GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CSE 312 /CSE504

Operating Systems

Homework #3 Report

171044073

Berkan AKIN

Problem

FAT12 file system design will be done. In the book, the form of agriculture found in 4.30 and 4.31 will be discussed. file attributes will include size, last modification date and time, and name of the file. Define your directory table and directory entries; Define how and where you keep the free blocks; Define your super block that contains crucial information about the file system such as the block size, root directory position, block positions,etc

Write a C/C++ program that creates an empty file system as a (16 MB max) file. This file will include all the information about your file system including the super block, data blocks, free blocks, directories, data, etc.

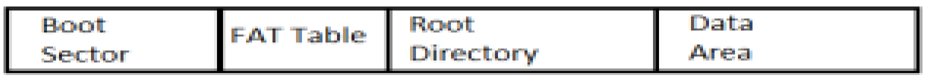
Solition

The above mentioned problems have been done. Part1 and part2 tasks given in the homework were done, part 3 and part 4 were not done

1)Disk Organization

To simulate the disk, we have a .dat file. Therefore, we need to edit this file.

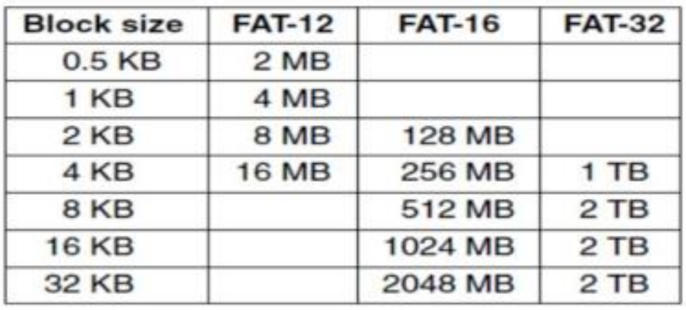
Fat12 File System



Disk organization of the FAT12 file system

1)Boot Sector

This section of the system contains information about the file system, such as Bytes per Sector, Sectors per Cluster, and the number of Reserved Sectors. I will keep it simple and place the information based on the block size, as shown in the diagram (superblock).



Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

Figure 4-31. Maximum partition size for different block sizes.

SuperBlock is the most important data structure of a file system. It is typically the first block created or encountered when a file system is created or mounted. The SuperBlock contains the fundamental properties and parameters of the file system.

The SuperBlock can include the following information:

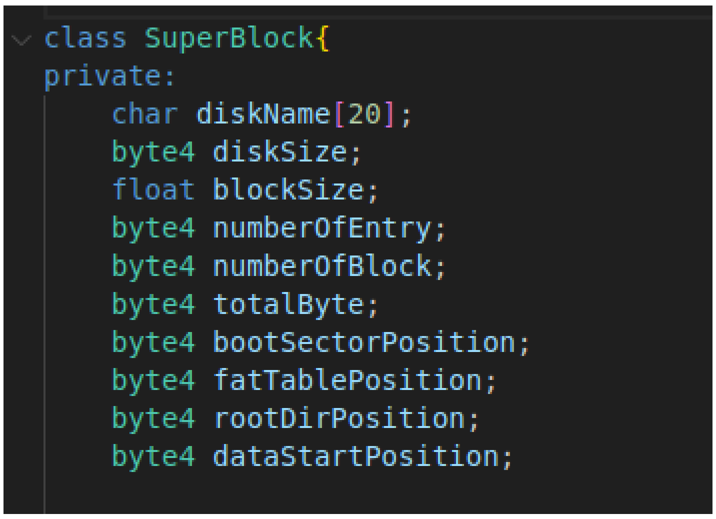
Disk name or label: The name or label of the disk or partition used by the file system.

- Disk size: The total size of the disk or partition used by the file system.

- Block size: The size of the blocks in the file system. A block is the smallest unit of operation for a file.
- Number of entries: The total number of entries (files or directories) in the file system.
- Number of blocks: The total number of blocks in the file system.
- Total bytes: The total size of the file system in bytes.
- Boot sector position: The memory position or disk location of the boot sector located at the beginning of the file system.
- FAT table position: The memory position or disk location of the File Allocation Table (FAT).
- Root directory position: The memory position or disk location of the root directory.
- Data start position: The memory position or disk location where the data region starts.

The SuperBlock contains the configuration of the file system and the necessary information for placing files and directories. These details are important for ensuring the proper functioning of the file system.

Code Snipped



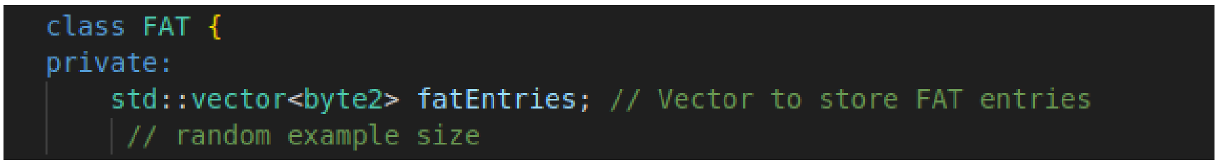
```
class SuperBlock{
private:
    char diskName[20];
    byte4 diskSize;
    float blockSize;
    byte4 numberOfEntry;
    byte4 numberOfBlock;
    byte4 totalByte;
    byte4 bootSectorPosition;
    byte4 fatTablePosition;
    byte4 rootDirPosition;
    byte4 dataStartPosition;
```

2) FAT(File Allocation Table)

FAT (File Allocation Table) is a data structure used in file systems. FAT is a table that tracks the allocation status of each cluster in a disk partition and manages the disk locations of files.

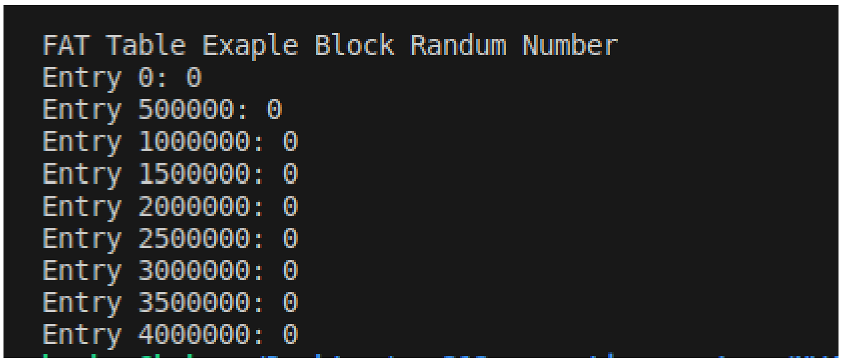
FAT keeps track of the scattered blocks of a file and their sequence on the disk. Each cluster represents the smallest unit in which a file is stored and typically consists of multiple sectors. FAT is used to determine the status of each cluster (such as free, occupied, or corrupted). A free cluster indicates an available space, while an occupied cluster represents a part of a file.

FAT is important for ensuring the integrity of file systems. During file system operations, the FAT table is updated to record the cluster numbers where files are stored in a distributed manner. The data blocks of a file are connected based on the specified sequence in the FAT table, enabling access to the file system and reading or writing files.



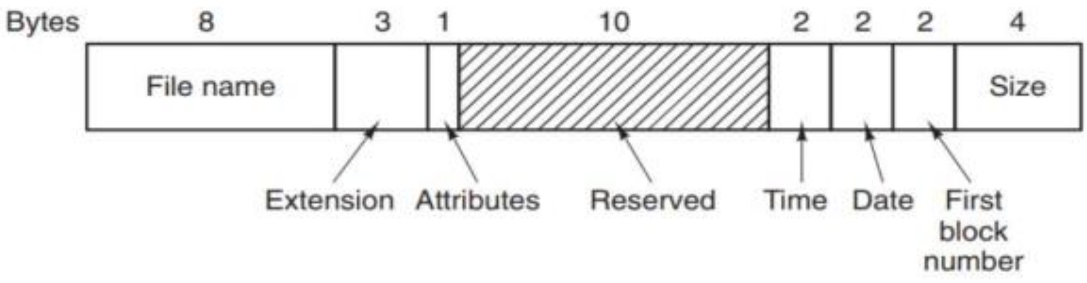
```
class FAT {
private:
    std::vector<byte2> fatEntries; // Vector to store FAT entries
     // random example size
```

Random printing of certain segments of the table

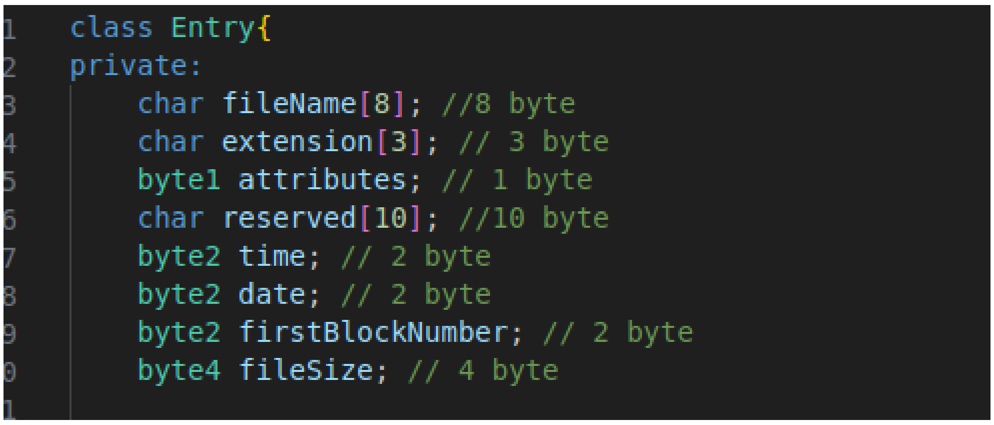


```
FAT Table Exaple Block Randum Number
Entry 0: 0
Entry 500000: 0
Entry 1000000: 0
Entry 1500000: 0
Entry 2000000: 0
Entry 2500000: 0
Entry 3000000: 0
Entry 3500000: 0
Entry 4000000: 0
```

The directory entries will be like this and will total 32 bytes.



Directory Entries Code Snippet

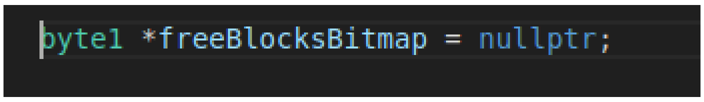


```
class Entry{
private:
    char fileName[8]; //8 byte
    char extension[3]; // 3 byte
    byte1 attributes; // 1 byte
    char reserved[10]; //10 byte
    byte2 time; // 2 byte
    byte2 date; // 2 byte
    byte2 firstBlockNumber; // 2 byte
    byte4 fileSize; // 4 byte

```

Free Blocks in File System

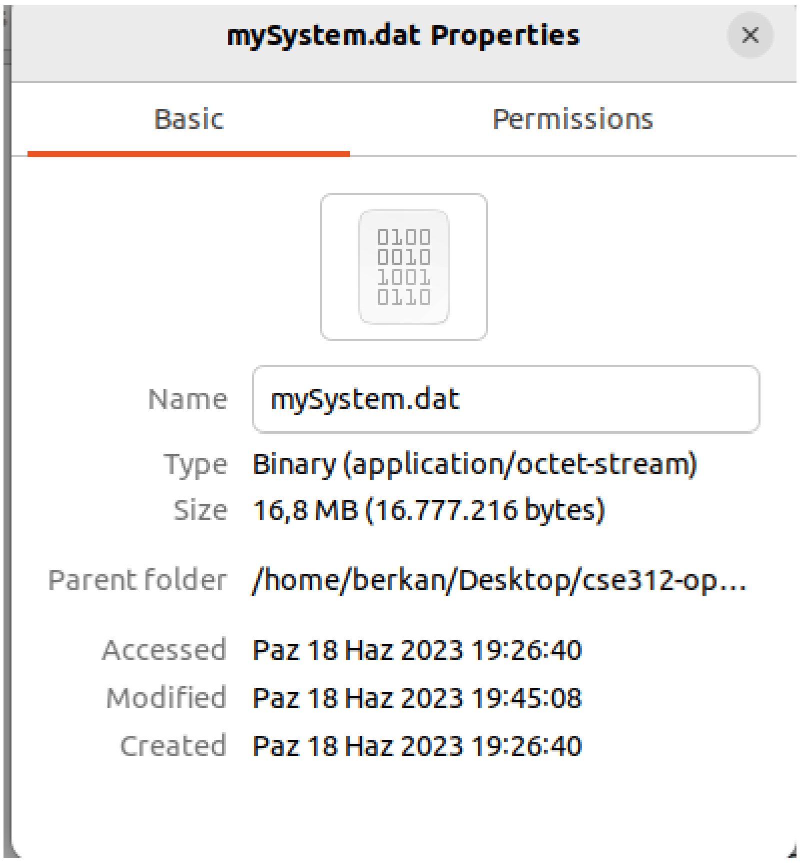
The bitmap data structure was used to keep track of empty blocks in the file system.



```
byte1 *freeBlocksBitmap = nullptr;
```

Test Case 4

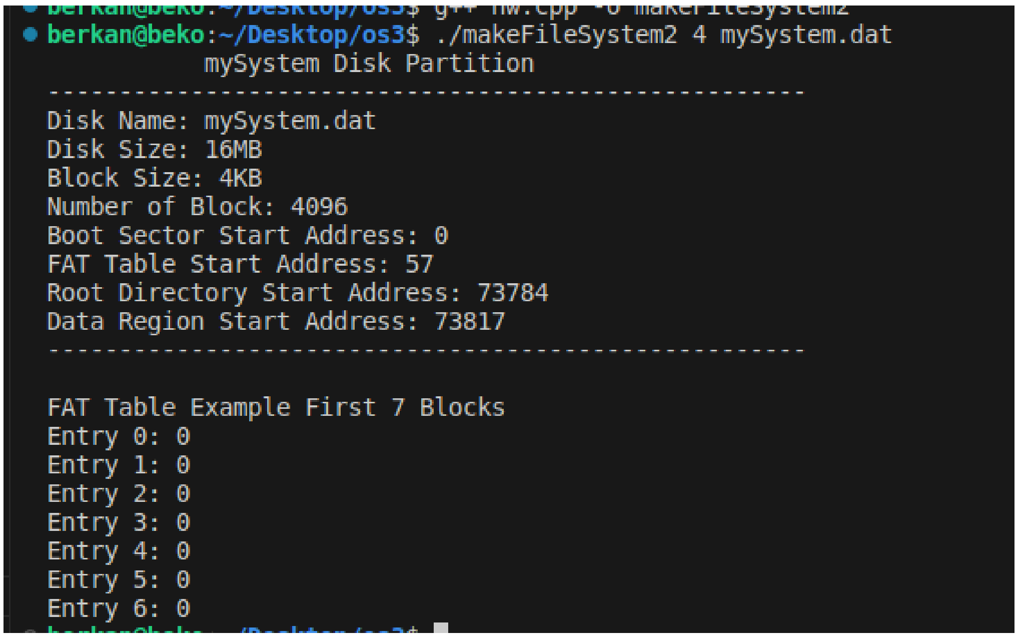
mySystem.dat file Information


mySystem.dat Properties

Basic | Permissions

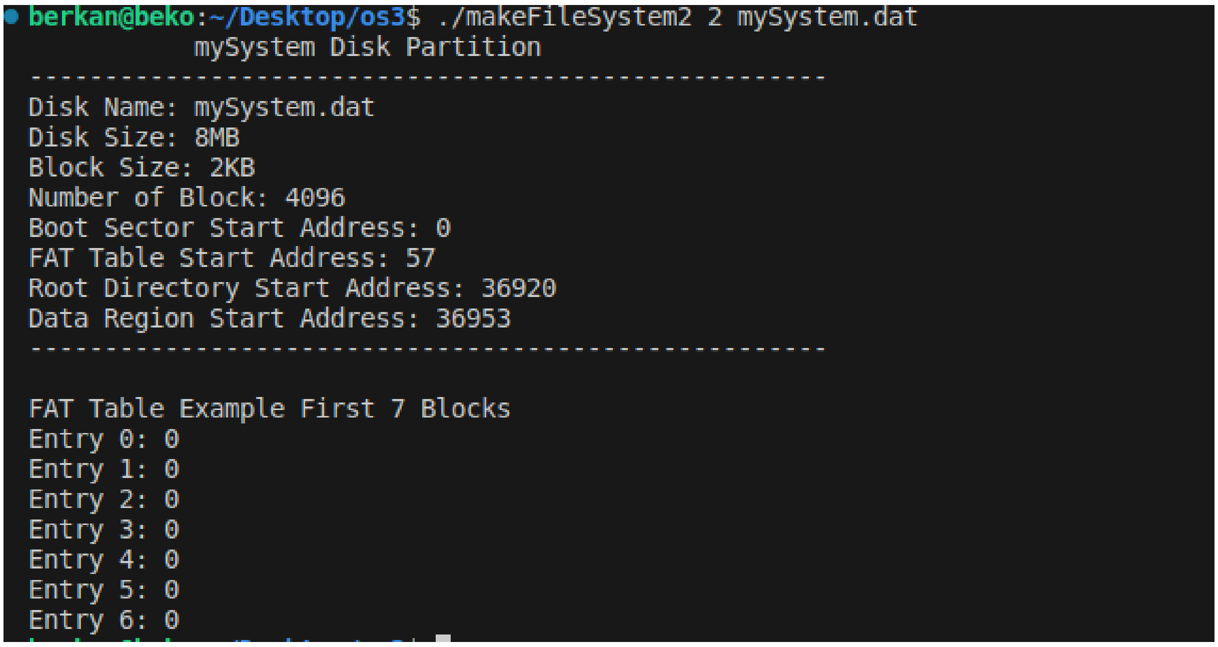
Name	mySystem.dat
Type	Binary (application/octet-stream)
Size	16,8 MB (16.777.216 bytes)
Parent folder	/home/berkan/Desktop/cse312-op…
Accessed	Paz 18 Haz 2023 19:26:40
Modified	Paz 18 Haz 2023 19:45:08
Created	Paz 18 Haz 2023 19:26:40

Block Size 4


```
• berkan@beko:~/Desktop/os3$ g++ hw.cpp -o makeFileSystem2
• berkan@beko:~/Desktop/os3$ ./makeFileSystem2 4 mySystem.dat
             mySystem Disk Partition
  -----------------------------------------------------
  Disk Name: mySystem.dat
  Disk Size: 16MB
  Block Size: 4KB
  Number of Block: 4096
  Boot Sector Start Address: 0
  FAT Table Start Address: 57
  Root Directory Start Address: 73784
  Data Region Start Address: 73817
  -----------------------------------------------------

  FAT Table Example First 7 Blocks
  Entry 0: 0
  Entry 1: 0
  Entry 2: 0
  Entry 3: 0
  Entry 4: 0
  Entry 5: 0
  Entry 6: 0
○ berkan@beko:~/Desktop/os3$
```

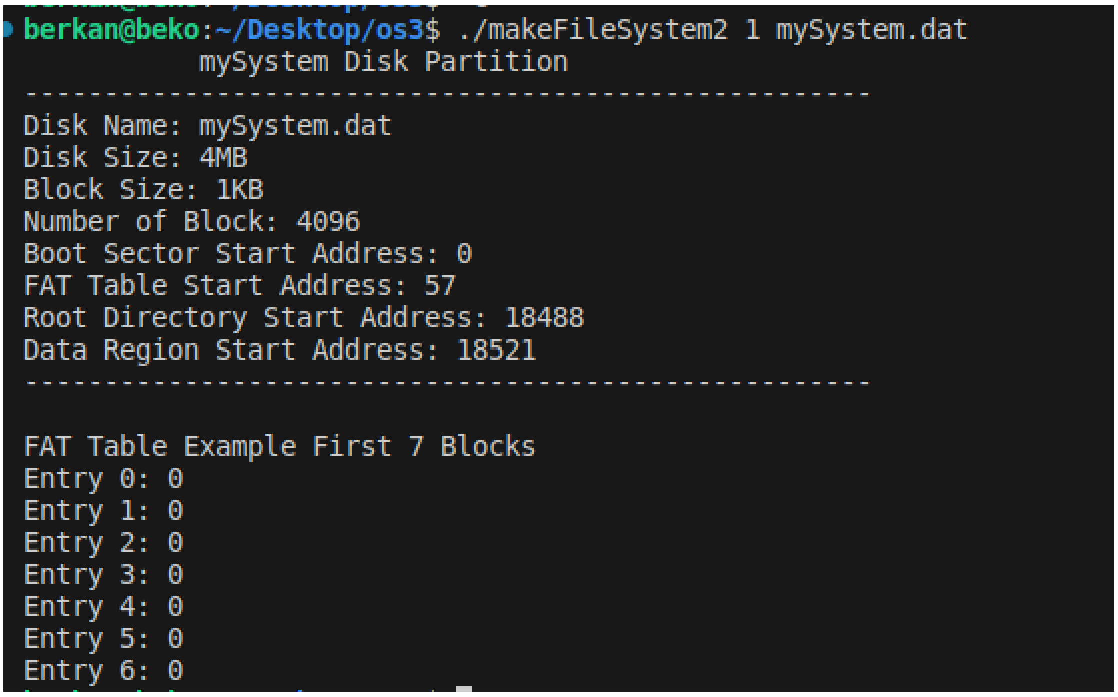
Block Size 2



```
berkan@beko:~/Desktop/os3$ ./makeFileSystem2 2 mySystem.dat
              mySystem Disk Partition
 -----------------------------------------------------
 Disk Name: mySystem.dat
 Disk Size: 8MB
 Block Size: 2KB
 Number of Block: 4096
 Boot Sector Start Address: 0
 FAT Table Start Address: 57
 Root Directory Start Address: 36920
 Data Region Start Address: 36953
 -----------------------------------------------------

 FAT Table Example First 7 Blocks
 Entry 0: 0
 Entry 1: 0
 Entry 2: 0
 Entry 3: 0
 Entry 4: 0
 Entry 5: 0
 Entry 6: 0
```

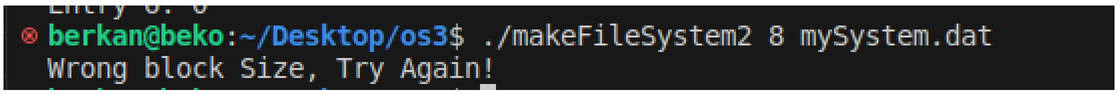
Block Size 1



```
berkan@beko:~/Desktop/os3$ ./makeFileSystem2 1 mySystem.dat
              mySystem Disk Partition
 -----------------------------------------------------
 Disk Name: mySystem.dat
 Disk Size: 4MB
 Block Size: 1KB
 Number of Block: 4096
 Boot Sector Start Address: 0
 FAT Table Start Address: 57
 Root Directory Start Address: 18488
 Data Region Start Address: 18521
 -----------------------------------------------------

 FAT Table Example First 7 Blocks
 Entry 0: 0
 Entry 1: 0
 Entry 2: 0
 Entry 3: 0
 Entry 4: 0
 Entry 5: 0
 Entry 6: 0
```

Wrong Block Size



```
berkan@beko:~/Desktop/os3$ ./makeFileSystem2 8 mySystem.dat
Wrong block Size, Try Again!
```