CSE 312

HW 3

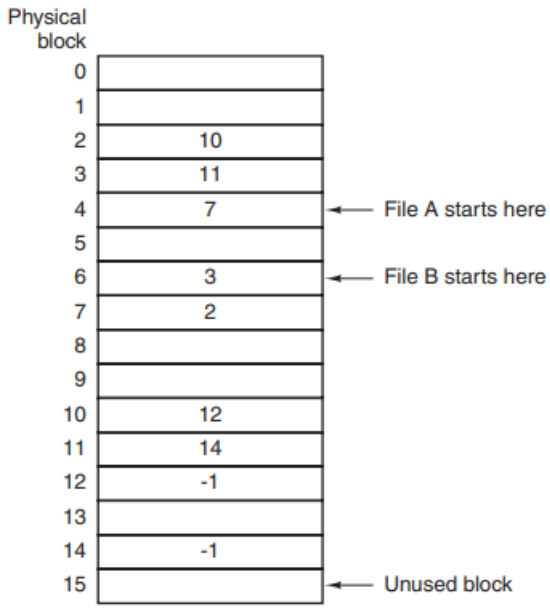
Burak Kocausta

1901042605

Contents

1- Part 1 – FAT 12 Design

- I designed my system as similar to the fat table in the book. Each block holds next block index in table for accessing the next block.



If block ends for that file or directory at some point next block is -1. It accesses next block with next block index.

Magic number	Root position	Block size	Number of blocks	Block count	Free block count	File count	Directory count

- My superblock contains those informations, it initalizes the file system according to those data.

superblock	blocks	Free blocks

- File system is simply like that, when it is started super block is read.

Block number	Next block	type	data

- Each block has its block number, next block information, type and data field.

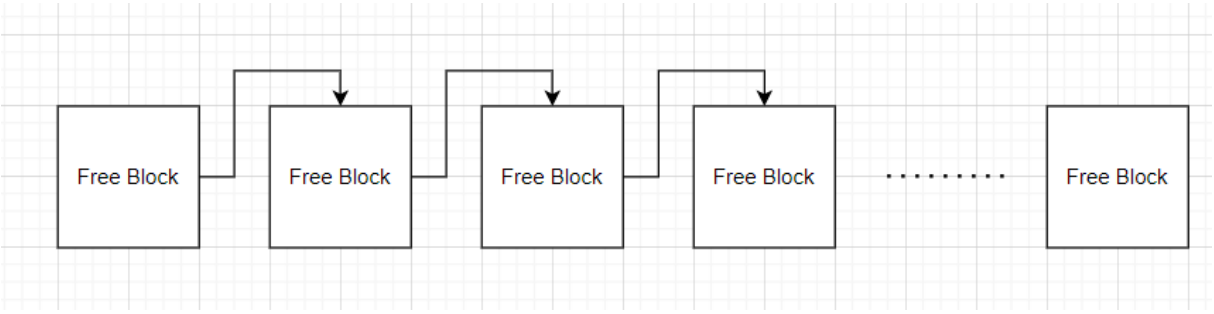
Block size	FAT-12
0.5 KB	2 MB
1 KB	4 MB
2 KB	8 MB
4 KB	16 MB
8 KB	
16 KB	
32 KB	

I defined block size partition size relation as same as this. Block sizes can only be 0.5, 1, 2, 4 KB. The number of blocks are determined according to that block size, Partition size relation. Other combinations are forbidden.

- I held a directory entry list in directory, and each directory entry is accessed by it. Every directory entry holds block number, and type. It is similar to the directory entry structure in the book. If a directory is changed or something, its block is accessed by directory entry block number. Directory entry can be a file or directory.

File name	Type	Length	Time/date	extension	Block Number

- My directory entry is like this. Each directory holds directory entries.



- I held free blocks as an idea of holding them with linked list. When a block is needed, one of the free block is pulled from that list, and used.

2- Part 2 – File System Creation and Design in C++

- I used C++ for designing the system. I used structures for representing block, superblock. To create file system, I wrote them to given file name for file system, and read from it. I defined another structures and classes, that will be used in operations.

```
struct superblock
{
    int magicNumber;
    size_t rootPosition;
    size_t blockSize;
    size_t numberOfBlocks;
    size_t blockCount;
    size_t freeBlockCount;
    size_t fileCount;
    size_t directoryCount;
};
```

Super block has magic number to recognize file system, it has all other information that I mentioned in first part.

```
struct block
{
    size_t blockNumber;
    ssize_t nextBlock;
    bool isDirectory;
    std::vector<char> data;
};
```

Block has block number, next block type information, and data part.

```
struct fat12_b
{
    superblock sb;
    std::vector<block> blocks;
    std::vector<block> freeBlocks;
};
```

This is the type that will be written to the file. Initially superblock must be read.

```cpp
struct directoryEntry
{
    std::string fileName;
    bool isDirectory;
    size_t length;
    std::string extension;
    std::string time;
    std::string date;
    size_t blockNumber;
};
```

```cpp
struct directory
{
    std::string directoryPath;
    size_t numberOfEntries;
    size_t blockNumber;
    std::vector<directoryEntry> entries;
};
```

- If a directory is needed, I used this structure. I traversed the entries, and if a block is needed, I used block number of that directory entry also I used its type.

```cpp
class fatTable
{

private:
    fat12_b fat12;
    bool isInitialized;
    std::string fileSystemName;
```

This is the fat table class that encapsulates all file system information and operations.

File system is created with makeFileSystem command.

0.5 KB block size

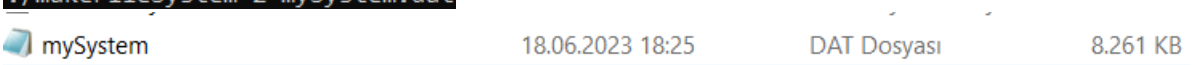
```
./makeFileSystem 0.5 mySystem.dat
```

1 KB block size

```
./makeFileSystem 1 mySystem.dat
```

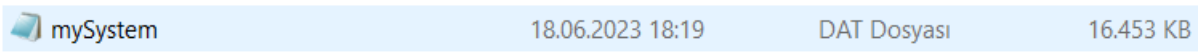
2 KB block size

```
./makeFileSystem 2 mySystem.dat
```



4 KB block size

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



Block size	FAT-12
0.5 KB	2 MB
1 KB	4 MB
2 KB	8 MB
4 KB	16 MB
8 KB	
16 KB	
32 KB	

Those are the sizes that are required.

3- Part 3 – File System Operations

```
// list the contents of the directory
bool opDir (std::string path);

// create a new directory
bool opMkdir (std::string path);

// remove a directory
bool opRmdir (std::string path);

// give information about file system
bool opDumpe2fs ();

// creates and writes data to the file
bool opWrite (std::string path, std::string fileNameToReadContent);

// reads data from the file
bool opRead (std::string path, std::string fileNameToWrite);

// removes a file
bool opDel (std::string path);
```

- File system operations are implemented as this.

```
W3/hw3$ ./makeFileSystem 4 fileSystem.data
W3/hw3$ man scanf > linuxFile.data
W3/hw3$
```

File system and linux file is created.

```
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data mkdir "\usr"
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data mkdir "\usr\ysa"
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data mkdir "\bin\ysa"
Error: Directory not found
```

It created the directories, and gave error because there are no bin directory.

```
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data write "\usr\ysa\file1" linuxFile.data

burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data write "\usr\file2" linuxFile.data
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data write "\file3" linuxFile.data
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data dir "\\"

Directory: \
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
DIR             18.06.2023 18:32              0       usr
FILE            18.06.2023 18:36              17798   file3
```

Files are created and root directory only has 1 file and 1 directory as expected.

```
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data dir "\usr\ysa"

Directory: \usr\ysa
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
FILE            18.06.2023 18:35              17798   file1
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data del "\usr\ysa\file1"
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data dir "\usr\ysa"

Directory: \usr\ysa
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
```

File 1 is deleted.

```
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data dumpe2fs

FAT-12 FILE SYSTEM

Block Size: 4096 KB
Block Count: 13
Number of Blocks: 4096
Number of Free Blocks: 4083
Number Of Files: 2
Number Of Directories: 3
```

There are 3 directories which are root, usr, ysa. And there are 2 files which are file2 and file3.

```
3$ ./fileSystemOper fileSystem.data read "\usr\file2" linuxFile2.data
3$ cmp linuxFile2.data linuxFile.data
3$
```

File is read, and they have no differences.

```
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data dir "\usr"

Directory: \usr
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
DIR             18.06.2023 18:33              0       ysa
FILE            18.06.2023 18:36              17798   file2
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data rmdir "\usr\ysa"
burak@LAPTOP-7FLC2OAS:/mnt/c/Users/burak kocausta/Desktop/cse312/homework assignments/HW3/hw3$ ./fileSystemOper fileSystem.data dir "\usr"

Directory: \usr
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
FILE            18.06.2023 18:36              17798   file2
```

ysa directory is deleted from usr directory.

4- Test Cases and Results

`./makeFileSystem 4 fileSystem.data` File System is created

```
./fileSystemOper fileSystem.data dir "\\"
```

```
Directory: \
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
```

Empty root directory

```
./fileSystemOper fileSystem.data mkdir "\usr"
./fileSystemOper fileSystem.data mkdir "\bin"
./fileSystemOper fileSystem.data mkdir "\usr\ysa"
./fileSystemOper fileSystem.data write "\usr\ysa\book.pdf" book.pdf
man printf > linuxFile.data
./fileSystemOper fileSystem.data write "\bin\file1" linuxFile.data
./fileSystemOper fileSystem.data dir "\\"
```

“\usr”, “\bin”, “\usr\ysa”, directories, and “usr\ysa\book.pdf” file is created book.pdf is approximately 7 mb.
Manual of printf is written to linuxFile.data, and it is stored in “\bin\file1”

```
Directory: \
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
DIR             18.06.2023 18:53              0       usr
DIR             18.06.2023 18:53              0       bin
```

```
./fileSystemOper fileSystem.data dir "\usr"
```

```
Directory: \usr
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
DIR             18.06.2023 18:53              0       ysa
```

```
./fileSystemOper fileSystem.data dir "\usr\ysa"
```

```
Directory: \usr\ysa
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
FILE            18.06.2023 18:48              7535476 book
```

```
./fileSystemOper fileSystem.data dir "\bin"
```

```
Directory: \bin
TYPE            LAST WRITE                    LENGTH  NAME
----            ----------                    ------  ----
FILE            18.06.2023 18:54              2629    file1
```

Directories and files are displayed.

```
./fileSystemOper fileSystem.data read "\bin\file1" linuxFile2.data
cmp linuxFile.data linuxFile2.data
./fileSystemOper fileSystem.data read "\usr\ysa\book.pdf" book2.pdf
cmp book.pdf book2.pdf
./fileSystemOper fileSystem.data dumpe2fs
```

file1 is read to linuxFile2.data, book is read to book2.pdf, and both have no differences.

```
FAT-12 FILE SYSTEM

Block Size: 4096 KB
Block Count: 1845
Number of Blocks: 4096
Number of Free Blocks: 2251
Number Of Files: 2
Number Of Directories: 4
```

Situation of file system.

```
./fileSystemOper fileSystem.data del "\usr\ysa\book.pdf"
./fileSystemOper fileSystem.data dumpe2fs
```

```
FAT-12 FILE SYSTEM

Block Size: 4096 KB
Block Count: 5
Number of Blocks: 4096
Number of Free Blocks: 4091
Number Of Files: 1
Number Of Directories: 4
```

After book.pdf is deleted, most of the blocks become free.

```
./fileSystemOper fileSystem.data rmdir "\usr\ysa"
./fileSystemOper fileSystem.data write "\usr\image" image.png
./fileSystemOper fileSystem.data read "\usr\image" image2.png
cmp image.png image2.png
./fileSystemOper fileSystem.data dir "\usr"
```

Image file is loaded to system, and no difference detected.

```
Directory: \usr
TYPE            LAST WRITE                      LENGTH  NAME
----            ----------                      ------  ----
FILE            18.06.2023 18:58                72869   image
```

```
./fileSystemOper fileSystem.data dir "\\"
```

```
Directory: \
TYPE            LAST WRITE                      LENGTH  NAME
----            ----------                      ------  ----
DIR             18.06.2023 18:53                0       usr
DIR             18.06.2023 18:53                0       bin
```

Directories and files displayed.

```
./fileSystemOper fileSystem.data write "\book.pdf" book.pdf
./fileSystemOper fileSystem.data dir "\\"
```

```
Directory: \
TYPE            LAST WRITE                      LENGTH  NAME
----            ----------                      ------  ----
DIR             18.06.2023 18:53                0       usr
DIR             18.06.2023 18:53                0       bin
FILE            18.06.2023 18:59                7535476 book.pdf
```

```
./fileSystemOper fileSystem.data dumpe2fs
```

```
FAT-12 FILE SYSTEM

Block Size: 4096 KB
Block Count: 1862
Number of Blocks: 4096
Number of Free Blocks: 2234
Number Of Files: 3
Number Of Directories: 3
```

After book.pdf added again, most of the blocks are reclaimed.

```
./fileSystemOper fileSystem.data write "\book.pdf" linuxFile.data
./fileSystemOper fileSystem.data read "\book.pdf" linuxFile3.data
cmp linuxFile.data linuxFile3.data
./fileSystemOper fileSystem.data write "\book.pdf" book.pdf
./fileSystemOper fileSystem.data read "\book.pdf" book3.pdf
cmp book.pdf book3.pdf
```

Some of the data is overwritten, but still no difference detected after reading and comparing.