원문 : <https://digital-forensics.sans.org/blog/2010/12/20/digital-forensics-understanding-ext4-part-1-extents/comment-page-1/?reply-to-comment=15172>

# fsstat /dev/mapper/elk-home

명령어를 통해 알아낼 수 있는 것 :

1 inode의 블록 그룹 위치

2 inode의 블록 그룹 위치 주소

Ext4의 중요한 변화 : inode의 크기가 256Bytes로 커졌다.

즉, 4K 블록 당 16개의 inode가 존재한다.

Byte 4-7은 low-order 32bit로 파일 사이즈 표시

하지만 Ext4는 블록 포인터 대신 extents를 사용하기 때문에 40-99까지의 60Bytes를 extent 정보를 담는데 사용한다.

extent 구조체 크기는 12bytes이기 때문에 inode당 최대 5개의 extents가 존재할 수 있을 것으로 예상한다.

하지만 첫 12bytes는 extent header 구조체가 차지하기 때문에 하나의 inode당 최대 extents 개수는 실제로는 4개이다.

extent header 값 :

Bytes 40-41: Magic number (0xF30A = 62218)

42-43: Number of extents (0x0001 = 1)

44-45: Max number of extents (0x0004 = 4)

46-47: Depth of tree (0x0000 = 0)

48-51: Generation ID (0x00000000 = 0)

extent에 대해 알아야 할 것들:

Bytes 52-55: Logical block number (0x0000) : 파일 시작점 기준 extent 시작 위치

56-57: Number of blocks in extent (0x0001) : 이 extent가 포함하는 블록 개수

58-59: Upper 16 bits of physical block address (0x0000)

60-63: Lower 32 bits of physical block address (0x003A883F)

: 이 6비트는 extent의 첫번째 블록의 물리적 블록 번호, 즉 extent가 실제 디스크의 시작 위치

파일이 삭제되어도 데이터 블록은 clear되지 않는다.

하지만 inode에서는 변화가 일어난다 :

파일 사이즈 값이 0으로 설정됨

extent header의 extents 개수가 0으로 설정됨

extent 자신도 clear됨 : 첫번째 블록의 물리적 블록 주소와 extent의 길이를 잃어버렸다.

즉, 삭제된 파일을 복구하는데 필요한 metadata가 inode에 남아있지 않다.