 Block Group에 같은 값으로 저장되어 있다.

3) Data block bitmap, inode bitmap 이 영역은 현재 inode와 data의 할당 정보에 대한 정보를 담고 있다. 할당되어 있으면, 1로 할당되어있지 않으면 0으로 표현한다.

4) inode table 각 파일의 inode 정보가 표 형식으로 배열되어 있다. ext4의 경우 기본적으로 하나의 inode의 크기는 256bytes이다. 각 inode에는 접근시각, 수정시각, 삭제시각, extents 트리 데이터, extents 할당 블록 개수, 데이터의 위치 등의 정보를 담고 있다. 즉, 실제 파일이 어떻게 생겨먹은 놈인지, 실제로 어디에 저장되어 있는지를 알려주는 표지판이라고 생각하면 된다.

5) Data Blocks 실제 데이터들이 저장되어 있는 곳이다. Block 단위로 계산을 하며, inode 정보를 통해 파일의 실제 데이터 위치를 파악한다.

복호화 관련 정보는 많이 못 찾았음...영영슨