CSE 312 Operating Systems HW3 Part 1
Yakup Talha Yolcu - 1801042609

Filesystem represented with size of 16 Mega bytes which is **16777216 Bytes.**

```
Size of superblock:144
Size of 1 inode:32 Size of inodes:12832
Size of 1 block:4 Size of blocks:16760832
Remained empty space:3552
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v
```

File system consists of Super block,Inodes, blocks
File system is **16 MB** as needed in pdf

At first file system is like this

144 Byte SuperBlock	**16777072 Byte** Uninitialized Space

This numbers are for 4 KB for each block and 400 + 1 inodes

144 Byte SuperBlock	**3552 Byte** Empty Space	**12832 Byte** Inodes	**16760832 Byte** Blocks

```
typedef struct SuperBlock {
    uint32_t first_block_address;
    uint32_t first_inode_address;
    uint8_t block_bitmap[128];
    uint16_t block_size;
    uint16_t block_count;
    uint16_t inode_count;
    uint16_t file_count;
}SuperBlock;
```

In super block struct, I keep # of blocks, how many kb one block occupies, # of inodes, # of files in the filesystem, address of the first block, address of the first inode and a bitmap to keep track of free blocks

```
typedef struct File {
    uint16_t inode;
    char file_name[FILE_NAME_LEN];
}File;
```

My file structure is simple, it has an inode and a name

```
typedef struct Inode {
    uint16_t direct[DIRECT_COUNT];
    uint16_t singleI[INDEX1_COUNT];
    uint16_t doubleI[INDEX2_COUNT];
    uint16_t tripleI[INDEX3_COUNT];
    uint8_t link_count;
    uint8_t type;
    uint32_t size;
    int32_t last_access_time;

}Inode;
```

link count -> number of links to that inode
i_node_type -> file/directory/symbolic link
i_node_size -> size of file/directory/symbolic
last_access_time -> last access time to tha
direct_inode -> addresses of the blocks belong that inode
index_1_inode -> addresses of the single indirect blocks
index_2 inode -> addresses of the double indirect blocks
index_3_inode -> addresses of the triple indirect block

We have . and .. files for each inode because we need to go current and parent directories later.
I represented directories as files which I shown the structure.

all functions return 0 on success and prints erron message and exits on failure (given path does not exists, given file does not exists, etc..)
filesystem is global variable. It was giving segmentation fault when it is local variable. These functions are used depend on the given command respectively. Each of them calls specific functions.

mkdir_command -> makes a directory
rmdir_command -> removes the directory
dir_command -> lists the contents of the directory
dumpe2fs_command -> gives information about the filesystem

```
Inode *get_inode(uint8_t *, uint16_t)
```
Returns inode which is in the given place.

```
uint8_t *get_block(uint8_t *, uint16_t )
```
Returns glock which is in the given position

```
void check_available(uint8_t *, uint8_t, uint8_t)
```
checks for available space in the inode

```
uint16_t emptyinode_count(uint8_t *)
```
returns the # of empty inodes

```
uint16_t emptyblock_count(uint8_t *)
```
returns the # of empty blocks

```
void free_file_block(uint8_t *,  File *)
```
makes given file empty

```
size_t get_length_until_path(char *)
```
returns length of the string until see a ‘/’

```
Inode *find_inode_of_path(uint8_t *, char *)
```
returns the inode of the given path

```
File *getFile(uint8_t *,  Inode *, char *);
```
returns file corresponding to given inode and path

```
void find_parent_direcotry(char *, char *)
```
returns the parent directory of the given path

```
uint8_t get_file_block_number(uint8_t *,  Inode *, uint64_t, uint16_t *)
```
returns the block # of the given inode

```
uint8_t *get_file_block(uint8_t *,  Inode *, uint64_t)
```
returns the file block of given inode and position

```
File *get_last_nonempty_place(uint8_t *,  Inode *)
```
returns the last nonempty file

```
File *get_first_empty_place(uint8_t *,  Inode *)
```
returns the first empty file

```
File *get_file_in_directory(uint8_t *,  Inode *, uint64_t)
```
returns the file from given inode

```
uint16_t get_first_empty_block(uint8_t *)
```
returns first available block

```
uint16_t get_first_empty_inode(uint8_t *)
```
returns first available inode

I detected my directories by giving them a type.
If it is a file, type is 0
If it is a directory type is 1

Sample outputs:

```
DIR      64 Tue May 31 15:00:00 2022 ..
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v3/part3$ ./fileSystemOper mySystem.dat mkdir "/usr"
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v3/part3$ ./fileSystemOper mySystem.dat mkdir "/usr/ysa"
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v3/part3$ ./fileSystemOper mySystem.dat mkdir "/bin/ysa"
Could not find path: Success
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v3/part3$
```

```
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v3/part3$ ./fileSystemOper mySystem.dat dir "/"
DIR     128 Tue May 31 18:59:56 2022 .
DIR     128 Tue May 31 18:59:56 2022 ..
DIR      96 Tue May 31 18:59:09 2022 usr
DIR      64 Tue May 31 18:59:56 2022 a
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v3/part3$
```

```
talha@talha-X580VD:~/Masaüstü/CSE312 OS/HW3/v3/part3$ ./fileSystemOper mySystem.dat dumpe2fs
Inodes(free):397
Inodes(total):401
Block Size(kb):4
Blocks(free):4088
Blocks(total):4092
---
Inode:     0 DIR
Names:  (root)
Blocks: 0,
--- --- --- --- ---
Inode:     1 DIR
Names:  usr,
Blocks: 1,
--- --- --- --- ---
Inode:     2 DIR
Names:  ysa,
Blocks: 2,
--- --- --- --- ---
Inode:     3 DIR
Names:  a,
Blocks: 3,
---
Files:0
Directories:4
```

I could not implement write, del, read functions
Special requirements:
I implemented whole filesystem based on linux, so paths should be given like that :/usr/bin
I took one more arguments from the user , it was the inode count. Because without knowing inode count, we can't decide inode size and other fields.

part2:

```
./makeFileSystem 4 400 mySystem.dat
```

part3:

```
./fileSystemOper mySystem.dat mkdir "/usr"
```