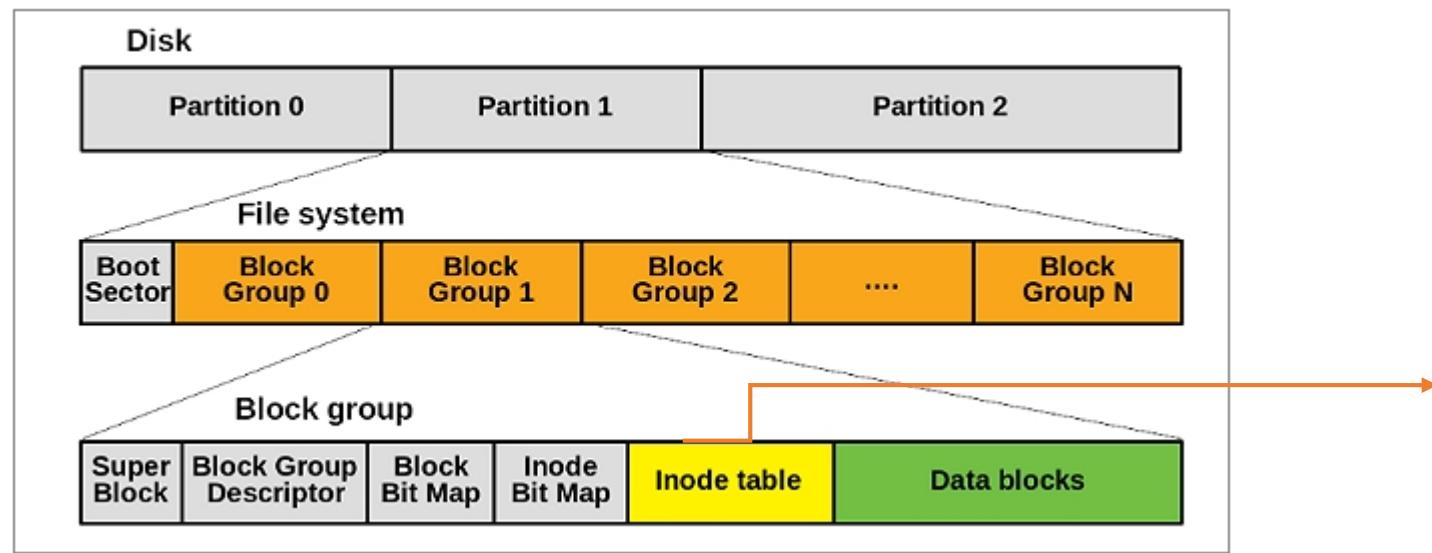
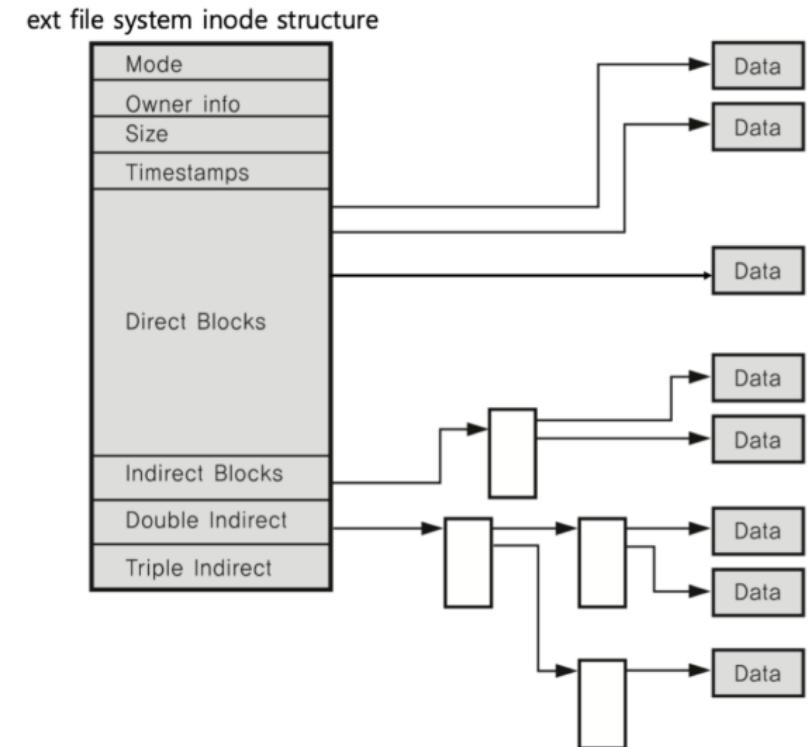


# EXT2 파일시스템 구조



#N inode in inode table



파일의 이름은 inode 자료구조에 없다!!!

# 디렉토리 파일의 내용

(디렉토리 파일의 data block 내용)

```
Directory File (on disk)  
|  
| [dirent #1] inode = 1234, name=". ", rec_len |  
| [dirent #2] inode = 1234, name=".. ", rec_len |  
| [dirent #3] inode = 2001, name="a.txt", rec_len |  
| [dirent #4] inode = 2002, name="sub/", rec_len |  
| [dirent #5] inode = 2003, name="bin", rec_len |  
| ... |
```

각 항목(dirent)이 다음 정보를 가짐:

```
rust  
  
struct ext4_dir_entry {  
    __le32 inode;           ← 이 이름이 가리키는 inode 번호  
    __le16 rec_len;         ← 디렉터리 엔트리 길이  
    __u8 name_len;          ← 파일명 길이  
    __u8 file_type;        ← REG/DIR/LNK 등  
    char name[];           ← 파일명 문자열  
}
```

```
struct inode  
|  
| i_mode : 파일 타입 + 퍼미션 | (S_IFREG, S_IFDIR, rwxr-xr-x 등)  
| i_uid : 소유자 UID  
| i_gid : 그룹 GID  
| i_size : 파일 크기 (bytes)  
| i_blocks : 사용 중인 블록 수  
| i_atime : access time  
| i_mtime : modify time  
| i_ctime : change time  
| i_nlink : 하드링크 개수  
| i_block[] : 데이터 블록 번호 리스트  
| i_sb : struct super_block * | ← 어느 파일 시스템에 속하는지  
| i_op : struct inode_operations* | ← create, lookup, link, unlink...  
| i_fop : struct file_operations* | ← open 후 사용할 파일 연산  
| i_mapping : struct address_space * | ← 페이지 캐시 / 블록 매핑  
| ... |
```

**Mission:**

>> ls -alF /home/adio2000/test.c

ls 프로그램 내에서 위 파일의 정보를 프린트해야 함. 이때 위 파일의 정보를 찾는 과정은?

# 각 프로세스의 파일과 관련된 자료구조들

