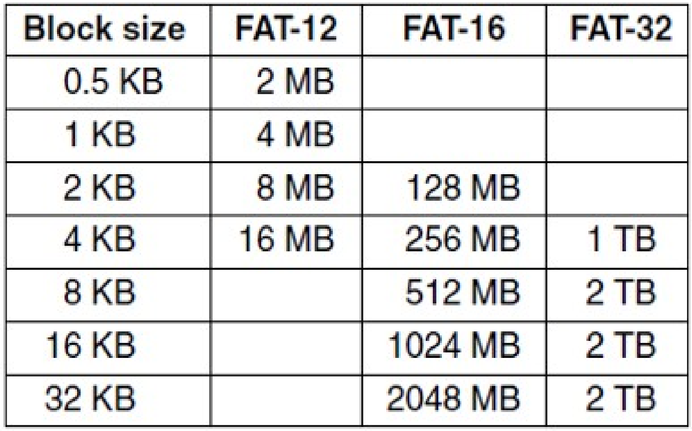
CSE 312
OPERATING SYSTEM
HOMEWORK 4 REPORT

1801042005
OĞUZHAN SEZGİN

GENERAL STRUCTURE

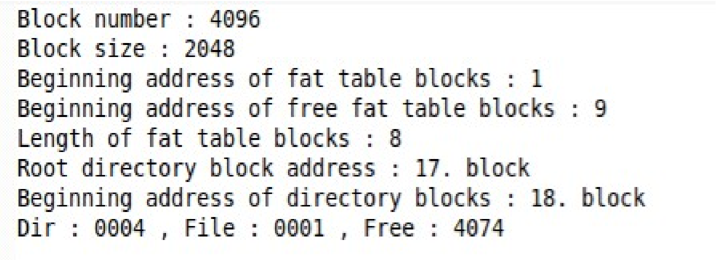
In my system there is a disk file and the system files stores in this files. The disk files divided by blocks. Each line of file (except for superblock) represents the disk blocks. Block size can change according to the user decision. There are 4 block sizes user can choose.



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 1 : Block Sizes

Each character of the disk file represents 1 byte. When user choose 1 KB block size, this means that there are 1024 character each block. There is a 1 superblock at the top of disk file. This block records the file system information.



```
Block number : 4096
Block size : 2048
Beginning address of fat table blocks : 1
Beginning address of free fat table blocks : 9
Length of fat table blocks : 8
Root directory block address : 17. block
Beginning address of directory blocks : 18. block
Dir : 0004 , File : 0001 , Free : 4074
```

Figure 2 : Super Block

Dir : Directory number in FAT-12 file system.
File : File number in FAT-12 file system.
Free : Free block number in FAT-12 file system.

Files and directory informations stores in little entry byte groups. Each file and directory has this entry.

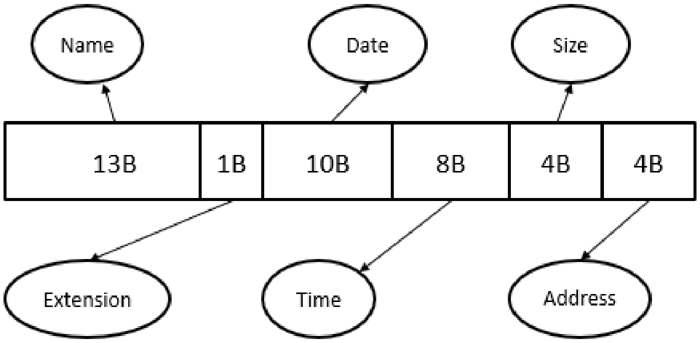
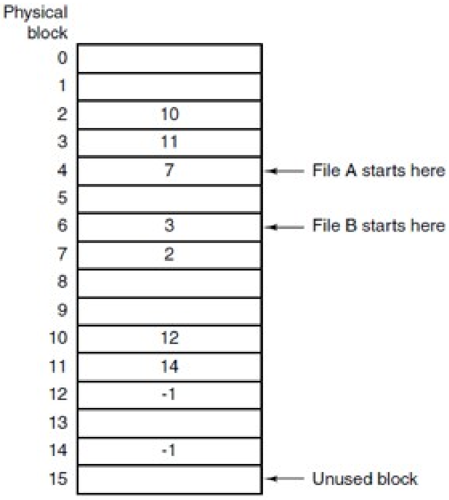


Figure 2 : Directory Entry

In this entry first 13 byte reserved for the file or directory name, 1 byte reserved for file extension (1 = directory , 0 = regular file), 10 byte for date , 8 byte for time , 4 byte for file size and 4 last 4 byte reserved for first block address of the file. Each file has this entry and stores in the parent directory block.

The fat table begins always after the super block. In this table use for traverse the all system and finds the file contents. Table has as many rows as the number of block number. Each row represents the block (e.g. 3. row represents 3.block). Each row has 4 byte. If a file size greater than the block size, system checks the fat table and finds the related block and then gets the block value. This value is the block with the rest of the file. If this value is -1 then this means that this block is end of the file.



Physical block		
0		
1		
2	10	
3	11	
4	7	← File A starts here
5		
6	3	← File B starts here
7	2	
8		
9		
10	12	
11	14	
12	-1	
13		
14	-1	
15		← Unused block

Figure 3 : Fat Table Structure

There is also free fat table. This table use for finding free block without traverse all blocks. This table has same row number. This rows value represents the free block. When system wants to free block, iterate this table then if it finds the greater than 0 value , this means that this value is the free block address. When file removed than system updates free table, it writes the first less than 0 value it sees.

GENERAL PROCESS

My FAT-12 file system support 6 file operations. They are ; “dir”,”mkdir”, “dumpe2fs”,”write”,”read” and “del” operations.

Operation “dir”

First it parse the given path and stores file names in 2D array(part3.c l:341). Then it iterate path with use fat table in loop(part3.c l:344). When it comes to the required block then it traverse block. It checks file name bytes from directory entries (part3.c l:374). Then it prints the all files in the directory.

Operation “mkdir”

First it parse the given path and stores file names in 2D array(part3.c l:808). Then checks the path rate(it is under the root or not (part3.c l:814). Then if it is not under the root it traverse blocks. When it comes to required block first reserve a free block for its contents(part3.c l:851). After that it creates directory entry with its information(part3.c l:863). Then it records directory entry to current block(part3.c l:864).

Operation “dumpe2fs”

This operations writes the superblock of the file system. It stores all data of the file system. When a file created or removed, this information updated (part3.c l:446).

Operation “write”

In this operation program finds the required block as the other. Then it reads the file(part3.c l:628) after that it creates the directory entry for new file (part3.c l:640) then it records the current block. After that it writes the file contents to disk.(part3.c l:642).

Operation “read”

In this operation program finds the required block as the other. Then it reads the required file block and stores a buffer array (part3.c l:550), after that it creates new file for write content and writes (part3.c l:567).

Operation “del”

In this operation program finds the required block as the other. Then save first block address of the file and removes directory entry from the current block(part3.c l:170) . Then it goes to the file block and removes the file entry from disk(part3.c l:176).