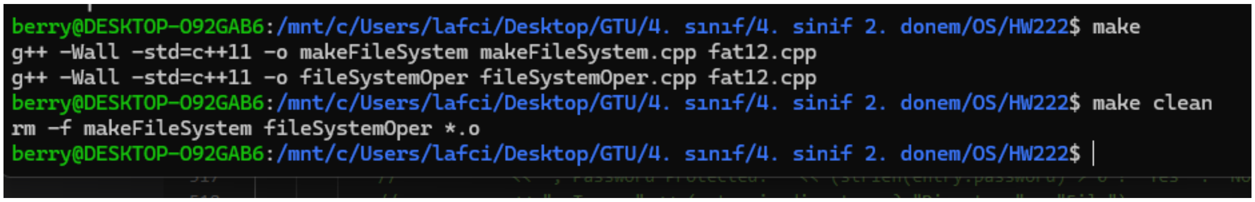
BERRU LAFCI - 1901042681

OPERATING SYSTEMS HW 2 REPORT

Compile and Clean:



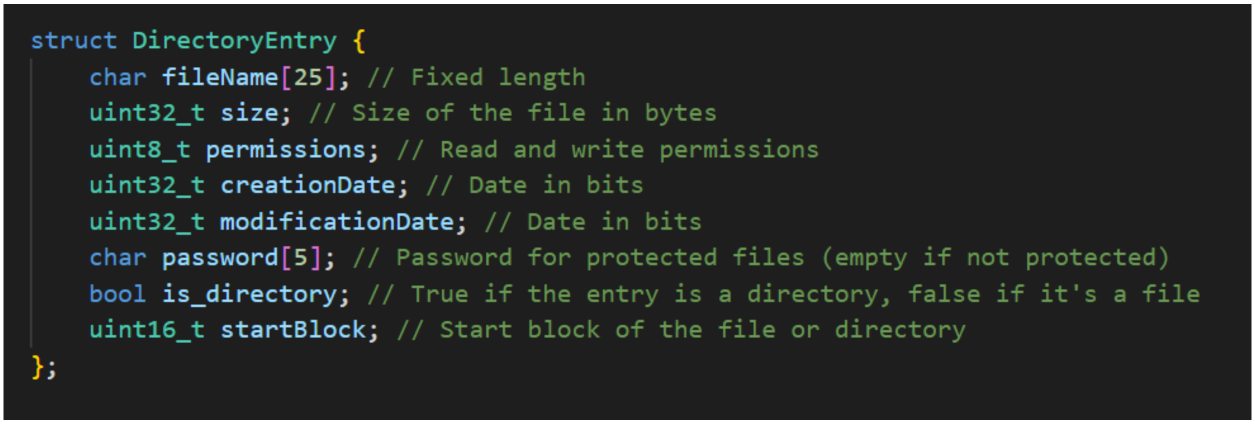
```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ make
g++ -Wall -std=c++11 -o makeFileSystem makeFileSystem.cpp fat12.cpp
g++ -Wall -std=c++11 -o fileSystemOper fileSystemOper.cpp fat12.cpp
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ make clean
rm -f makeFileSystem fileSystemOper *.o
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```

PART1:

1. Directory Table and Directory Entries

The directory table is a collection of DirectoryEntry structs stored in the file system. Each DirectoryEntry represents a file or directory within the file system and contains metadata about that file or directory.

DirectoryEntry Structure:



```
struct DirectoryEntry {
    char fileName[25]; // Fixed length
    uint32_t size; // Size of the file in bytes
    uint8_t permissions; // Read and write permissions
    uint32_t creationDate; // Date in bits
    uint32_t modificationDate; // Date in bits
    char password[5]; // Password for protected files (empty if not protected)
    bool is_directory; // True if the entry is a directory, false if it's a file
    uint16_t startBlock; // Start block of the file or directory
};
```

I couldn’t implement the exact same structure, but I did a similar structure with MS-DOS directory entry as written in the pdf. I used the reserved part for modification date, password and is_directory variables. Also, I couldn’t make arbitrary file lengths so I limited its size to 25.

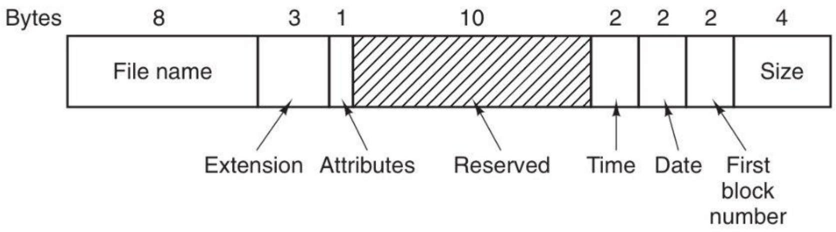
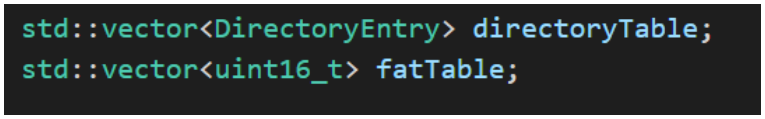


Figure 4-30. The MS-DOS directory entry.

- fileName: The name of the file or directory. It is a fixed-length character array to simplify serialization and deserialization.
- size: The size of the file in bytes.
- permissions: A bitmask representing the read and write permissions of the file or directory.
- creationDate: The creation date and time of the file or directory.
- modificationDate: The last modification date and time of the file or directory.
- password: A fixed-length character array storing the password for protected files.
- is_directory: A boolean flag indicating whether the entry is a directory.
- startBlock: The starting block of the file or directory in the file system's data area.

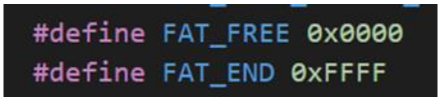
2. Free Blocks Management

Free blocks in the file system are managed using the FAT (File Allocation Table). The FAT is a linear array where each entry corresponds to a block in the file system.



```
std::vector<DirectoryEntry> directoryTable;
std::vector<uint16_t> fatTable;
```

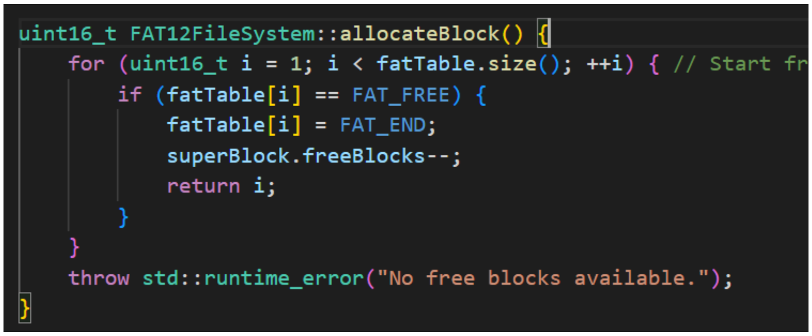
FAT Table:

- Each entry in the FAT can have one of the following values:
    - FAT_FREE (0x0000): Indicates that the block is free and available for allocation.
    - FAT_END (0xFFFF): Indicates the end of a file's block chain.
    - Any other value: Indicates the next block in a file's block chain.
    - 

```
#define FAT_FREE 0x0000
#define FAT_END 0xFFFF
```

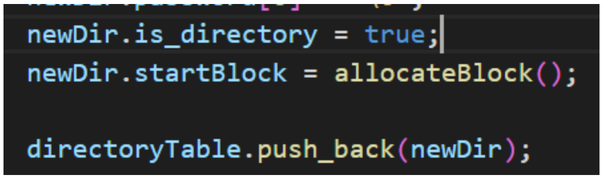
Allocation and Deallocation:

- **Allocation (allocateBlock)**: Searches the FAT for a free block, marks it as the end of a file's block chain, and decreases the count of free blocks in the superblock.



```
uint16_t FAT12FileSystem::allocateBlock() {
    for (uint16_t i = 1; i < fatTable.size(); ++i) { // Start fr
        if (fatTable[i] == FAT_FREE) {
            fatTable[i] = FAT_END;
            superBlock.freeBlocks--;
            return i;
        }
    }
    throw std::runtime_error("No free blocks available.");
}
```

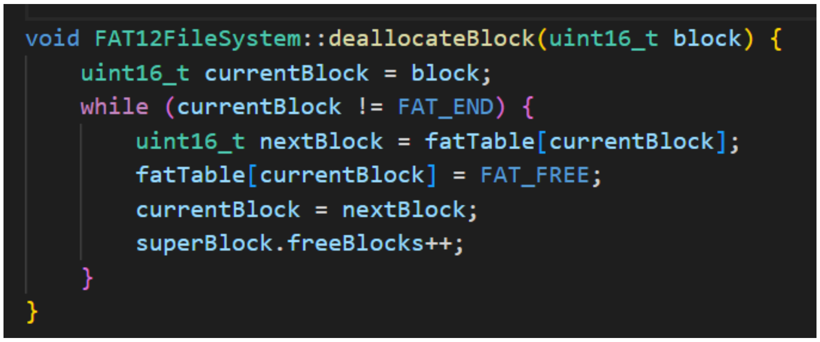
When we create directory or file, it finds its start block according to this FAT table:



```
newDir.is_directory = true;
newDir.startBlock = allocateBlock();

directoryTable.push_back(newDir);
```

- **Deallocation (deallocateBlock)**: Follows the block chain of a file and marks each block as free in the FAT, incrementing the count of free blocks in the superblock.

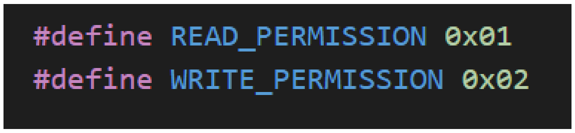


```
void FAT12FileSystem::deallocateBlock(uint16_t block) {
    uint16_t currentBlock = block;
    while (currentBlock != FAT_END) {
        uint16_t nextBlock = fatTable[currentBlock];
        fatTable[currentBlock] = FAT_FREE;
        currentBlock = nextBlock;
        superBlock.freeBlocks++;
    }
}
```

3. Permissions Handling

Permissions are managed using a uint16_t bitmask within the DirectoryEntry struct.

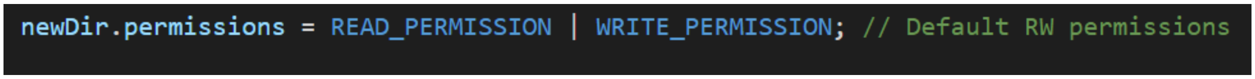
Permission Flags:



```
#define READ_PERMISSION 0x01
#define WRITE_PERMISSION 0x02
```

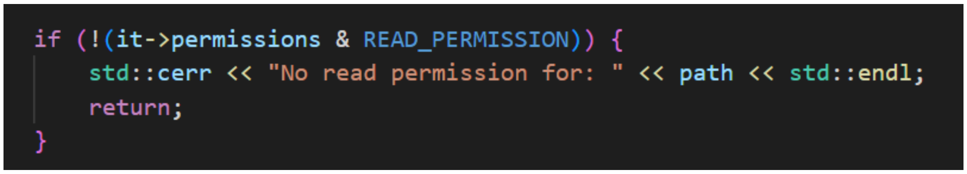
Permission Management:

- **Setting Permissions**: Permissions are set when a file or directory is created. By default, files and directories have both read and write permissions.

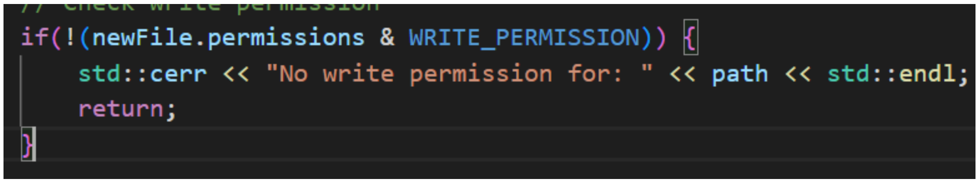


```
newDir.permissions = READ_PERMISSION | WRITE_PERMISSION; // Default RW permissions
```

- **Changing Permissions (changePermissions)**: Modifies the permissions bitmask based on the provided string (+r, -w, +rw, -rw). The changes are saved back to the directory table.
- **Checking Permissions**: Before performing read or write operations, the appropriate permission bits are checked. If the required permissions are not set, an exception is thrown.



```
if (!(it->permissions & READ_PERMISSION)) {
    std::cerr << "No read permission for: " << path << std::endl;
    return;
}
```



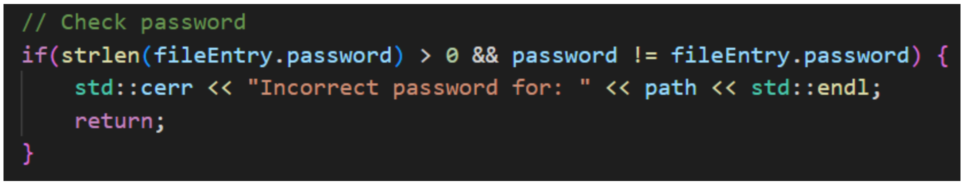
```
// Check write permission
if(!(newFile.permissions & WRITE_PERMISSION)) {
    std::cerr << "No write permission for: " << path << std::endl;
    return;
}
```

4. Password Protection

Password protection is handled using the password field in the DirectoryEntry struct. The password is a fixed-length character array.

Password Management:

- **Setting Password (addPassword)**: Adds a password to a file or directory. The password is stored in the password field of the corresponding DirectoryEntry. The changes are saved back to the directory table.
- **Checking Password**: Before performing any operation on a password-protected file, the provided password is checked against the stored password. If the passwords do not match, it returns with error.



```
// Check password
if(strlen(fileEntry.password) > 0 && password != fileEntry.password) {
    std::cerr << "Incorrect password for: " << path << std::endl;
    return;
}
```

Function names of my source code that handles the file system operations listed in the table of Part 3:

dir: fs.listDirectory(path);

mkdir: fs.makeDirectory(path);

rmdir: fs.removeDirectory(path);

dumpe2fs: fs.dumpFileSystemInfo();

write: fs.writeFile(path, linuxFile);

read: fs.readFile(path, linuxFile, password);

chmod: fs.changePermissions(path, permissions);

addpw: fs.addPassword(path, password);

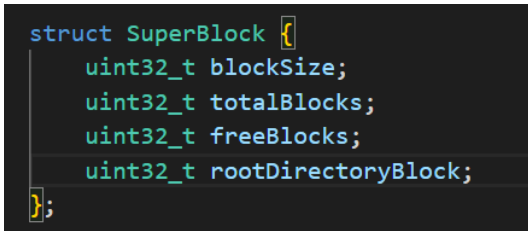
del: fs.deleteFile(path);

PART2:

FAT12 components include the superblock, the File Allocation Table (FAT), the directory table, and the data area.

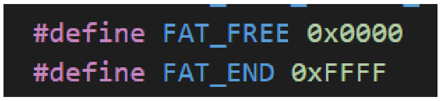
Components:

- **Superblock**: The superblock is a structure located at the beginning of the file system. It contains metadata about the entire file system, including:



```
struct SuperBlock {
    uint32_t blockSize;
    uint32_t totalBlocks;
    uint32_t freeBlocks;
    uint32_t rootDirectoryBlock;
};
```

- **File Allocation Table (FAT)**: The table maps the allocation status of each block in the file system. Each entry in the FAT corresponds to a block and can indicate:



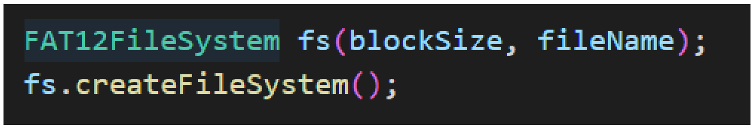
```
#define FAT_FREE 0x0000
#define FAT_END 0xFFFF
```

  - FAT_FREE (0x0000): The block is free and available for allocation.
  - FAT_END (0xFFFF): The block marks the end of a file's block chain.
  - Any other value: The next block in a file's block chain.

- **Directory Table**: The directory table is a collection of directory entries. It stores metadata about all files and directories within the file system. The table is loaded into memory when the file system is initialized and saved back to disk when changes occur.
- **Data Area**: The data area is the part of the file system where the actual content of files and directories is stored. It is divided into blocks, and the FAT is used to manage the allocation of these blocks.

When you execute the command **.\makeFileSystem 1 fat.dat**, the process initializes and creates the FAT12 file system on a file named **fat.dat** with a block size of 1 KB.

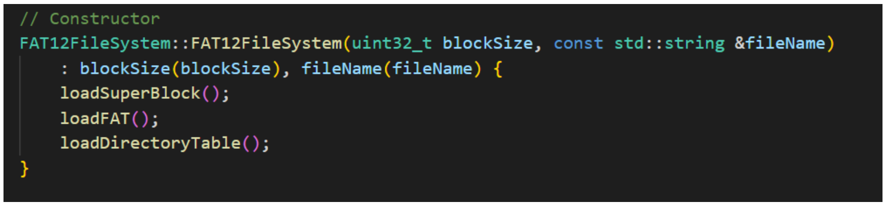
File System Creation:



```
FAT12FileSystem fs(blockSize, fileName);
fs.createFileSystem();
```

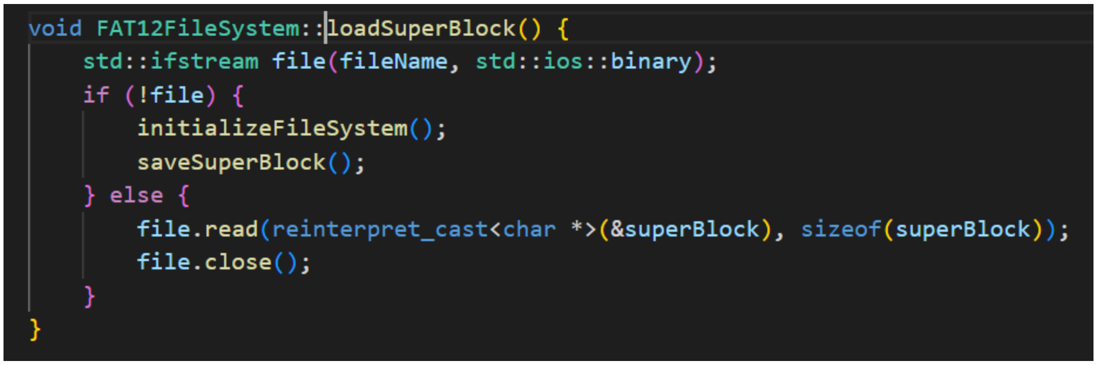
Constructor:

The constructor initializes the block size and file name, then calls methods to load the superblock, FAT, and directory table from the file system.



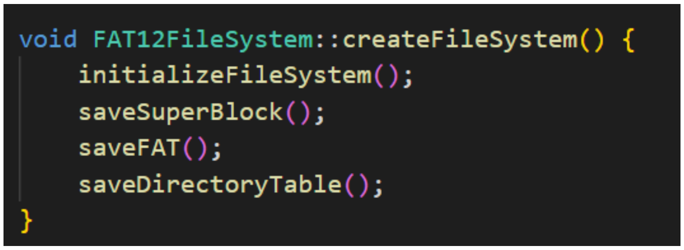
```
// Constructor
FAT12FileSystem::FAT12FileSystem(uint32_t blockSize, const std::string &fileName)
    : blockSize(blockSize), fileName(fileName) {
    loadSuperBlock();
    loadFAT();
    loadDirectoryTable();
}
```

- The **loadSuperBlock**, **loadFAT**, and **loadDirectoryTable** functions are essential for initializing the file system by loading the superblock, FAT, and directory table from the disk into memory. Each function follows a similar process: opening the file in binary read mode, positioning the file pointer, reading the data into memory, and closing the file.



```
void FAT12FileSystem::loadSuperBlock() {
    std::ifstream file(fileName, std::ios::binary);
    if (!file) {
        initializeFileSystem();
        saveSuperBlock();
    } else {
        file.read(reinterpret_cast<char *>(&superBlock), sizeof(superBlock));
        file.close();
    }
}
```

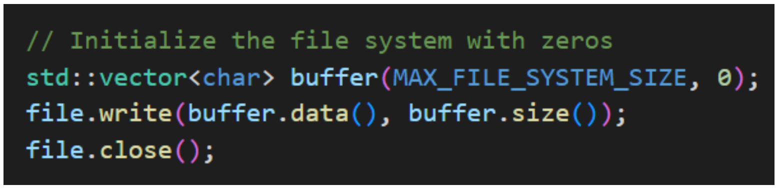
Method: createFileSystem:



```
void FAT12FileSystem::createFileSystem() {
    initializeFileSystem();
    saveSuperBlock();
    saveFAT();
    saveDirectoryTable();
}
```

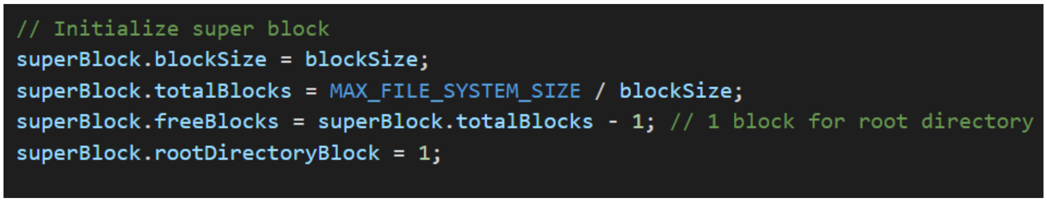
initializeFileSystem:

- The file representing the file system (fat.dat) is created and initialized with zeros. This sets up the file to hold the file system's data.



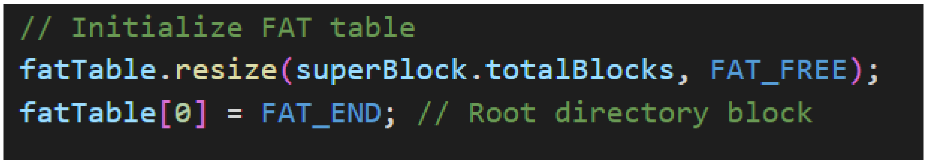
```
// Initialize the file system with zeros
std::vector<char> buffer(MAX_FILE_SYSTEM_SIZE, 0);
file.write(buffer.data(), buffer.size());
file.close();
```

- The superblock is configured with metadata about the file system, including the block size, total number of blocks, number of free blocks, and the starting block for the root directory. MAX_FILE_SYSTEM_SIZE is 4 MB (4 * 1024 * 1024) as written in the pdf.



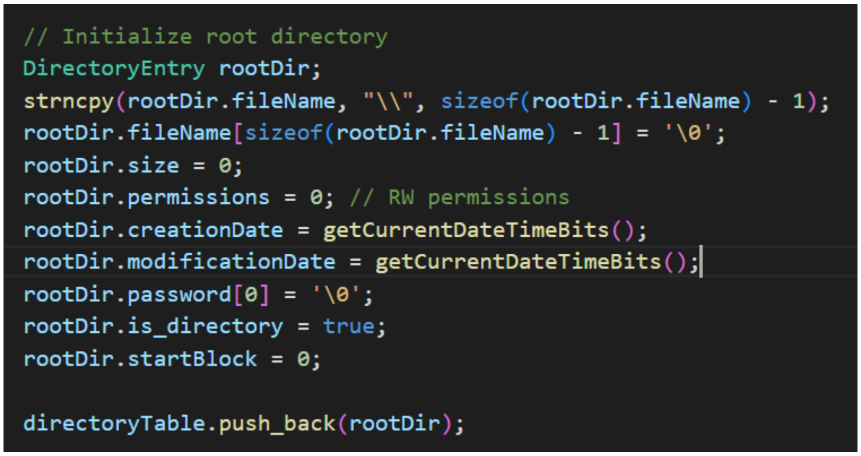
```
// Initialize super block
superBlock.blockSize = blockSize;
superBlock.totalBlocks = MAX_FILE_SYSTEM_SIZE / blockSize;
superBlock.freeBlocks = superBlock.totalBlocks - 1; // 1 block for root directory
superBlock.rootDirectoryBlock = 1;
```

- The FAT is initialized with `FAT_FREE` values, indicating that all blocks are free, except for the first entry, which is set to `FAT_END` to mark the end of the reserved root directory block.



```
// Initialize FAT table
fatTable.resize(superBlock.totalBlocks, FAT_FREE);
fatTable[0] = FAT_END; // Root directory block
```

- The root directory initialized in the file system. The **filename** is set to "\\". This represents the root directory's file name. The **startBlock** is set to 0, indicating that the root directory starts at the first block of the file system. Finally, the **rootDir** object is added to a **directoryTable** vector with push_back.



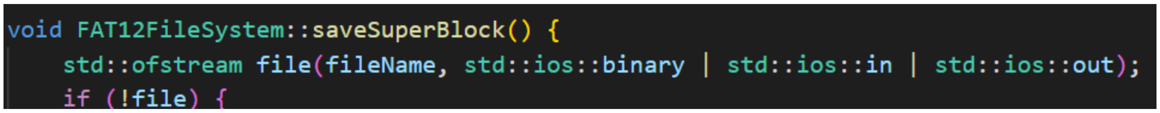
```
// Initialize root directory
DirectoryEntry rootDir;
strncpy(rootDir.fileName, "\\", sizeof(rootDir.fileName) - 1);
rootDir.fileName[sizeof(rootDir.fileName) - 1] = '\0';
rootDir.size = 0;
rootDir.permissions = 0; // RW permissions
rootDir.creationDate = getCurrentDateTimeBits();
rootDir.modificationDate = getCurrentDateTimeBits();
rootDir.password[0] = '\0';
rootDir.is_directory = true;
rootDir.startBlock = 0;

directoryTable.push_back(rootDir);
```

saveSuperBlock:

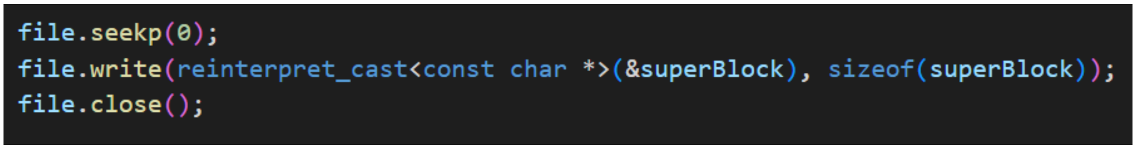
It writes the superblock data from memory to the disk file that represents the file system. This ensures that the file system's metadata is saved and can be retrieved later.

- It opens the file representing the file system **(fat.dat)** in binary mode with both input and output capabilities. This allows the function to modify the file without truncating its contents.



```
void FAT12FileSystem::saveSuperBlock() {
    std::ofstream file(fileName, std::ios::binary | std::ios::in | std::ios::out);
    if (!file) {
```

- The file pointer is moved to the beginning of the file using **file.seekp(0).** The superblock is always located at the start of the file as I show in **hexdump** output.
- The current state of the superblock is written on the file. The **reinterpret_cast** is used to convert the superblock data to a format suitable for binary writing.

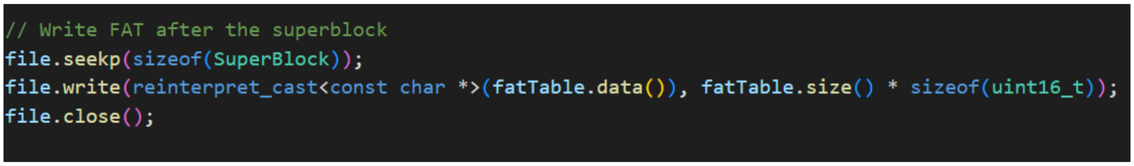


```
file.seekp(0);
file.write(reinterpret_cast<const char *>(&superBlock), sizeof(superBlock));
file.close();
```

saveFat:

It writes the current state of the FAT from memory to the disk. The FAT keeps track of which blocks are allocated and which are free, and it links the blocks of each file.

- It again opens the file system in binary mode.
- The file pointer is moved to the position immediately after the superblock with **seekp**. In a FAT12 file system, the superblock is located at the very beginning of the file, so this moves the pointer to the correct position to start writing the FAT.
- The current state of the FAT, stored in the **fatTable** vector in memory, is written on the file.

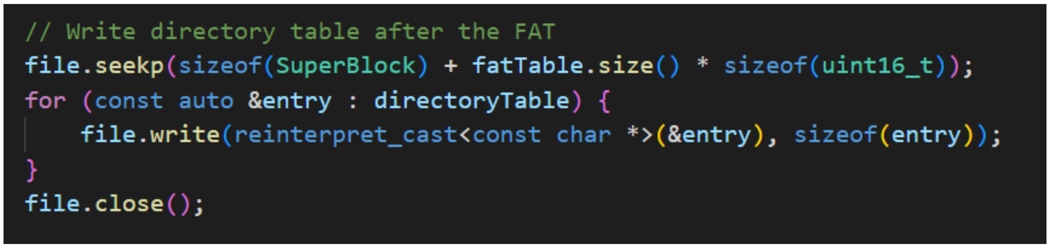


```
// Write FAT after the superblock
file.seekp(sizeof(SuperBlock));
file.write(reinterpret_cast<const char *>(fatTable.data()), fatTable.size() * sizeof(uint16_t));
file.close();
```

saveDirectoryTable:

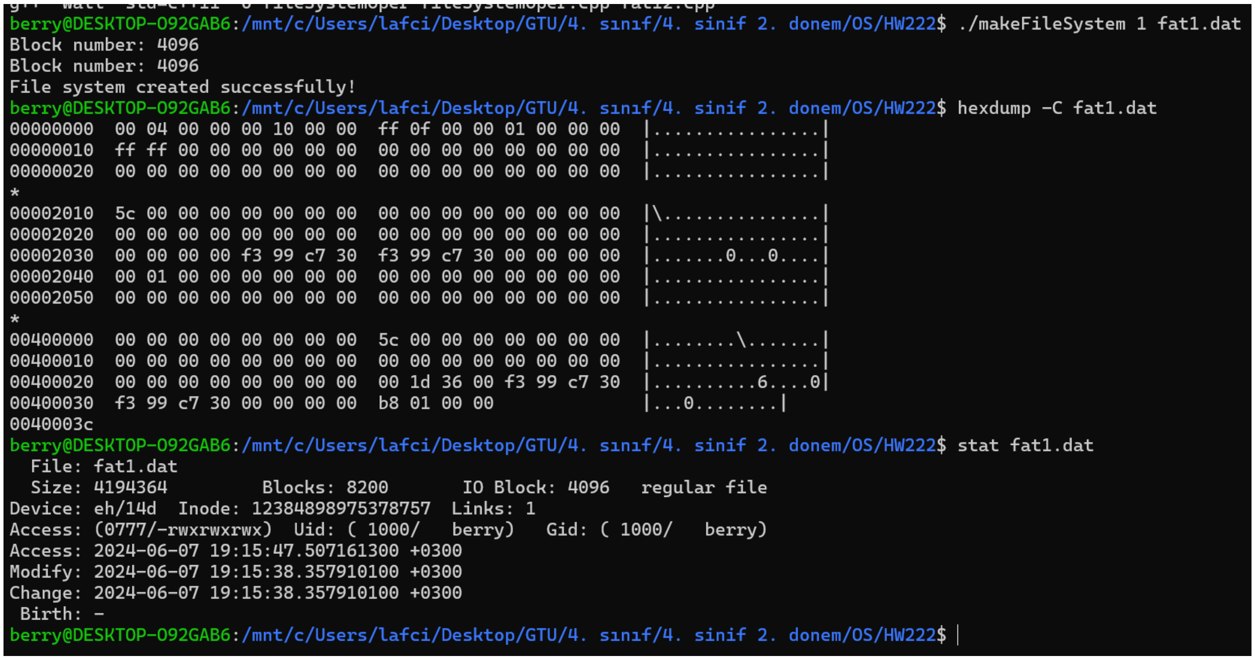
It writes the current state of the directory table from memory to the disk. The directory table holds metadata about all files and directories in the file system.

- It again opens the file system in binary mode.
- The file pointer is moved to the position immediately after the FAT. This is calculated as **sizeof(SuperBlock) + fatTable.size() * sizeof(uint16_t).**
- The function iterates over the directoryTable vector. Each DirectoryEntry is written to the file system.

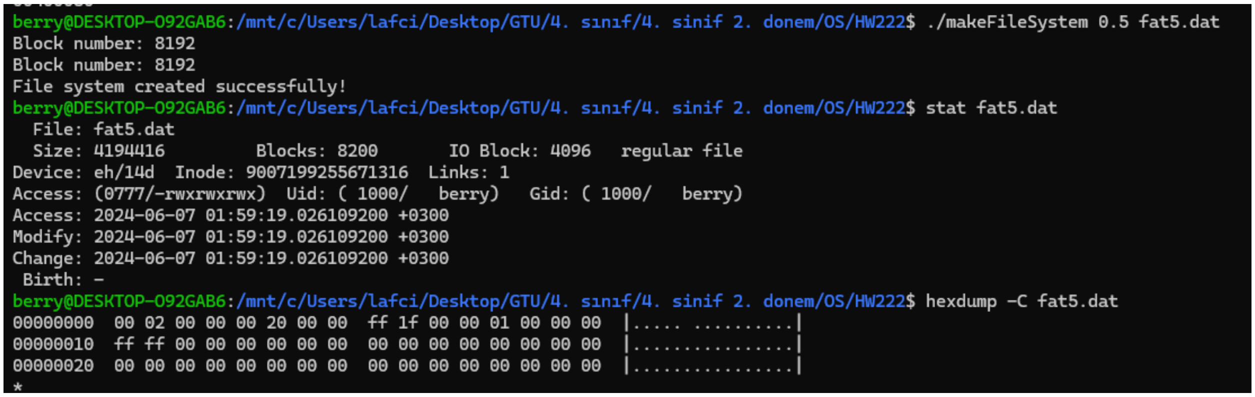


```
// Write directory table after the FAT
file.seekp(sizeof(SuperBlock) + fatTable.size() * sizeof(uint16_t));
for (const auto &entry : directoryTable) {
    file.write(reinterpret_cast<const char *>(&entry), sizeof(entry));
}
file.close();
```

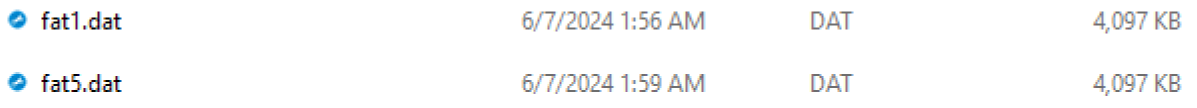
Test cases:



```
g++ -Wall -std=c++11 -o fileSystemOper fileSystemOper.cpp fat12.cpp
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./makeFileSystem 1 fat1.dat
Block number: 4096
Block number: 4096
File system created successfully!
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ hexdump -C fat1.dat
00000000  00 04 00 00 00 10 00 00  ff 0f 00 00 01 00 00 00  |................|
00000010  ff ff 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00002010  5c 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |\...............|
00002020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00002030  00 00 00 00 f3 99 c7 30  f3 99 c7 30 00 00 00 00  |.......0...0....|
00002040  00 01 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00002050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00400000  00 00 00 00 00 00 00 00  5c 00 00 00 00 00 00 00  |........\.......|
00400010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00400020  00 00 00 00 00 00 00 00  00 1d 36 00 f3 99 c7 30  |..........6....0|
00400030  f3 99 c7 30 00 00 00 00  b8 01 00 00              |...0........|
0040003c
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ stat fat1.dat
  File: fat1.dat
  Size: 4194364         Blocks: 8200       IO Block: 4096   regular file
Device: eh/14d  Inode: 12384898975378757  Links: 1
Access: (0777/-rwxrwxrwx)  Uid: ( 1000/   berry)   Gid: ( 1000/   berry)
Access: 2024-06-07 19:15:47.507161300 +0300
Modify: 2024-06-07 19:15:38.357910100 +0300
Change: 2024-06-07 19:15:38.357910100 +0300
 Birth: -
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$
```



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./makeFileSystem 0.5 fat5.dat
Block number: 8192
Block number: 8192
File system created successfully!
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ stat fat5.dat
  File: fat5.dat
  Size: 4194416         Blocks: 8200       IO Block: 4096   regular file
Device: eh/14d  Inode: 9007199255671316  Links: 1
Access: (0777/-rwxrwxrwx)  Uid: ( 1000/   berry)   Gid: ( 1000/   berry)
Access: 2024-06-07 01:59:19.026109200 +0300
Modify: 2024-06-07 01:59:19.026109200 +0300
Change: 2024-06-07 01:59:19.026109200 +0300
 Birth: -
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ hexdump -C fat5.dat
00000000  00 02 00 00 00 20 00 00  ff 1f 00 00 01 00 00 00  |..... ..........|
00000010  ff ff 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
```



fat1.dat	6/7/2024 1:56 AM	DAT	4,097 KB
fat5.dat	6/7/2024 1:59 AM	DAT	4,097 KB

Explanation for ./makeFileSystem 1 fat1.dat:

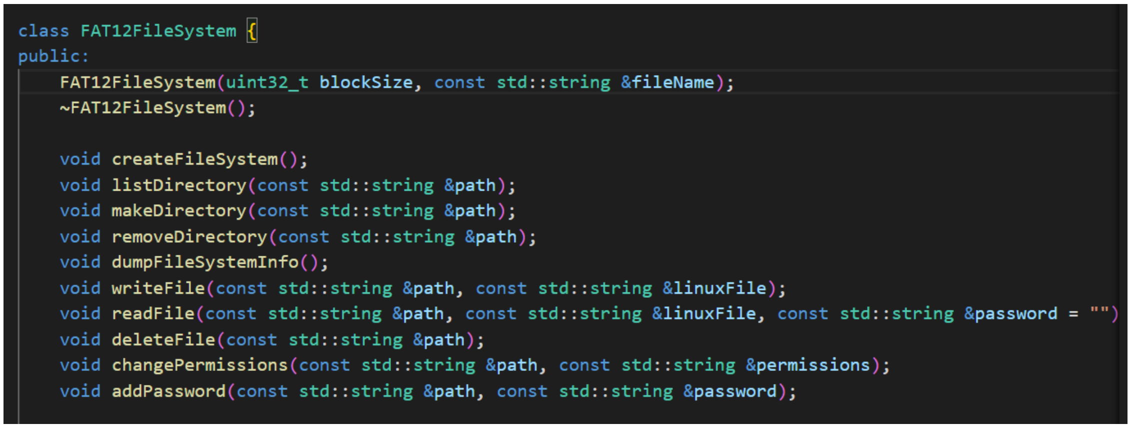
1. **Superblock (First 16 bytes)**
    - 00 04 00 00: Block size (1024 bytes or 1 KB, 0x0400)

- 00 10 00 00: Total blocks (16 blocks, 0x10)
- ff 0f 00 00: Free blocks (4095 blocks, 0x0fff)
- 01 00 00 00: Root directory block (block 1, 0x01)
- These values indicate the configuration of the file system, including the block size, total blocks, number of free blocks, and the root directory block.

2. **FAT Table (Next 16 bytes)**
    - ff ff: Indicates that the first block is used and marked as FAT_END.
    - 00 00: Indicates that the second block is free.
    - Remaining bytes are also set to 00 00, indicating that these blocks are free.
3. **Directory Table (Starting at offset 0x2010)**
    - 5c 00 00 00: Represents the ASCII value for \, which indicates the root directory.
    - Remaining bytes until offset 0x2040 are zeroed out or contain metadata:
        - 00 00 00 00: Directory size (0 for directories)
        - f3 99 c7 30 f3 99 c7 30: Timestamps for creation and modification dates
        - 00 00 00 00: Permissions (default, no specific permissions set)
        - 00 01 00 00: Start block (block 1 for the root directory)
        - Remaining bytes are zeroed out.

PART 3:

Here are my operations:



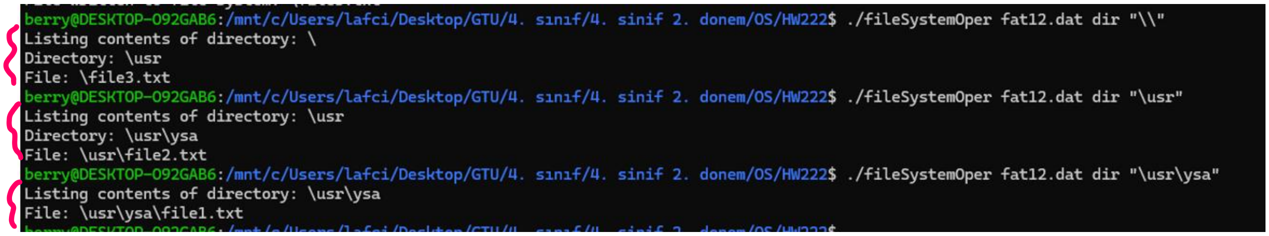
```
class FAT12FileSystem {
public:
    FAT12FileSystem(uint32_t blockSize, const std::string &fileName);
    ~FAT12FileSystem();

    void createFileSystem();
    void listDirectory(const std::string &path);
    void makeDirectory(const std::string &path);
    void removeDirectory(const std::string &path);
    void dumpFileSystemInfo();
    void writeFile(const std::string &path, const std::string &linuxFile);
    void readFile(const std::string &path, const std::string &linuxFile, const std::string &password = "")
    void deleteFile(const std::string &path);
    void changePermissions(const std::string &path, const std::string &permissions);
    void addPassword(const std::string &path, const std::string &password);
```

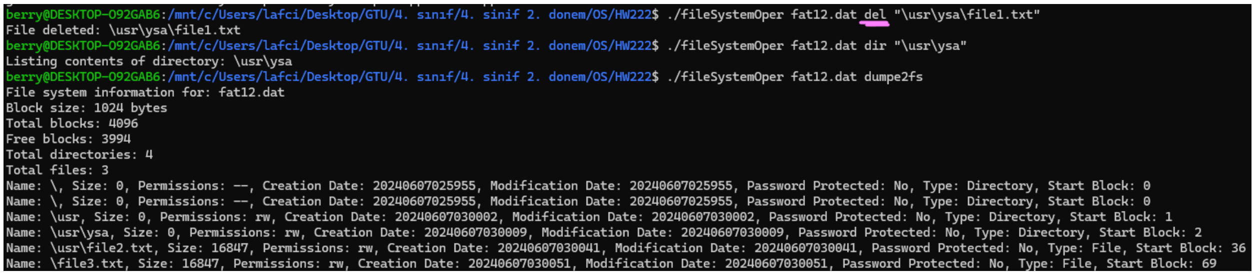
Test Cases:

dir: fs.listDirectory(path);

It lists the created directories and files under the given path.



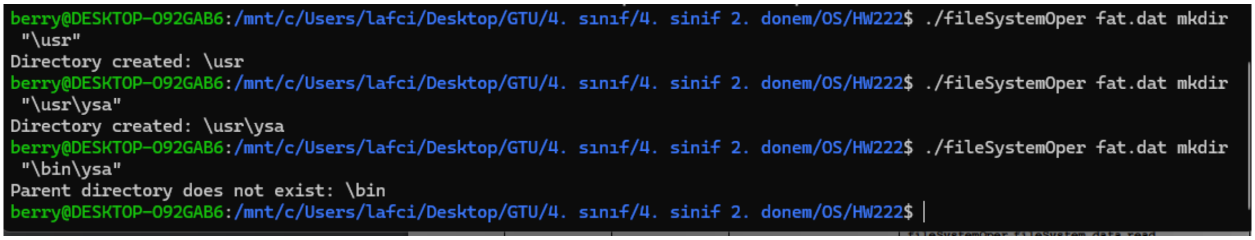
```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dir "\\"
Listing contents of directory: \
Directory: \usr
File: \file3.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dir "\usr"
Listing contents of directory: \usr
Directory: \usr\ysa
File: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dir "\usr\ysa"
Listing contents of directory: \usr\ysa
File: \usr\ysa\file1.txt
```



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat del "\usr\ysa\file1.txt"
File deleted: \usr\ysa\file1.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dir "\usr\ysa"
Listing contents of directory: \usr\ysa
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dumpe2fs
File system information for: fat12.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 3994
Total directories: 4
Total files: 3
Name: \, Size: 0, Permissions: --, Creation Date: 20240607025955, Modification Date: 20240607025955, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607025955, Modification Date: 20240607025955, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: rw, Creation Date: 20240607030002, Modification Date: 20240607030002, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\ysa, Size: 0, Permissions: rw, Creation Date: 20240607030009, Modification Date: 20240607030009, Password Protected: No, Type: Directory, Start Block: 2
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607030041, Modification Date: 20240607030041, Password Protected: No, Type: File, Start Block: 36
Name: \file3.txt, Size: 16847, Permissions: rw, Creation Date: 20240607030051, Modification Date: 20240607030051, Password Protected: No, Type: File, Start Block: 69
```

mkdir: fs.makeDirectory(path);

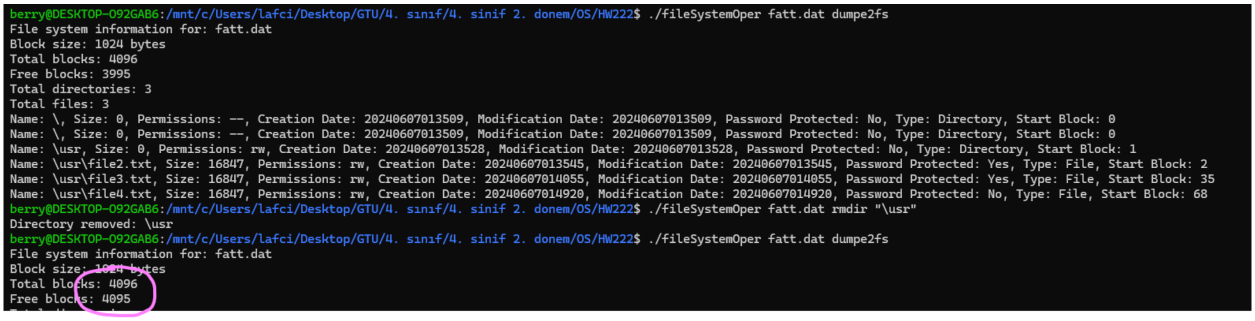
It gives error to \bin\ysa



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat mkdir
 "\usr"
Directory created: \usr
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat mkdir
 "\usr\ysa"
Directory created: \usr\ysa
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat mkdir
 "\bin\ysa"
Parent directory does not exist: \bin
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$
```

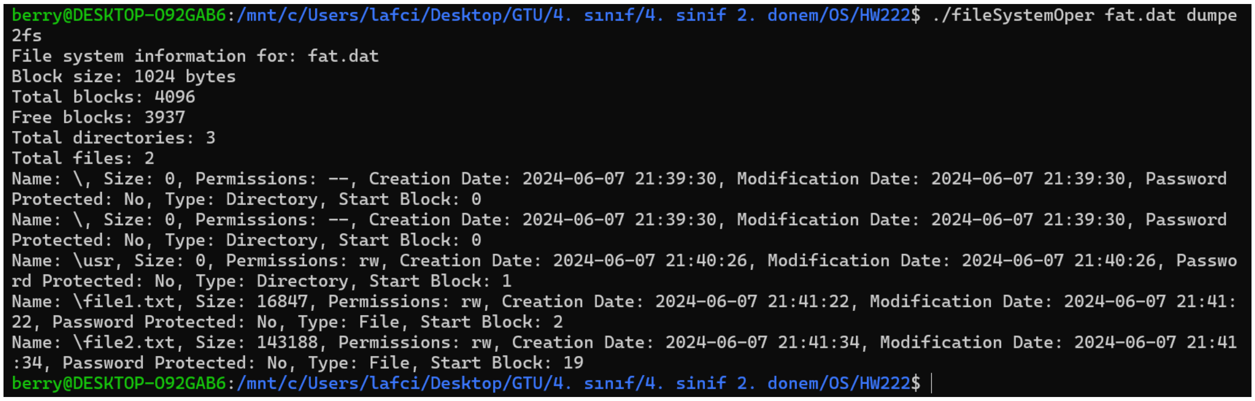
rmdir: fs.removeDirectory(path);

It deleted all the directory with its contents.



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat dumpe2fs
File system information for: fatt.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 3995
Total directories: 3
Total files: 3
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: rw, Creation Date: 20240607013528, Modification Date: 20240607013528, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607013545, Modification Date: 20240607013545, Password Protected: Yes, Type: File, Start Block: 2
Name: \usr\file3.txt, Size: 16847, Permissions: rw, Creation Date: 20240607014055, Modification Date: 20240607014055, Password Protected: Yes, Type: File, Start Block: 35
Name: \usr\file4.txt, Size: 16847, Permissions: rw, Creation Date: 20240607014920, Modification Date: 20240607014920, Password Protected: No, Type: File, Start Block: 68
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat rmdir "\usr"
Directory removed: \usr
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat dumpe2fs
File system information for: fatt.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4095
```

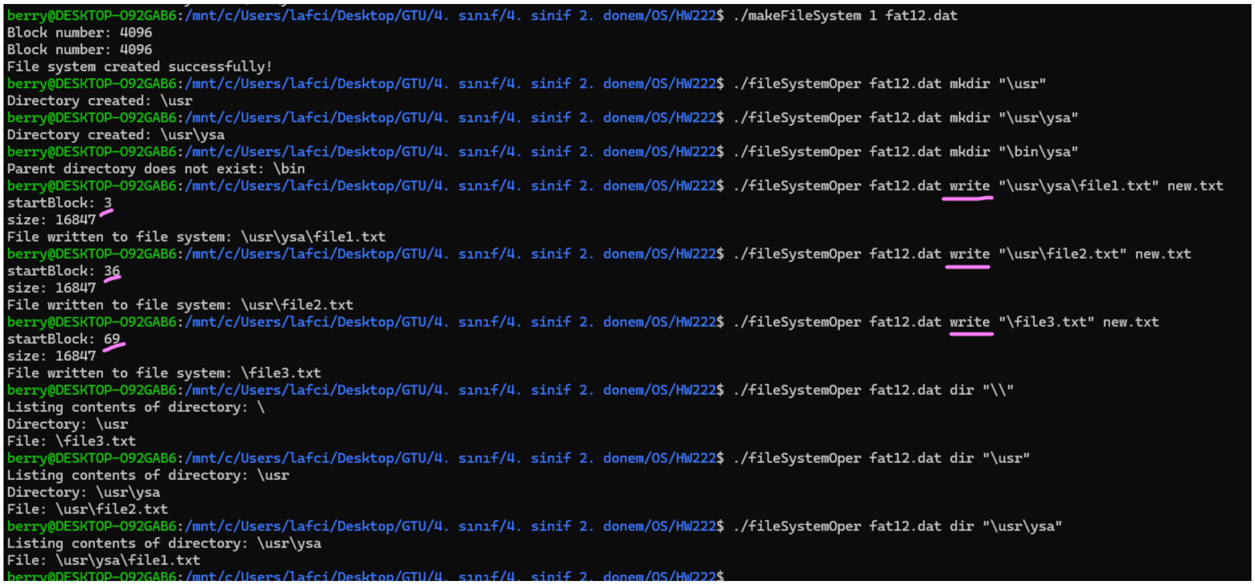
dumpe2fs: fs.dumpFileSystemInfo();



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat dumpe
2fs
File system information for: fat.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 3937
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 2024-06-07 21:39:30, Modification Date: 2024-06-07 21:39:30, Password
Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 2024-06-07 21:39:30, Modification Date: 2024-06-07 21:39:30, Password
Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: rw, Creation Date: 2024-06-07 21:40:26, Modification Date: 2024-06-07 21:40:26, Passwo
rd Protected: No, Type: Directory, Start Block: 1
Name: \file1.txt, Size: 16847, Permissions: rw, Creation Date: 2024-06-07 21:41:22, Modification Date: 2024-06-07 21:41:
22, Password Protected: No, Type: File, Start Block: 2
Name: \file2.txt, Size: 143188, Permissions: rw, Creation Date: 2024-06-07 21:41:34, Modification Date: 2024-06-07 21:41
:34, Password Protected: No, Type: File, Start Block: 19
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$
```

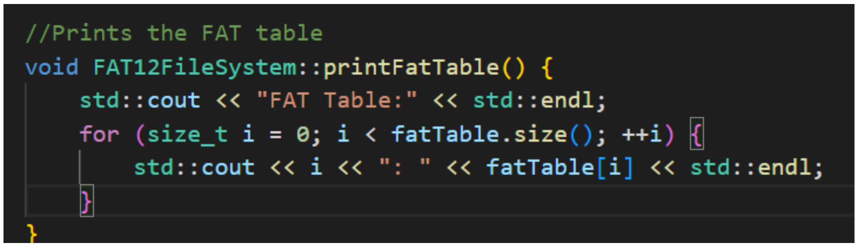
write: fs.writeFile(path, linuxFile);

You can see the corresponding files under the paths.



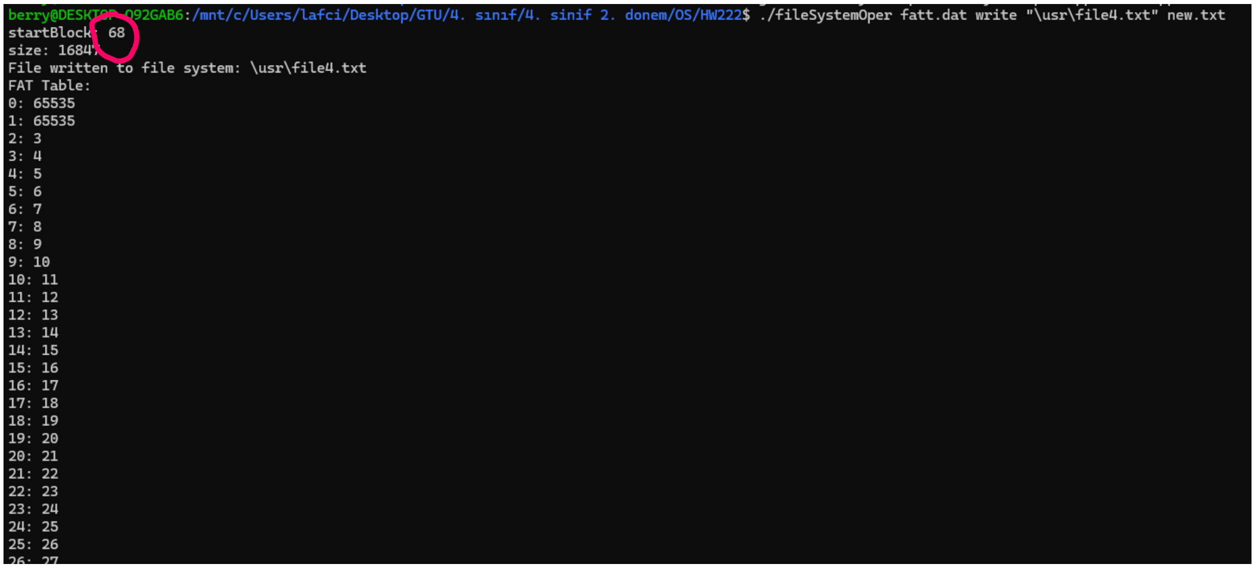
```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./makeFileSystem 1 fat12.dat
Block number: 4096
Block number: 4096
File system created successfully!
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat mkdir "\usr"
Directory created: \usr
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat mkdir "\usr\ysa"
Directory created: \usr\ysa
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat mkdir "\bin\ysa"
Parent directory does not exist: \bin
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat write "\usr\ysa\file1.txt" new.txt
startBlock: 3
size: 16847
File written to file system: \usr\ysa\file1.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat write "\usr\file2.txt" new.txt
startBlock: 36
size: 16847
File written to file system: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat write "\file3.txt" new.txt
startBlock: 69
size: 16847
File written to file system: \file3.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dir "\\"
Listing contents of directory: \
Directory: \usr
File: \file3.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dir "\usr"
Listing contents of directory: \usr
Directory: \usr\ysa
File: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat12.dat dir "\usr\ysa"
Listing contents of directory: \usr\ysa
File: \usr\ysa\file1.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$
```

I did an extra function to see the FAT table and linkings.



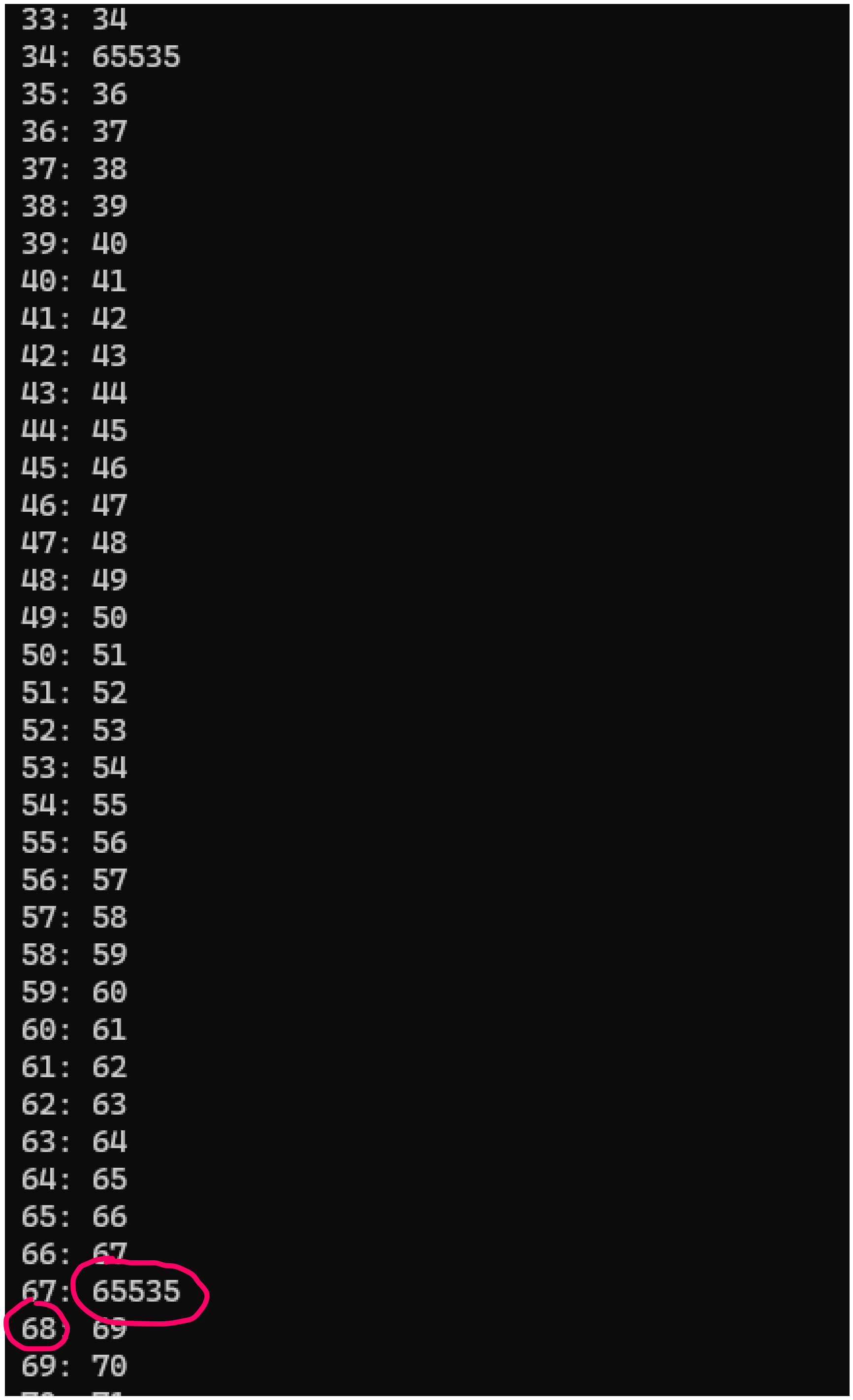
```cpp
//Prints the FAT table
void FAT12FileSystem::printFatTable() {
    std::cout << "FAT Table:" << std::endl;
    for (size_t i = 0; i < fatTable.size(); ++i) {
        std::cout << i << ": " << fatTable[i] << std::endl;
    }
}
```

The first two are allocated for super block and the root.

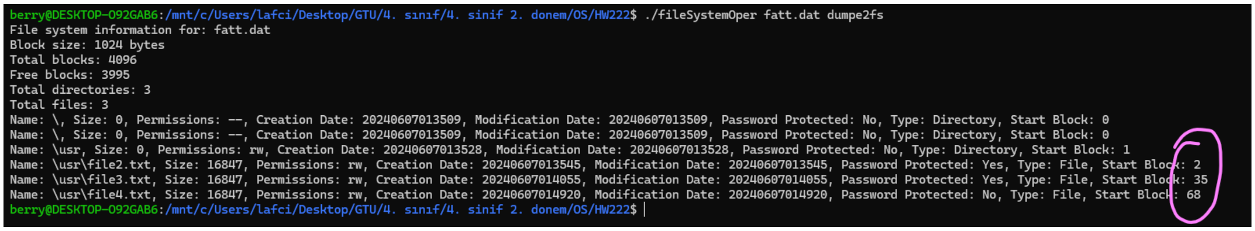


```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat write "\usr\file4.txt" new.txt
startBlock: 68
size: 16847
File written to file system: \usr\file4.txt
FAT Table:
0: 65535
1: 65535
2: 3
3: 4
4: 5
5: 6
6: 7
7: 8
8: 9
9: 10
10: 11
11: 12
12: 13
13: 14
14: 15
15: 16
16: 17
17: 18
18: 19
19: 20
20: 21
21: 22
22: 23
23: 24
24: 25
25: 26
26: 27
```

As you can see, file4.txt starts from 68th block. 65535 corresponds to 0xFFFF in decimal which is end of the file.

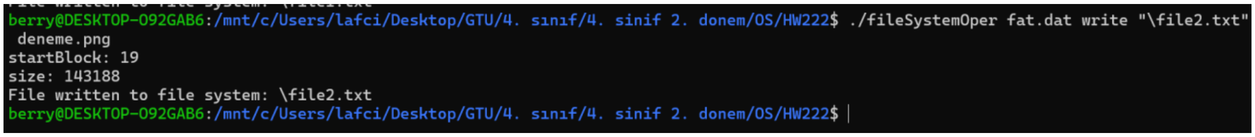


```
33: 34
34: 65535
35: 36
36: 37
37: 38
38: 39
39: 40
40: 41
41: 42
42: 43
43: 44
44: 45
45: 46
46: 47
47: 48
48: 49
49: 50
50: 51
51: 52
52: 53
53: 54
54: 55
55: 56
56: 57
57: 58
58: 59
59: 60
60: 61
61: 62
62: 63
63: 64
64: 65
65: 66
66: 67
67: 65535
68: 69
69: 70
70: 71
```



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat dumpe2fs
File system information for: fatt.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 3995
Total directories: 3
Total files: 3
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: rw, Creation Date: 20240607013528, Modification Date: 20240607013528, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607013545, Modification Date: 20240607013545, Password Protected: Yes, Type: File, Start Block: 2
Name: \usr\file3.txt, Size: 16847, Permissions: rw, Creation Date: 20240607014055, Modification Date: 20240607014055, Password Protected: Yes, Type: File, Start Block: 35
Name: \usr\file4.txt, Size: 16847, Permissions: rw, Creation Date: 20240607014920, Modification Date: 20240607014920, Password Protected: No, Type: File, Start Block: 68
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```

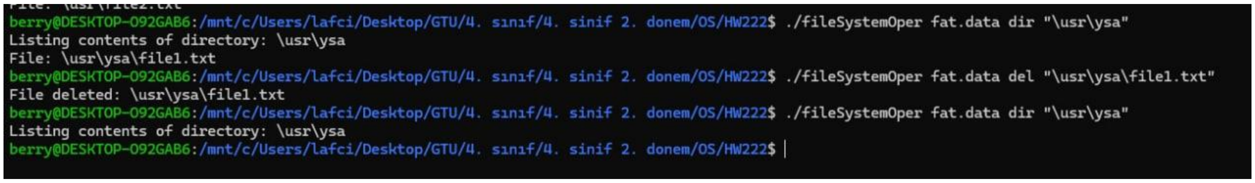
Here I also write a png file:



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat write "\file2.txt"
 deneme.png
startBlock: 19
size: 143188
File written to file system: \file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```

del: fs.deleteFile(path);

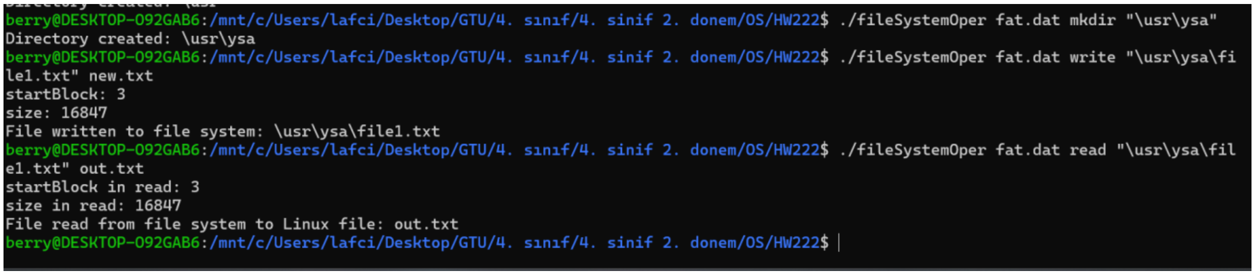
As you can see the file1.txt deleted under the \usr\ysa path.



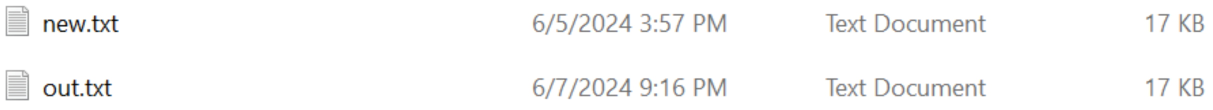
```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.data dir "\usr\ysa"
Listing contents of directory: \usr\ysa
File: \usr\ysa\file1.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.data del "\usr\ysa\file1.txt"
File deleted: \usr\ysa\file1.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.data dir "\usr\ysa"
Listing contents of directory: \usr\ysa
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```

read: fs.readFile(path, linuxFile, password);

You can see the starting block and the sizes are same as written file and read file.

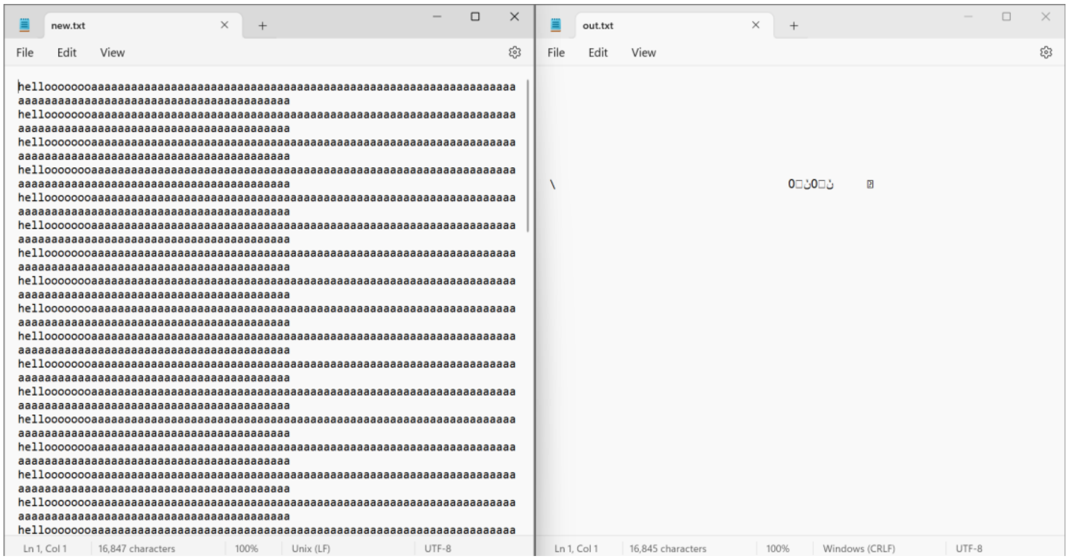


```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat mkdir "\usr\ysa"
Directory created: \usr\ysa
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat write "\usr\ysa\fi
le1.txt" new.txt
startBlock: 3
size: 16847
File written to file system: \usr\ysa\file1.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat read "\usr\ysa\fil
e1.txt" out.txt
startBlock in read: 3
size in read: 16847
File read from file system to Linux file: out.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```



new.txt	6/5/2024 3:57 PM	Text Document	17 KB
out.txt	6/7/2024 9:16 PM	Text Document	17 KB

But I couldn’t read the contents properly. I examined and debugged the code for a long time, but I could not find where the problem was.



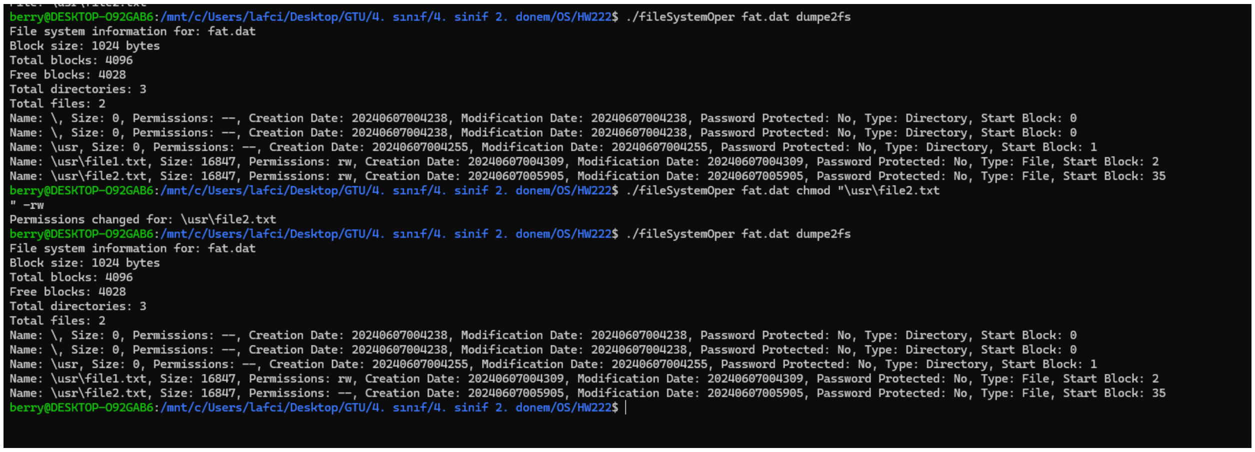
```
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
helloooooooaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Ln 1, Col 1 | 16,847 characters | 100% | Unix (LF) | UTF-8

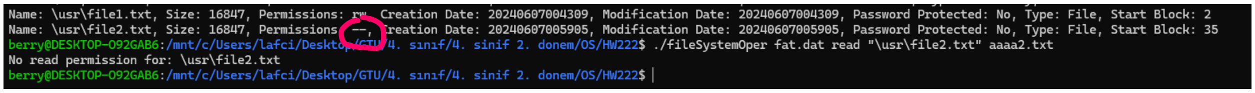
```
\                              0□ن0□ن       ⍰
```

Ln 1, Col 1 | 16,845 characters | 100% | Windows (CRLF) | UTF-8

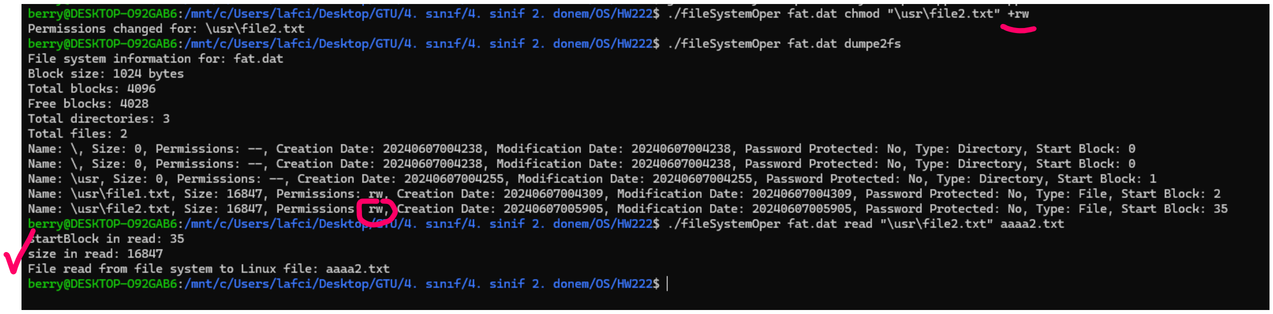
chmod: fs.changePermissions(path, permissions);



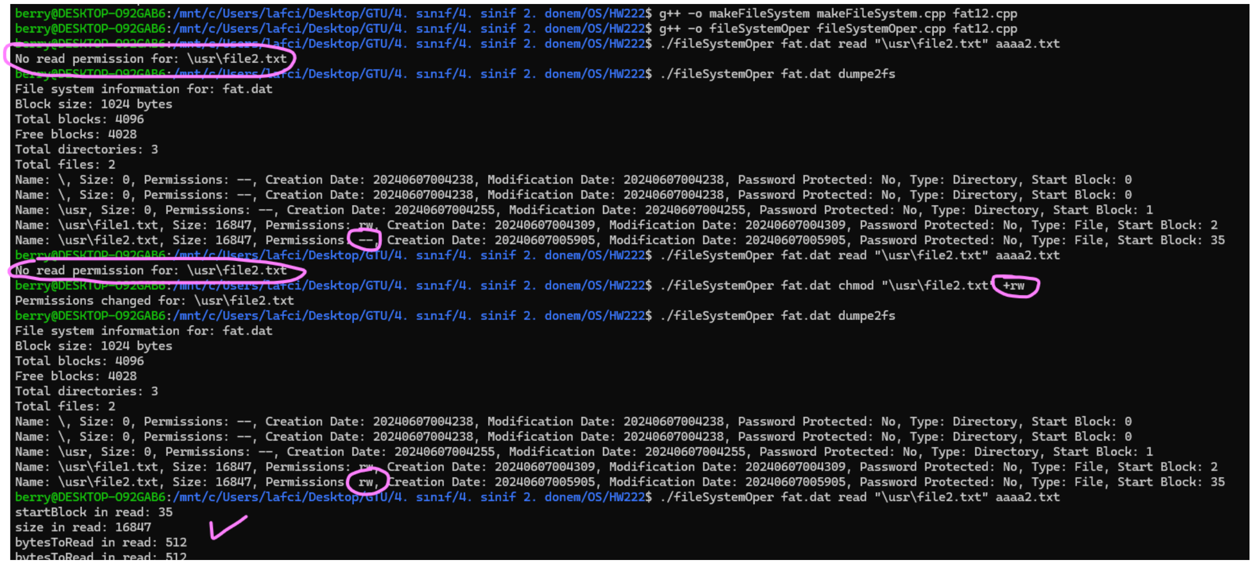
```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat dumpe2fs
File system information for: fat.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4028
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: --, Creation Date: 20240607004255, Modification Date: 20240607004255, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file1.txt, Size: 16847, Permissions: rw, Creation Date: 20240607004309, Modification Date: 20240607004309, Password Protected: No, Type: File, Start Block: 2
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607005905, Modification Date: 20240607005905, Password Protected: No, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat chmod "\usr\file2.txt
" -rw
Permissions changed for: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat dumpe2fs
File system information for: fat.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4028
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: --, Creation Date: 20240607004255, Modification Date: 20240607004255, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file1.txt, Size: 16847, Permissions: rw, Creation Date: 20240607004309, Modification Date: 20240607004309, Password Protected: No, Type: File, Start Block: 2
Name: \usr\file2.txt, Size: 16847, Permissions: --, Creation Date: 20240607005905, Modification Date: 20240607005905, Password Protected: No, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```



```
Name: \usr\file1.txt, Size: 16847, Permissions: rw, Creation Date: 20240607004309, Modification Date: 20240607004309, Password Protected: No, Type: File, Start Block: 2
Name: \usr\file2.txt, Size: 16847, Permissions: --, Creation Date: 20240607005905, Modification Date: 20240607005905, Password Protected: No, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat read "\usr\file2.txt" aaaa2.txt
No read permission for: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```

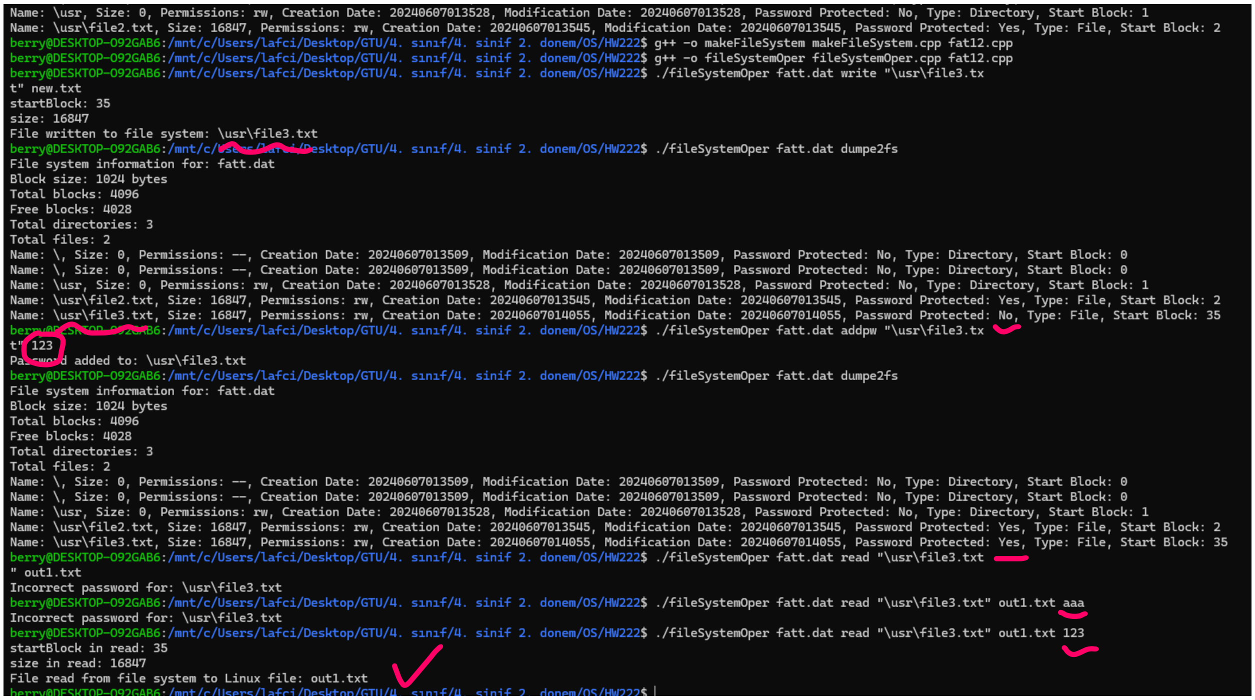


```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat chmod "\usr\file2.txt" +rw
Permissions changed for: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat dumpe2fs
File system information for: fat.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4028
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: --, Creation Date: 20240607004255, Modification Date: 20240607004255, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file1.txt, Size: 16847, Permissions: rw, Creation Date: 20240607004309, Modification Date: 20240607004309, Password Protected: No, Type: File, Start Block: 2
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607005905, Modification Date: 20240607005905, Password Protected: No, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat read "\usr\file2.txt" aaaa2.txt
startBlock in read: 35
size in read: 16847
File read from file system to Linux file: aaaa2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ |
```



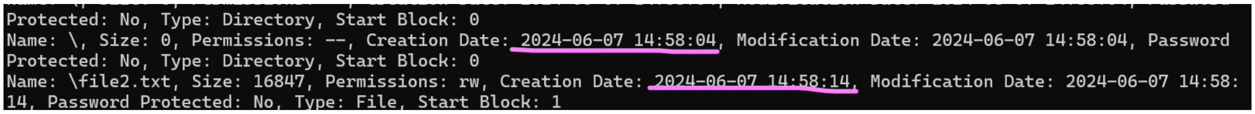
```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ g++ -o makeFileSystem makeFileSystem.cpp fat12.cpp
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ g++ -o fileSystemOper fileSystemOper.cpp fat12.cpp
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat read "\usr\file2.txt" aaaa2.txt
No read permission for: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat dumpe2fs
File system information for: fat.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4028
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: --, Creation Date: 20240607004255, Modification Date: 20240607004255, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file1.txt, Size: 16847, Permissions: rw, Creation Date: 20240607004309, Modification Date: 20240607004309, Password Protected: No, Type: File, Start Block: 2
Name: \usr\file2.txt, Size: 16847, Permissions: --, Creation Date: 20240607005905, Modification Date: 20240607005905, Password Protected: No, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat read "\usr\file2.txt" aaaa2.txt
No read permission for: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat chmod "\usr\file2.txt" +rw
Permissions changed for: \usr\file2.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat dumpe2fs
File system information for: fat.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4028
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607004238, Modification Date: 20240607004238, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: --, Creation Date: 20240607004255, Modification Date: 20240607004255, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file1.txt, Size: 16847, Permissions: rw, Creation Date: 20240607004309, Modification Date: 20240607004309, Password Protected: No, Type: File, Start Block: 2
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607005905, Modification Date: 20240607005905, Password Protected: No, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat read "\usr\file2.txt" aaaa2.txt
startBlock in read: 35
size in read: 16847
bytesToRead in read: 512
bytesToRead in read: 512
```

addpw: fs.addPassword(path, password);

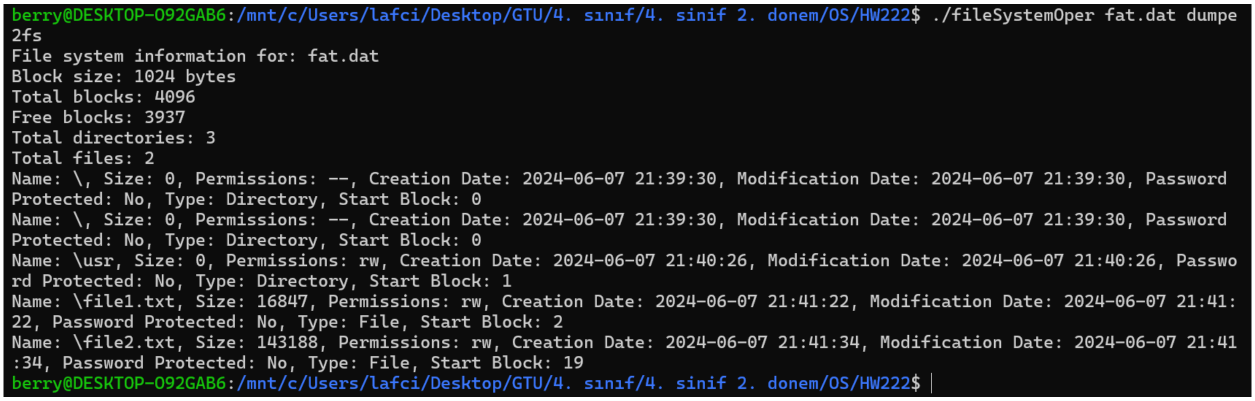


```
Name: \usr, Size: 0, Permissions: rw, Creation Date: 20240607013528, Modification Date: 20240607013528, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607013545, Modification Date: 20240607013545, Password Protected: Yes, Type: File, Start Block: 2
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ g++ -o makeFileSystem makeFileSystem.cpp fat12.cpp
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ g++ -o fileSystemOper fileSystemOper.cpp fat12.cpp
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat write "\usr\file3.tx
t" new.txt
startBlock: 35
size: 16847
File written to file system: \usr\file3.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat dumpe2fs
File system information for: fatt.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4028
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: rw, Creation Date: 20240607013528, Modification Date: 20240607013528, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607013545, Modification Date: 20240607013545, Password Protected: Yes, Type: File, Start Block: 2
Name: \usr\file3.txt, Size: 16847, Permissions: rw, Creation Date: 20240607014055, Modification Date: 20240607014055, Password Protected: No, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat addpw "\usr\file3.tx
t" 123
Password added to: \usr\file3.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat dumpe2fs
File system information for: fatt.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 4028
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 20240607013509, Modification Date: 20240607013509, Password Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: rw, Creation Date: 20240607013528, Modification Date: 20240607013528, Password Protected: No, Type: Directory, Start Block: 1
Name: \usr\file2.txt, Size: 16847, Permissions: rw, Creation Date: 20240607013545, Modification Date: 20240607013545, Password Protected: Yes, Type: File, Start Block: 2
Name: \usr\file3.txt, Size: 16847, Permissions: rw, Creation Date: 20240607014055, Modification Date: 20240607014055, Password Protected: Yes, Type: File, Start Block: 35
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat read "\usr\file3.txt
" out1.txt
Incorrect password for: \usr\file3.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat read "\usr\file3.txt" out1.txt aaa
Incorrect password for: \usr\file3.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fatt.dat read "\usr\file3.txt" out1.txt 123
startBlock in read: 35
size in read: 16847
File read from file system to Linux file: out1.txt
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$
```

As an addition, I did my test cases with the older version of the date and time. Then I made their bits smaller for the directory entry structure. Here is their updated appreance:



```
Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 2024-06-07 14:58:04, Modification Date: 2024-06-07 14:58:04, Password
Protected: No, Type: Directory, Start Block: 0
Name: \file2.txt, Size: 16847, Permissions: rw, Creation Date: 2024-06-07 14:58:14, Modification Date: 2024-06-07 14:58:
14, Password Protected: No, Type: File, Start Block: 1
```



```
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$ ./fileSystemOper fat.dat dumpe
2fs
File system information for: fat.dat
Block size: 1024 bytes
Total blocks: 4096
Free blocks: 3937
Total directories: 3
Total files: 2
Name: \, Size: 0, Permissions: --, Creation Date: 2024-06-07 21:39:30, Modification Date: 2024-06-07 21:39:30, Password
Protected: No, Type: Directory, Start Block: 0
Name: \, Size: 0, Permissions: --, Creation Date: 2024-06-07 21:39:30, Modification Date: 2024-06-07 21:39:30, Password
Protected: No, Type: Directory, Start Block: 0
Name: \usr, Size: 0, Permissions: rw, Creation Date: 2024-06-07 21:40:26, Modification Date: 2024-06-07 21:40:26, Passwo
rd Protected: No, Type: Directory, Start Block: 1
Name: \file1.txt, Size: 16847, Permissions: rw, Creation Date: 2024-06-07 21:41:22, Modification Date: 2024-06-07 21:41:
22, Password Protected: No, Type: File, Start Block: 2
Name: \file2.txt, Size: 143188, Permissions: rw, Creation Date: 2024-06-07 21:41:34, Modification Date: 2024-06-07 21:41
:34, Password Protected: No, Type: File, Start Block: 19
berry@DESKTOP-092GAB6:/mnt/c/Users/lafci/Desktop/GTU/4. sınıf/4. sinif 2. donem/OS/HW222$
```