Directory Structure

System holds directory information in data blocks, it holds directory name and inode number of that directory.

```
typedef struct Directory {
    uint16_t inode;
    char name[MAX_FILE_NAME_SIZE];
}Directory;
```

I-Node Structure

```
typedef struct INode
{
    char name[MAX_FILE_NAME_SIZE];
    uint8_t is_directory;
    uint16_t parent_inode;
    uint32_t size;
    uint8_t active_blocks;
    uint16_t disk_block[DISK_BLOCK_COUNT];
    int32_t last_access;
    uint16_t single_indirect;
    uint16_t double_indirect;
    uint16_t triple_indirect;
} INode;
```

Name: File name
is_directory: Is inode holds information for directory or a file
parent_inode: Parent i-node address, to acces directory of the files and directories.
size: Size of the contents.
active_blocks: Active block count.
disk_block: Address of direct blocks.
last_access: Last access (date - time) to a file or directory.
single_indirect: Address of single level indirect block.
double_indirect: Address of double level indirect block.
triple_indirect: Address of triple level indirect block.

Free Blocks

I used remaining area of the file system for free blocks and used bitmap to check is block filled or not.

Superblock

```c
typedef struct SuperBlock{
    uint16_t block_size;
    uint16_t block_count;
    uint16_t free_blocks;
    uint16_t inode_count;
    uint32_t block_start;
    uint32_t inode_start;
    uint8_t bitmap[2048];
} SuperBlock;
```

block_size: Size of each block.
free_block: Blocks that are free.
inode_count: I-node count
block_start: Starting address of block section.
inode_start: Starting address of i-node section.
bitmap: Bitmap to track blocks are empty or not (2048 * 8 = 16MB)

Part 3 Functions:

dumpe2fs:

```
emreoztrk@Yunuss-Air CSE312_Homework-3 % ./fileSystemOper test.dat dumpe2fs
Block size: 4
Block count: 4086
Inode count: 0
Block start: 40960
Inode start: 4960
Free blocks: 500
```

Prints system file information.