GEBZE TECHNİCAL UNIVERSITY

CSE312

Operating Systems

Homework 2 Report

1. How to Run?

Open the terminal and navigate to the source where the makefile directory. Then, compile the program by typing "**make**" and to clean up generated files, use "**make clean**", which will remove the files generated. The commands I've tested:

- ./fat12_file_system makeFileSystem 1 mySystem.dat
- ./fat12_file_system mkdir mySystem.dat '\usr'
- ./fat12_file_system mkdir mySystem.dat '\usr\test'
- ./fat12_file_system write mySystem.dat '\usr\test\testfile.txt' "$(cat testfile.txt)"
- ./fat12_file_system read mySystem.dat '\usr\test\testfile.txt'
- ./fat12_file_system addpw mySystem.dat '\usr\test\testfile.txt' 'mypassword'
- ./fat12_file_system dir mySystem.dat '\usr'
- ./fat12_file_system dumpe2fs mySystem.dat

2. Directory Entry, Directory entry table and FAT12 (Part 1)

Directory File Entry

- Each directory entry is 64 bytes, sufficient for handling arbitrary-length filenames and passwords. Directory entries are stored in clusters, with each 1 KB cluster holding 16 entries.
- Permissions, protection status, and filename exceed check are managed within the file_attributes struct, occupying only 1 byte. This design is similar to Fig. 4.30 in the course book.
- File creation and last access times are stored using 4 bytes (2 bytes for date and 2 bytes for time), utilizing bitfields in file_time and file_date structs, resembling the design in Fig. 4.30.
- The reserved part of the directory entry struct is used for storing long filenames, ensuring filenames exceeding 8 characters can be accommodated effectively.

FAT12 Table and Block Management

- Maximum partition size is 4096 MB
- Block size is 1 KB, resulting in 4096 blocks
- FAT12 table holds 12-bit entries for each of the 4096 clusters, 6 KB (4096 clusters × 1.5 bytes per entry). Clusters marked as free using 0x000 in the FAT12 table.

3. Creating an empty File System

Superblock

The SuperBlock is implemented as a C++ struct and is read from or written to the file system using the readSuperBlock and writeSuperBlock functions.

```
// SuperBlock structure

typedef struct SuperBlock {

   uint32_t totalBlocks;
   uint32_t freeBlocks;
   uint32_t blockSize;
   uint32_t rootDirectory;
   uint32_t firstDataBlock;
} SuperBlock;
```

Free Blocks Bitmap

The free blocks bitmap is an array of bytes where each bit represents the allocation status of a block. It is managed by the readFreeBlocks and writeFreeBlocks functions.

FAT12 Table

The FAT12 table is an array that maps each block to the next block in a file. It is managed by the readFAT12 and writeFAT12 functions.

Directory and File Entries

Directories and files are represented by the DirectoryEntry struct, which includes fields for the filename, attributes, and pointers to the data blocks.

```
// File time structure

typedef struct file_time {
   uint8_t hours : 5;
   uint8_t minutes : 6;
   uint8_t seconds : 5;
} file_time;
```

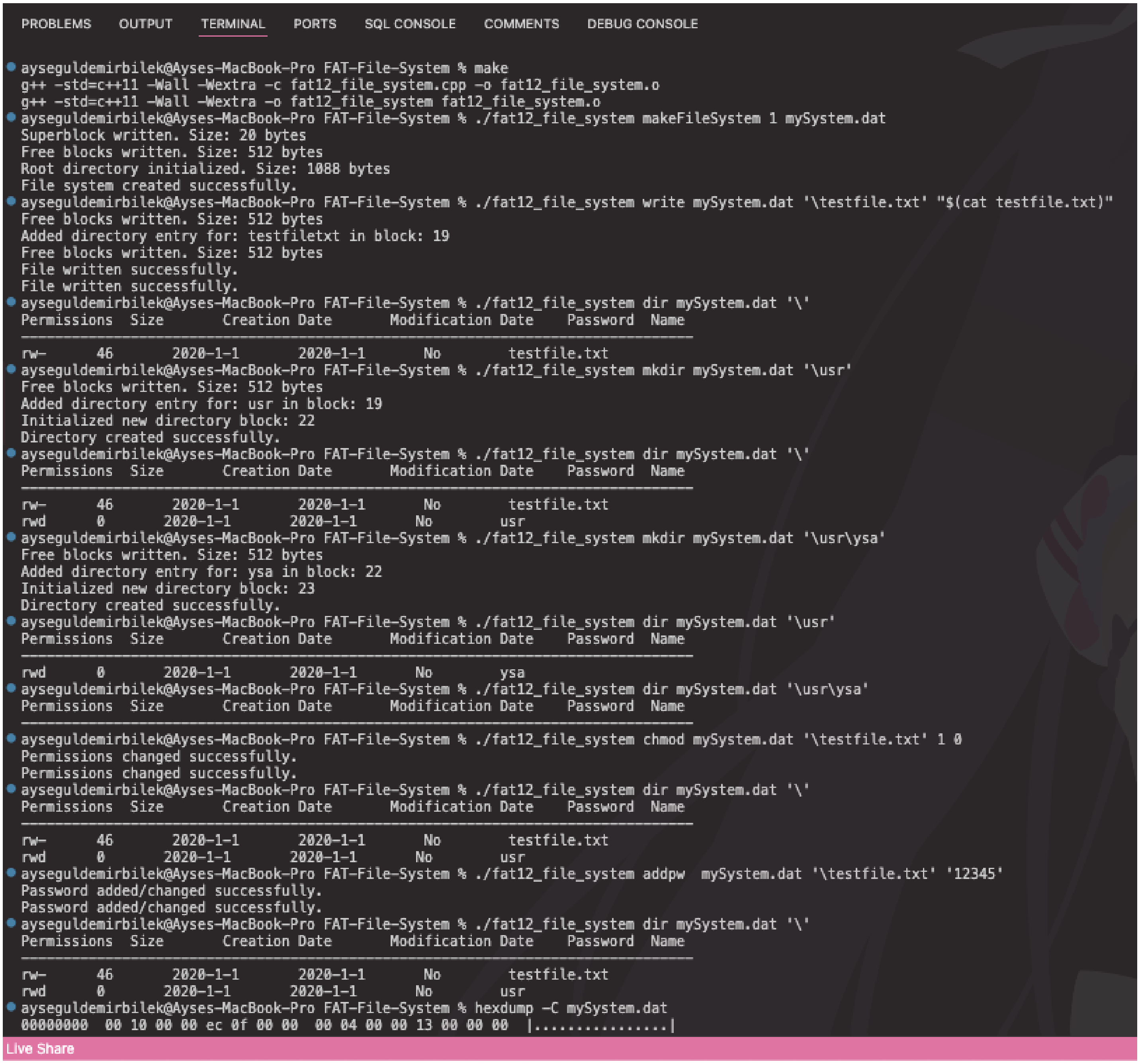
```
typedef struct file_date {
    uint8_t day : 5;
    uint8_t month : 4;
    uint16_t year : 7;
} file_date;
typedef struct file_attributes {
    uint8_t read_permission : 1;
    uint8_t write_permission : 1;
    uint8_t is_directory : 1;
    uint8_t is_protected : 1;
    uint8_t filename_exceeds : 1;
    uint8_t reserved : 3;
} file_attributes;
// Directory entry structure
typedef struct DirectoryEntry {
    char filename[8]; // 64 bits for the filename (8 characters with 8 bits each)
    char extension[3]; // File extension
    file_attributes attributes; // File attributes
    char reserved[18]; // Extended filename (if it exceeds 8 characters)
    file_time last_modificaton_time; // Time of last modification
    file_date last_modification_date; // Date of last modification
    file_time creation_time; // Time of creation
    file_date creation_date; // Date of creation
    uint16_t first_block_number; // The first block number of the file
    uint32_t file_size; // Size of the file in bytes
    char password[16]; // Password for the file
} DirectoryEntry;
```

Function Prototypes

```
void initializeSuperBlock(SuperBlock &superBlock, uint32_t blockSize);
void writeSuperBlock(std::ofstream &file, SuperBlock &superBlock);
void writeSuperBlock(std::fstream &file, SuperBlock &superBlock);
void readSuperBlock(std::ifstream &file, SuperBlock &superBlock);
void readSuperBlock(std::fstream &file, SuperBlock &superBlock);
void initializeFreeBlocks(uint8_t *free_blocks);
void writeFreeBlocks(std::ofstream &file, uint8_t *free_blocks);
void writeFreeBlocks(std::fstream &file, uint8_t *free_blocks);
void readFreeBlocks(std::ifstream &file, uint8_t *free_blocks);
void readFreeBlocks(std::fstream &file, uint8_t *free_blocks);
void initializeFAT12(FAT12Entry *fat);
void writeFAT12(std::ofstream &file, FAT12Entry *fat);
void writeFAT12(std::fstream &file, FAT12Entry *fat);
void readFAT12(std::ifstream &file, FAT12Entry *fat);
void readFAT12(std::fstream &file, FAT12Entry *fat);
```

```
uint32_t readFAT12Entry(const FAT12Entry *fat, uint32_t currentBlock);
vector<DirectoryEntry> readDirectoryEntries(fstream &file, uint32_t block);
int findFreeBlock(uint8_t *free_blocks);
```

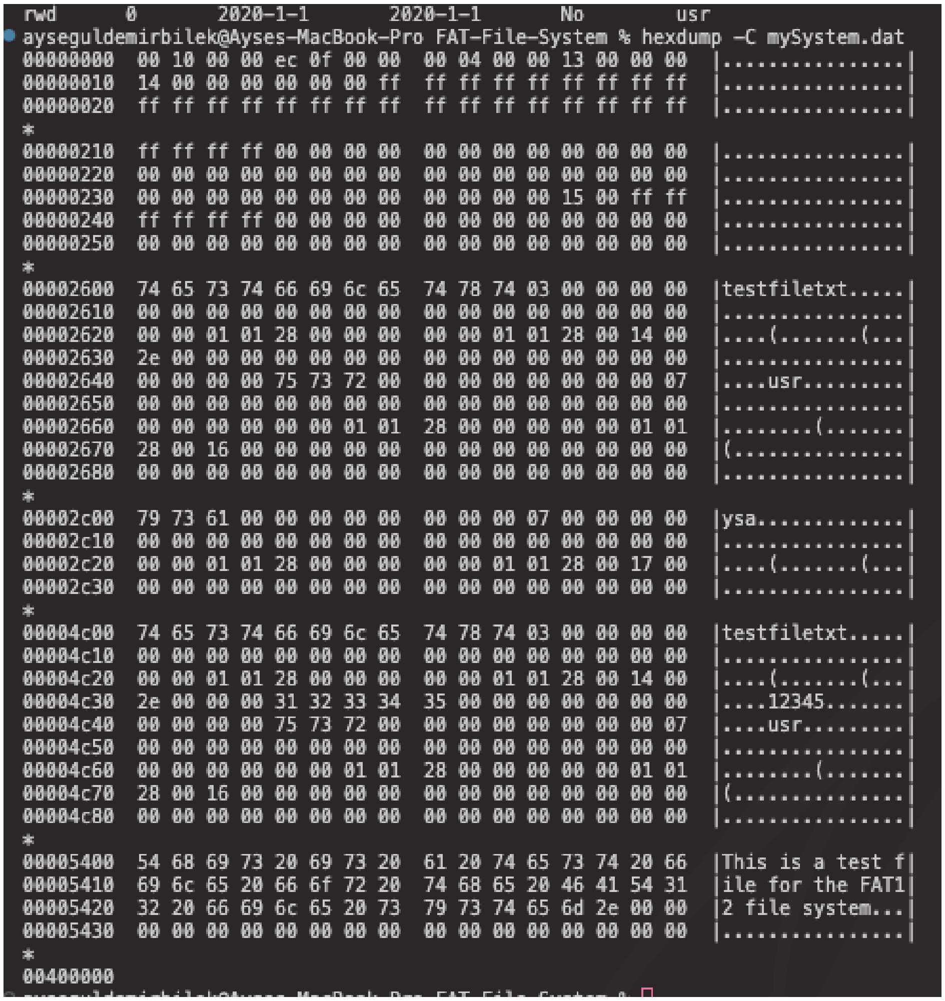
4. Example Output



```
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % make
g++ -std=c++11 -Wall -Wextra -c fat12_file_system.cpp -o fat12_file_system.o
g++ -std=c++11 -Wall -Wextra -o fat12_file_system fat12_file_system.o
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system makeFileSystem 1 mySystem.dat
Superblock written. Size: 20 bytes
Free blocks written. Size: 512 bytes
Root directory initialized. Size: 1088 bytes
File system created successfully.
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system write mySystem.dat '\testfile.txt' "$(cat testfile.txt)"
Free blocks written. Size: 512 bytes
Added directory entry for: testfiletxt in block: 19
Free blocks written. Size: 512 bytes
File written successfully.
File written successfully.
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system dir mySystem.dat '\'
Permissions  Size       Creation Date       Modification Date    Password  Name
--------------------------------------------------------------------------------
rw-      46       2020-1-1       2020-1-1       No        testfile.txt
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system mkdir mySystem.dat '\usr'
Free blocks written. Size: 512 bytes
Added directory entry for: usr in block: 19
Initialized new directory block: 22
Directory created successfully.
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system dir mySystem.dat '\'
Permissions  Size       Creation Date       Modification Date    Password  Name
--------------------------------------------------------------------------------
rw-      46       2020-1-1       2020-1-1       No        testfile.txt
rwd      0       2020-1-1       2020-1-1       No        usr
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system mkdir mySystem.dat '\usr\ysa'
Free blocks written. Size: 512 bytes
Added directory entry for: ysa in block: 22
Initialized new directory block: 23
Directory created successfully.
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system dir mySystem.dat '\usr'
Permissions  Size       Creation Date       Modification Date    Password  Name
--------------------------------------------------------------------------------
rwd      0       2020-1-1       2020-1-1       No        ysa
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system dir mySystem.dat '\usr\ysa'
Permissions  Size       Creation Date       Modification Date    Password  Name
--------------------------------------------------------------------------------
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system chmod mySystem.dat '\testfile.txt' 1 0
Permissions changed successfully.
Permissions changed successfully.
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system dir mySystem.dat '\'
Permissions  Size       Creation Date       Modification Date    Password  Name
--------------------------------------------------------------------------------
rw-      46       2020-1-1       2020-1-1       No        testfile.txt
rwd      0       2020-1-1       2020-1-1       No        usr
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system addpw  mySystem.dat '\testfile.txt' '12345'
Password added/changed successfully.
Password added/changed successfully.
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % ./fat12_file_system dir mySystem.dat '\'
Permissions  Size       Creation Date       Modification Date    Password  Name
--------------------------------------------------------------------------------
rw-      46       2020-1-1       2020-1-1       No        testfile.txt
rwd      0       2020-1-1       2020-1-1       No        usr
aysegulderirbilek@Ayses-MacBook-Pro FAT-File-System % hexdump -C mySystem.dat
00000000  00 10 00 00 ec 0f 00 00  00 04 00 00 13 00 00 00  |................|
```

Problems:

! Rmdir, del an read is not working as expected.



```
rwd      0        2020-1-1        2020-1-1        No         usr
aysegulddemirbilek@Ayses-MacBook-Pro FAT-File-System % hexdump -C mySystem.dat
00000000  00 10 00 00 ec 0f 00 00  00 04 00 00 13 00 00 00  |................|
00000010  14 00 00 00 00 00 00 ff  ff ff ff ff ff ff ff ff  |................|
00000020  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
*
00000210  ff ff ff ff 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000230  00 00 00 00 00 00 00 00  00 00 00 00 15 00 ff ff  |................|
00000240  ff ff ff ff 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000250  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00002600  74 65 73 74 66 69 6c 65  74 78 74 03 00 00 00 00  |testfiletxt.....|
00002610  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00002620  00 00 01 01 28 00 00 00  00 00 01 01 28 00 14 00  |....(.......(...|
00002630  2e 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00002640  00 00 00 00 75 73 72 00  00 00 00 00 00 00 00 07  |....usr.........|
00002650  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00002660  00 00 00 00 00 00 01 01  28 00 00 00 00 00 01 01  |........(.......|
00002670  28 00 16 00 00 00 00 00  00 00 00 00 00 00 00 00  |(...............|
00002680  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00002c00  79 73 61 00 00 00 00 00  00 00 00 07 00 00 00 00  |ysa.............|
00002c10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00002c20  00 00 01 01 28 00 00 00  00 00 01 01 28 00 17 00  |....(.......(...|
00002c30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00004c00  74 65 73 74 66 69 6c 65  74 78 74 03 00 00 00 00  |testfiletxt.....|
00004c10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00004c20  00 00 01 01 28 00 00 00  00 00 01 01 28 00 14 00  |....(.......(...|
00004c30  2e 00 00 00 31 32 33 34  35 00 00 00 00 00 00 00  |....12345.......|
00004c40  00 00 00 00 75 73 72 00  00 00 00 00 00 00 00 07  |....usr.........|
00004c50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00004c60  00 00 00 00 00 00 01 01  28 00 00 00 00 00 01 01  |........(.......|
00004c70  28 00 16 00 00 00 00 00  00 00 00 00 00 00 00 00  |(...............|
00004c80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00005400  54 68 69 73 20 69 73 20  61 20 74 65 73 74 20 66  |This is a test f|
00005410  69 6c 65 20 66 6f 72 20  74 68 65 20 46 41 54 31  |ile for the FAT1|
00005420  32 20 66 69 6c 65 20 73  79 73 74 65 6d 2e 00 00  |2 file system...|
00005430  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00400000
```