(*) core data structure derivation & their purpose with example.

→ superblock
→ attributes :

① U32 magic // for identifying this as my filesystem
② U32 block_size // size of each block in disk
③ U32 free_blocks // available blocks for data
④ U32 total_inodes // total number of inodes
⑤ U32 free_inodes // available inodes
⑥ U32 inode_table_block // where inode table starts at disk.
⑦ U32 block_bitmap_block // where bitmap starts on disk
⑧ U32 data_block_start // where actual file data begins.
⑨ U32 total_blocks

U8 reserved[]; // padding to fill exactly 1 block

=> size of 4 Bytes size
∴ 9 × 4 = 36 Bytes
∴ BLOCK_size - 36 = reserved bytes
↳ for future extension.

property need
magic → filesystem identification helps in that
Block_size → for all operation need for computation

total_blocks (4B) → Bounds checking like allocating blocks know when to stop searching

free_blocks (4B) → fast space availability checks for operations like create_file, delete_file

total_inodes (4B) → Bound checking for operation allocate_inode()

free_inodes (4B) → fast node availability check
create_file check if we can
delete_file update free inode count

inode_table_block (4B) locating inode table on disk must needed for loading FS

Block_bitmap_block (4B) — " — locate free space on disk

data_blocks_start (4B) - know where data area begins to start searching for free blocks.

① Inode structure

→ attributes

```
U32   id          // unique identifier
U32   size        // file size in bytes
U32   direct [ptrs]   // max file ptrs
U8    is_dir      // Boolean flag
U8    used        // Boolean flag
      padding     // recalculated dynamically
U8    name [60]   // file filename length
U32   parent_id   // to make hierarchy like structure
```

| | |
|---|---|
| attributes | need |
| id (4B) | unique identification for files |
| size (4B) | know actual data length to avoid reading garbage. |
| direct [ptrs] | points to data block for making operation fasts such as read_data, write_data, delete_file. |
| parent (4B) | for navigating directory structure |
| is_dir (1B) | distinguish files from directories |
| used (1B) | Track allocation status. |
| name (60B) | human readable identification. |